TO:	All Bidders	FROM:	CTBTO Procurement Services Section
DATE:	05 November 2026	REF.:	RFP No. 2026-0108/MOGAPI
		TEL. NO.:	+43 1 26030 6350
		EMAIL:	procurement@ctbto.org

SUBJECT: **Deadline Extension and Clarifications No. 1 to RFP 2026-0108/MOGAPI - Provision of Software Development and Maintenance of NET-VISA on a Call-Off Basis**

In reference to request for proposal (RFP) No. 2026-0108/MOGAPI concerning "Provision of Software Development and Maintenance of NET-VISA on a Call-Off Basis", kindly note the following:

- The **deadline for the submission of proposals is hereby extended** from 11 November 2026 **to Friday 20 November 2026 17:00 hours, Vienna (Austria) local time**.

- Please find attached the **Clarifications No. 1** to questions raised by interested bidders, including:

  - **Updated Attachment 2 - Technical Evaluation Criteria** document part of the RFP, which is hereby replaced with the attached amended version of this document (*Rev. 04 Nov 2026*) (revision highlighted in yellow for ease of reference);
  - **New Attachment A (NET-VISA Software Description Document)** and **Attachment B (IDC Database Schema-ODB)** which are hereby added as "Attachment A" and "Attachment B", respectively, to Annex B Terms of Reference part of this RFP.

  Clarifications No. 1 and its attachments are an integral part of the RFP documents and shall be considered in the preparation and submission of proposals.

We are looking forward to receiving proposals prior to the extended submission deadline on **Friday 20 November 2026 17:00 hours, Vienna (Austria) local time**.

Sincerely,

CTBTO Procurement Services Section

Attachments:

1. *Questions and Answers – Clarifications No. 1*
2. *Attachment 2 - Technical Evaluation Criteria – Rev. 4 Nov 2026*
3. *Attachment A – NET-VISA Software Description Document*
4. *Attachment B – IDC Database Schema-ODB*

Clarifications No. 1 - questions and answers

#	Bidders' Question	Answer
1	Daily rates across Call-off Periods. Note 1 of the Price Schedule Format states that the rates shall be firm and fixed throughout the term of the Contract, and Clause 12(c) of the Model Contract fixes the Annex C unit prices for the entire duration. However, the Price Schedule has separate daily rate columns for Call-off Periods 1, 2 and 3, and Clauses 12(a)(ii), 12(a)(iii) and 12(b)(ii) refer to a daily rate agreed for the optional extensions. May bidders quote different daily rates for the optional Call-off Periods 2 and 3 than for the firm 36-month Call-off Period 1?	**Yes.** Bidders may quote different daily rates for Call-off Periods 1, 2 and 3. The separate rate columns in the Price Schedule Format are intended for this purpose. The rates quoted for each Call-off Period shall be firm and fixed and, once the Contract is awarded, shall not be subject to adjustment for the duration of the Contract.
2	Pricing of on-site visits. Section 8 of the Terms of Reference provides for reimbursement of return flights against supporting documents and per diem at UN DSA rates, while Clause 12(d) of the Model Contract states that the Contract Price covers all costs, including travel and allowances […] Would the Commission accept on-site visits priced as an on-site daily rate plus a fixed lump sum per trip, with proof of travel provided on request but no ticket receipts?	Please refer to Note 4 of Attachment 3: Price Schedule Form. Bidders are free to determine the pricing methodology for their proposed travel costs. Bidders may propose travel costs either as firm and fixed costs or as estimated costs, in accordance with the solicitation documents. Accordingly, the Commission would accept a pricing approach based on a fixed lump-sum amount per trip, provided such costs are clearly reflected in the Financial Proposal as firm and fixed travel costs, indicating a breakdown of what is included in the lump sum and the benefits of such approach as representing a more cost-efficient quote. Where travel costs are proposed as all-inclusive lump sum(s), the provisions related to daily subsistence allowance (DSA), and reimbursement of travel costs other than DSA upon presentation of supporting documents in accordance with the Contract provisions would not apply.

Clarifications No. 1 - questions and answers - RFP No. 2026-0108/MOGAPI: Provision of Software Development and Maintenance of NET-VISA on a Call-Off Basis

		Either way, any proposed travel costs will be evaluated as part of the overall Financial Proposal in accordance with the solicitation documents best value for money assessment.
3	Minimum technical score. Section 1 (Technical Evaluation) of the Evaluation Criteria and Method states that bidders must obtain a minimum weighted score of 75 in Stage 2, while the following paragraph refers to a minimum weighted score of 69. Could you confirm which minimum applies?	Kindly find attached a revised Attachment C - Rev. of 3 Nov 2026, which now refers to the correct "**minimum weighted score of 75**".
4	Team Roster and domain expertise. Sections 6.1 and 6.4 of the Terms of Reference require each staff member on the Team Roster to meet all of the requirements of Section 6.4, and Section 5 (item 8) refers to subject-matter experts in seismology, hydroacoustics and infrasound. Our team provides expertise […]. We also consult external experts from time to time, but would not name them in the contract. Could you confirm that a proposal naming only our own staff on the Team Roster, without separately named domain experts or external consultants, would be compliant?	The Team Roster requirements under Annex B Terms of references -Section 6.4 and the desirable requirement under Section 5, item 8, are assessed separately. The subject-matter experts referred to in Section 5, item 8, are not required to be included on the Team Roster and do not need to be employees of the Bidder. However, to enable the Commission to assess this criterion, Bidders should identify the available experts and provide information on their qualifications, areas of expertise and availability. Where no such experts are identified, no score will be awarded under Section 5, item 8.
5	Infrasound association (ToR Section 4.2) Section 4.2.1 states that "both the association of pure infrasound events and the association of infrasound detections with seismic events are not yet performing at the required operational quality level." Could the Commission indicate whether any open references or publicly available documentation describe the current state of the infrasound component of NET-VISA, beyond the NET-VISA Software Description	Currently about 12.5% of events from NET-VISA are matched with the interactive analysis results. Kindly refer to the current state described in the attached NET-VISA Software Description Document and IDC Database Schema-ODB. These documents are herewith attached and added as part of Annex B- Terms of Reference part of this RFP.

Clarifications No. 1 - questions and answers - RFP No. 2026-0108/MOGAPI: Provision of Software Development and Maintenance of NET-VISA on a Call-Off Basis

	Document (Attachment A) and the publications available in the open literature (e.g., Le Bras et al., 2021)?	
6	As foreseen in Section 4.1.1 for bulletin quality, will the evaluation baseline, metrics and measurable objectives for infrasound be defined in the applicable Work Order? Is the expected outcome an iterative improvement of the existing model, or may a new model be considered if the investigation shows this is justified?	Yes, as specified in the Annex B - Terms of Refences, the evaluation baseline, metrics and measurable objectives for infrasound will be defined in the applicable Work Order. The Commission expects both approaches to be investigated (and, in case of success, implemented): 1. Iterative improvement of the existing model. 2. Switching to a new model if it performs better than exciting one.
7	New hydroacoustic travel-time tables and blockage maps (ToR Section 4.3) Section 4.3.1 refers to the new 3D blockage maps and travel-time tables received by the IDC for all hydroacoustic stations (hydrophone stations and T-stations). The ToR does not specify the expected size of these 3D tables. Will the Contractor be expected to implement additional data-compression or compact-storage techniques, or can the 3D tables provided by the IDC be used as they are? Are there any reference figures or constraints on the computing, memory, storage and processing-time resources that the resulting deliverable must satisfy, whether in IDC operations or in “NDC in a Box” deployments?	With reference to the Annex B – Terms of Refence - 3D tables provided by the Commission can be used as they are. The Commission does not expect additional compression to be made. Regarding computational resources: • The Commission has enough servers, physical and virtual, dedicated to development, operational use, database hosting etc. • “NDC in a Box” environment is just a virtual machine, with everything (backend, frontend, database) inside, intended to be launched on a PC with limited resources.
8	Use of LocOO3D in NET-VISA (ToR Section 4.4) Source localisation could potentially be incorporated into NET-VISA at several stages: (a) training, for estimating event-related distributions; (b) preliminary event localisation; or (c) refinement with respect to the probabilistic model.	The Contractor should propose and justify the integration point as part of the work.

Clarifications No. 1 - questions and answers - RFP No. 2026-0108/MOGAPI: Provision of Software Development and Maintenance of NET-VISA on a Call-Off Basis

	At which of these stages does the Commission expect LocOO3D to be integrated, or should the Contractor propose and justify the integration point as part of the work?	
9	[…] whether an international software firm working remotely, without an Austrian entity, is eligible and whether any on-site presence, security clearance, nationality restriction, or Austrian/EU business registration is mandatory.	There is no requirement for the bidder to be registered in Austria/EU nor have presence in Austria/EU. However, in accordance with the RFP requirements, bidders (including subcontractors) shall be legally established in one of the States Signatories of the Comprehensive Nuclear-Test-Ban Treaty (CTBT), the list of which is available in the CTBTO website at www.ctbto.org under Status of Signatures and Ratifications | CTBTO. Moreover, in accordance with the requirements of the Terms of Reference (ToR), the subsequent contractor's staff must be able to travel to Vienna, Austria as requirement by the Commission – kindly refer to sections 7 and 8 of the ToR. In addition please see response to Question 12 below.
10	Could you please clarify whether the solicitation sets mandatory minimum thresholds for corporate age, turnover, reference contracts, team certifications, or prior work with visa/border-control systems.	Kindly refer to the contractor's requirements (section 5 of the Annex B - ToR) and requirements for the Team Roster (section 6 of the Annex B ToR) and the Vendor Profile Form document part of this RFP. The Bidder shall have sufficient financial capacity to perform the contract in relation to the value of the proposed Financial Proposal.
11	Could you please clarify whether consortium or subcontractor participation is permitted.	The Commission does not prohibit the participation of consortia (joint offers) or subcontractors. Where a proposal is submitted by a consortium, bidders should clearly identify the lead entity and the respective roles and responsibilities of the participating entities, provide a copy of the relevant consortium agreement and provide the Vendor Profile Form for each of the entities

		part of the consortium. All members of the consortium must meet the eligibility criteria of the relevant RFP, in particular, they shall all be legally established in one of the CTBT States Signatories. In the event of award to the consortium, the Commission will contract with a single entity, normally the lead entity of the consortium. Any proposed subcontractors should also be clearly identified in the proposal, where applicable. In accordance with the RFP instructions, the information shall include the names, legal status, address and qualifications of the proposed subcontractor(s), and the scope of the subcontracted services. Further, the Proposal shall provide a statement that the bidder shall be fully responsible for the performance of sub-contractors. All sub-contractors shall also be legally established in one of the CTBT States Signatories.
12	Could you please confirm whether an India-incorporated development company may bid on this assignment and whether remote delivery is acceptable, subject to any required security or access controls? Please also advise whether technical clarifications should be submitted through your procurement platform rather than by email.	1. **Bidder's eligibility:** Bidders and their sub-contractors (if any) shall be legally established in one of the Comprehensive Nuclear-Test-Ban Treaty (CTBT) States Signatories. In particular, the goods and services (if any) to be rendered under the Contract shall have their origin in the States Signatories of the Comprehensive Nuclear-Test-Ban Treaty (CTBT), a list of which is available in the CTBTO website at www.ctbto.org under Status of Signatures and Ratifications | CTBTO. For purposes of this paragraph, "the origin" means the place from where the materials, goods and/or from which the services are supplied. Bidders shall foremost assess

		their eligibility and the origin of the proposed goods and/or services against this requirement. Please refer to the "Instructions for preparation and submission of Proposals" document part of this RFP. 2. **Remote delivery of services:** Yes, this is acceptable. Kindly refer to the Terms of Reference concerning off-site work. 3. **Tender clarifications submission:** These shall be submitted by email to procurement@ctbto.org and in line with Section "6. Request for Clarifications and Contacting the Commission" of the "Instructions for preparation and submission of Proposals" part of this RFP.
13	Are India-incorporated companies eligible to participate directly, or is an EU/Austrian establishment or local delivery partner required? Are consortium or specialist subcontractor arrangements permitted, and can relevant experience be demonstrated by a named specialist partner?	Kindly refer to responses to Question 12 above.
14	Does the RFP require previous NET-VISA experience or specialist experience in seismic monitoring, nuclear-test monitoring, scientific algorithms or related systems?	Previous NET-VISA experience is not required. Bidders may further refer to the contractor's requirements Section 5 of the Annex B - ToR and requirements for the Team Roster Section 6 of the Annex B -ToR for the requirements.
15	What are the principal technologies, languages, architecture and call-off support expectations for the existing NET-VISA codebase?	Kindly refer to the "Background" Section 2 of the Annex B ToR and to NET-VISA Software Description Document (Attachment A) and IDC Database Schema-ODB (Attachment B) attached to these Clarifications and added as part of Annex B ToR.

Clarifications No. 1 - questions and answers - RFP No. 2026-0108/MOGAPI: Provision of Software Development and Maintenance of NET-VISA on a Call-Off Basis

16	Are there minimum turnover, insurance, security-clearance, personnel or on-site requirements, and is an estimated contract ceiling available? Is Basic-level UNGM registration sufficient, and should all formal questions and the proposal be submitted through the CTBTO procurement platform?	Kindly refer to the contractor's requirements Section 5 of the Annex B -ToR and response to question 10 above. For the procedure to submit requests for clarifications, kindly refer to response to question 12 above.

Attachment 2 - Rev. 4 Nov 2026
Evaluation Criteria and Method

Software Development and Maintenance of NET-VISA

Ref No. in TOR	QUALIFICATION REQUIREMENTS (PASS/FAIL)	
	Requirements for the Contractor (PASS /FAIL)	**PASS/FAIL**
5.1	The Bidder is a company/organization/institution established for at least 7 years.	**PASS**/FAIL
5.2	The Bidder has demonstrated a minimum of 7 years of experience with all elements of the project management and software development lifecycle: eliciting and documenting business process flows, use cases, requirements, quality management plans, user acceptance testing, and end-user training.	**PASS**/FAIL
5.3	The Bidder has demonstrated a minimum of 7 years of experience in developing, maintaining and managing software systems of comparable scientific complexity and operational criticality. The Bidder has demonstrated practical expertise in Bayesian methods, machine learning, scientific computing, large-scale data processing, performance optimization, and industrial C++ and Python development.	**PASS**/FAIL
5.4	The Bidder has demonstrated a minimum of 7 years of experience in developing database-driven operational systems, automated testing, CI/CD, containerized deployment and maintenance of legacy scientific software.	**PASS**/FAIL
5.5	The Bidder has confirmed that the working hours of its relevant staff (e.g. project manager and lead developers), assigned to a Software Team, will overlap at least three hours with the Commission's working hours (9am to 5pm CET).	**PASS**/FAIL
5.6	The Bidder has confirmed that under exceptional circumstances support will be fully available during the working hours of the Commission (9 AM – 5 PM Vienna local time (CET/CEST)), without additional costs. This would be no more than one week per year.	**PASS**/FAIL
5.7	The Bidder has confirmed that English will be used for all written and oral communication with the Commission.	**PASS**/FAIL
7.2	The Bidder has confirmed that they shall submit Sprint reports to document completed work, personnel involved, time spent, open issues and blockers.	**PASS**/FAIL
6.1	The Bidder has proposed at least three (3) staff members and their CVs for the Team Roster and has confirmed that the Team Roster will be maintained throughout the duration of the Contract.	**PASS**/FAIL
	Requirements for Staff and Team Roster (PASS /FAIL)	**PASS/FAIL**
	Skills and experience for each staff on the Team Roster	
6.4.1	Each proposed staff member has an advanced university degree in computer science, data science, mathematics, physics or a related field (with accompanying explanation).	**PASS**/FAIL
6.4.2	Each proposed staff member has demonstrated a minimum of five years of recent professional experience within the last seven years using Python and C/C++ in substantial production-grade software projects of significant complexity.	**PASS**/FAIL
6.4.3	Each proposed staff member has demonstrated a minimum of five years of recent professional experience within the last seven years with SQL and database programming, ideally using Oracle and/or PostgreSQL databases, including experience designing data access layers and data models for an application.	**PASS**/FAIL
6.4.4	Each proposed staff member has demonstrated a minimum of three years of recent professional experience within the last seven years providing software maintenance and support services for complex and custom software systems.	**PASS**/FAIL
6.4.5	Confirmation that each proposed staff member has good communication skills in written and spoken English.	**PASS**/FAIL
	Skills and experience available amongst Team Roster staff	**PASS**/FAIL
6.5	The Bidder has demonstrated that the following combined skills and experience are available in the proposed Team Roster; at a minimum, two (2) of these requirements under *Section 6.5 "Skills and experience available amongst Team Roster staff"* are met by each individual in the Team Roster, and all of these skills and experience are available in the Team Roster in aggregate.	**PASS**/FAIL
6.5.1	The Bidder has demonstrated that the following skill and experience is available in the Team Roster: A minimum of three years of recent professional experience within the last seven years with CI/CD (continuous integration/continuous delivery) software deployment environments.	**PASS**/FAIL
6.5.2	The Bidder has demonstrated that the following skill and experience is available in the Team Roster: A minimum of three years of recent professional experience within the last seven years with the elements of the software development lifecycle: documenting requirements, user acceptance testing, and end-user training.	**PASS**/FAIL
6.5.3	The Bidder has demonstrated that the following skill and experience is available in the Team Roster: A minimum of three years of recent professional experience within the last five years as a Data Scientist, including practical experience in the development, validation and deployment of machine-learning or statistical models.	**PASS**/FAIL
6.5.4	The Bidder has demonstrated that the following skill and experience is available in the Team Roster: Minimum of one year of recent professional experience within the last two years designing, developing or integrating agent-based or agentic software systems.	**PASS**/FAIL

Only bidders who pass all above criteria will be considered for the point system evaluation (2nd stage)

No.	2.TECHNICAL REQUIREMENTS - EVALUATION CRITERIA AND METHOD	Points		
	Quality of the Proposal	**Max Points**	**Factor**	**Weighted score**
General	Extent to which all aspects of the ToR have been addressed in sufficient detail and clarity.	5	4	20
General	Understanding of the scope of work and the responsibilities of the Contractor	5	2	10
General	The proposed Team Roster demonstrates the capacity to tackle Services described in Section 4 and is explained in sufficient detail. Curricula Vitae contain relevant information demonstrating the proposed personnel's ability to perform the specific tasks under this Contract. The following information is provided for each staff member: Name, Role, Employed since, Type(s) of Service(s) from Section 4 in which the staff member may be involved, Curriculum Vitae	5	3	15
5.8	It is desirable that the Bidder has demonstrated access to subject-matter experts in seismology, hydroacoustic and infrasound who may be engaged when required. The Bidder has identified the available experts and provided information on their qualifications, areas of expertise and availability.	5	2	10
	Experience of Team Roster Staff	**Max Points**	**Factor**	**Weighted score**
6.4.2	Staff member demonstrates a minimum of five years of recent professional experience within the last seven years using Python and C/C++ in substantial production-grade software projects of significant complexity.	5	4	20
6.4.3	Each proposed staff member demonstrates a minimum of five years of recent professional experience within the last seven years with SQL and database programming, ideally using Oracle and/or PostgreSQL databases, including experience designing data access layers and data models for an application.	5	1	5
6.4.4	Each proposed staff member demonstrates a minimum of three years of recent professional experience within the last seven years providing software maintenance and support services for complex and custom software systems.	5	3	15
6.5.3	Peer-reviewed relevant publications indexed in the Web of Science Core Collection or Scopus, or published in the proceedings of conferences ranked A* in the ICORE 2026 Conference Rankings, are desirable.	5	3	15
6.5	Adequacy of the number and composition of proposed staff to ensure full coverage of all required skills and experience. Each team member shall meet at least two (2) requirements, while all requirements shall be covered collectively by the Team Roster. (6.5.1; 6.5.2; 6.5.3; 6.5.4)	5	1	5
	TOTAL - Technical Evaluation	45		**115**

Technical Evaluation:

The technical evaluation process will be done in two stages:

1) Stage 1: Technical proposals will first be evaluated against the mandatory requirements outlined in section 1 above, on a PASS/FAIL basis.

Compliance with all mandatory requirements is required in order to pass stage 1 of the technical evaluation and to be further considered for stage 2 of the evaluation process;

2) Stage 2: The technical proposals that have passed stage 1 of the technical evaluation process, will be evaluated against the weighted criteria set forth in the evaluation matrix above.

In order to pass this stage, bidders must obtain a **minimum weighted score of 75** and in accordance with the scoring indicated in the "Table 2 – Scoring" below.

Failing one of the criteria in Stage 2 will not result in the entire technical proposal failing; provided that the technical proposal obtains a minimum weighted score of 75 as already stated.

TABLE 2

Points	Scoring
0	Unsatisfactory - Response incomplete, inadequate and/or non-responsive to the criterion. Bidder does not clearly understand the criterion.
1-2	Weak - Does not meet the minimum requirements of the criterion
3	Good - Meets the minimum requirements of the criterion
4	Very good - Exceeds the minimum requirements of the criterion in some areas
5	Excellent - Exceeds the minimum requirements of the criterion in all areas

2. Financial and commercial evaluation

Once the technical evaluation is finalized, the financial offers of the technically compliant bidders will be evaluated in accordance with the formula given below:

X= Max Available Points * Y/Z

Legend:

X= points to be assigned to the offer being evaluated

Y= price of the lowest priced, technically compliant offer

Z= price of the offer being evaluated

The weight of the technical and financial components is 60% and 40% respectively.
The Contract will be awarded to the bidder who receives the highest combined score resulting from the technical and financial evaluations, subject to contractual acceptability.

NET-VISA Software Description Document

Nimar S. Arora[1], Stuart Russell[1,2], Heidi A. Kuzma[1]

April 20, 2026

[1]Bayesian Logic, Inc.; [2]University of California, Berkeley

Sponsored by the Preparatory Commission for the Comprehensive Nuclear Test-Ban Treaty Organization (CTBTO) under Contract No. 2012-2107, Contract No. 2016-2048, Contract No. 2019-1884, and Contract No. 2023-1715.

Abstract

NET-VISA stands for *NETwork processing - Vertically Integrated Seismic Analysis*. The package is composed of a physics-based, probabilistic model and a heuristic inference algorithm designed to find the most probable set of seismic events which can explain a series of arrivals detected by a global seismic network. It has been extended to find events in any of three mediums – rock, air, and water and it supports seismic, hydro-acoustic, and infrasound sensors. The basic components of NET-VISA are a *Generative Model* (GM) and an *Inference Algorithm* (IA). The GM contains the information necessary to evaluate the probability of events and arrivals. The IA searches for the set of events that has the *maximum a posteriori* (MAP) probability - that is the one with highest probability given the input data. It works by proposing a set of possible events and iterating between associating arrivals and refining the time, location and magnitude of the events given the associated arrivals. This document is divided into three parts: an *Introduction* which gives an overview of NET-VISA and the problems it solves; a *Scientific Description* which details the mathematics and algorithms used by the NET-VISA package, including notes as to how it is implemented in the IDC-specific package, IDCNETVISA: and a *Practical Guide* which gives details of the various scripts and utilities necessary to install and run IDCNETVISA. A README file is included with each distribution of IDCNETVISA which describes in detail how to install the package and includes examples of how to run it.

Contents

I Introduction **7**

1 NET-VISA project **7**

2 Deliverables **7**

3 Statement of the problem **8**

4 IDCNETVISA **9**

II Scientific Description **10**

5 Method **10**
5.1 Bayes' rule . . . 11
5.2 Conditional independence . . . 11
5.3 MAP world . . . 12
5.4 Event and arrival scores . . . 13
5.5 Training . . . 14
5.5.1 New stations . . . 14
5.5.2 Modifications to internal models which do not affect detections . 15
5.5.3 Fundamental changes . . . 15
5.6 Inputs and outputs . . . 15
5.7 General program structure . . . 15

6 Generative Model **17**
6.1 Mathematical conventions . . . 17
6.2 Probability of attributes of events and arrivals . . . 18
6.2.1 Relationship between an event and an associated arrival . . . 19
6.2.2 Probability of the attributes of an event . . . 19
6.2.2.1 Event Medium . . . 19
6.2.2.2 Rate of Events . . . 21
6.2.2.3 Event Time . . . 21
6.2.2.4 Event Location . . . 21
6.2.2.5 Event Depth . . . 22
6.2.2.6 Event Magnitude . . . 22
6.2.3 Probability of the attributes of an associated arrival given an event 23
6.2.3.1 Predicted Arrival Time, Amplitude, Azimuth, Slowness 23
6.2.3.2 Event Derived Feature (EDF) sets . . . 24
6.2.3.3 Data is Available . . . 24
6.2.3.4 Detection is Made . . . 24
6.2.3.5 Measured Arrival Amplitude . . . 28
6.2.3.6 Hydro-Acoustic Arrival Energy . . . 30
6.2.3.7 Hydro Fraction Time . . . 30
6.2.3.8 Measured Arrival Azimuth . . . 32
6.2.3.9 Measured Arrival Slowness . . . 32

6.2.3.10 Arrival Phase Label . . . 34
6.2.3.11 Infrasound Center Frequency . . . 34
6.2.3.12 Infrasound Family Size . . . 35
6.2.3.13 Infrasound Consistency . . . 35
6.2.3.14 Infrasound Duration / Family Size . . . 35
6.2.4 Probability of the attributes of false arrivals . . . 37
6.2.4.1 Number of false arrivals. . . . 37
6.2.4.2 Measured Arrival Time (false arrival) . . . 37
6.2.4.3 Measured Arrival Amplitude (false arrival) . . . 39
6.2.4.4 Measured Arrival Energy (false arrival) . . . 39
6.2.4.5 Measured Arrival Azimuth (false arrival) . . . 40
6.2.4.6 Measured Arrival Slowness (false arrival) . . . 40
6.2.4.7 Arrival Phase Label (false arrival) . . . 42
6.2.4.8 Infrasound Clutter . . . 42
6.2.4.9 Hydroacoustic Clutter . . . 42
6.2.5 Probability of the attributes of coda arrivals . . . 44
6.2.5.1 Determining which arrivals in SEL3 are coda arrivals to construct a coda training set . . . 44
6.2.5.2 Detection is made (coda arrival) . . . 45
6.2.5.3 Measured Arrival Time (coda arrival) . . . 46
6.2.5.4 Measured Arrival Amplitude (coda arrival) . . . 46
6.2.5.5 Measured Arrival Azimuth (coda arrival) . . . 47
6.2.5.6 Measured Arrival Slowness (coda arrival) . . . 48
6.2.5.7 Arrival Phase Label (coda arrivals) . . . 48
6.3 Probability of a world . . . 49
6.3.1 Probability of an event . . . 49
6.3.2 Joint probability of a set of events . . . 49
6.3.2.1 Probability of the total number of events in a time period 49
6.3.2.2 Probability of a set of events . . . 49
6.3.3 Probability of an arrival . . . 49
6.3.4 Joint probability of sets of arrivals . . . 50
6.3.4.1 Joint probability of a set of associated arrivals . . . 50
6.3.4.2 Joint probability of a set of false arrivals . . . 51
6.3.4.3 Joint probability of a set of coda arrivals . . . 51
6.3.5 Probability of a world . . . 51

7 Inference Algorithm 52
7.1 Overview . . . 52
7.2 Event and Arrival scores . . . 52
7.3 Algorithm . . . 53
7.3.1 Birth Move . . . 54
7.3.2 Improve-arrival . . . 55
7.3.3 Improve Event . . . 55
7.3.4 Death-Move . . . 55
7.3.5 Finish . . . 56

III Practical Guide for IDCNETVISA 56

8 Scripts and Utilities **57**
8.1 README . . . 57
8.2 MANIFEST.IN . . . 57
8.3 PYPROJECT.TOML . . . 57
8.4 REQUIREMENTS.TXT . . . 57
8.5 SETUP.PY . . . 57
8.6 IDCSHEMA.SQL . . . 57
8.7 MAKEINFRAPRIOR.PY . . . 58
8.8 POPULATE_ISC.PY . . . 58
8.9 Installed Scripts . . . 58
8.9.1 IDCNETVISA_LEARN . . . 59
8.9.2 IDCNETVISA_INFER . . . 61
8.9.3 IDCNETVISA_INFER_BG . . . 68
8.9.4 IDCNETVISA_DEBUG . . . 68
8.9.5 IDCNETVISA_MATCH . . . 69
8.9.6 IDCNETVISA_ADD_SITE . . . 73
8.9.7 IDCNETVISA_MERGE_MODEL . . . 74
8.9.8 IDCNETVISA_VISUALIZE . . . 74
8.9.9 IDCNETVISA_VISUALIZE_PARQUET . . . 74
8.9.10 IDCNETVISA_RUNNER . . . 75
8.9.11 IDCNETVISA_COMPARE_MODELS . . . 77
8.10 Docker Images . . . 79
8.10.1 NET-VISA Image . . . 79
8.10.2 PREREQ Image . . . 80

9 Learning Schedule **80**

10 Model Schema **81**
10.1 Arrival SNR . . . 81
10.2 Arrival SNR header . . . 82
10.3 Arrival Amplitude . . . 82
10.4 Arrival Azimuth . . . 82
10.5 Arrival Phase . . . 82
10.6 Arrival Slowness . . . 82
10.7 Arrival Time . . . 83
10.8 Event Detection . . . 83
10.9 Event Magnitude . . . 83
10.10SNR of Noise Arrivals . . . 84
10.11Amplitude of Noise Arrivals . . . 84
10.12Phase of Noise Arrivals . . . 84
10.13Slowness of Noise Arrivals . . . 84
10.14Hydro Fraction Time . . . 85
10.15Info . . . 85
10.16Rate of Events . . . 85
10.17False Detection Rate . . . 85
10.18Phase . . . 85
10.19QC . . . 86
10.20Coda Model . . . 86

10.20.1 Coda Arrival Azimuth . . . 86
10.20.2 Coda Detection Probability . . . 86
10.20.3 Coda Arrival Phase . . . 87
10.20.4 Coda Arrival SNR . . . 87
10.20.5 Coda Arrival Amplitude . . . 88
10.20.6 Coda Arrival Slowness . . . 88
10.20.7 Coda Arrival Time . . . 88
10.21 Site . . . 89

11 Parquet Data File Schema **89**
11.1 Assoc Parquet File (assoc.parquet) . . . 89
11.2 Arrival Parquet File (arrival.parquet) . . . 91
11.3 Site_Up Parquet File (site_up.parquet) . . . 92
11.4 Event_Det Parquet File (event_det.parquet) . . . 93
11.5 Site and Phase CSV files (phase.csv and site.csv) . . . 93

12 Event Bulletin Database Schema **94**

13 Release Management at the IDC **100**
13.1 Version Numbers . . . 100
13.2 Testing and Deployment . . . 101
13.2.1 Release Candidate . . . 101
13.2.2 DEVLAN and OPS . . . 101
13.2.3 QANDEF . . . 102

A List of Symbols **103**

B Algorithms **104**

C Debugging Missed Events **106**
C.1 Bulletin Matching and Missed Event Detection . . . 106
C.1.1 Running the Matching . . . 106
C.1.2 Parameter File Configuration . . . 107
C.1.3 Understanding Match Output . . . 107
C.2 Event Score Diagnostics . . . 108
C.2.1 Running the Score Diagnostic . . . 109
C.2.2 Understanding Score Diagnostic Output . . . 109

Part I
Introduction

1 NET-VISA project

NET-VISA stands for *NETwork processing - Vertically Integrated Seismic Analysis*. It is a software package composed of a physics-based, probabilistic model and a heuristic inference algorithm designed to find the most probable set of events which can explain a series of arrivals detected by a global network of seismic, hydro-acoustic and infrasound sensors. The version of NET-VISA described here (called IDCNETVISA) is currently being installed for testing at the International Data Centre (IDC) of the Comprehensive Nuclear Test Ban Treaty Organization (CTBTO). It has been shown that under normal circumstances, NET-VISA produces a bulletin that is more complete and accurate than the IDC's best automatically produced bulletin, the third Standard Event List, SEL3. In fact, it has also been shown to be more complete than even the analyst-reviewed LEB bulletin [Arora, 2012a, Arora et al, 2010(b), Arora et al, 2012].

Originally conceived during a series of workshops held to explore the application of Data Mining and Machine Learning technologies to problems in data analysis at IDC, NET-VISA was developed by scientists at the University of California, Berkeley in collaboration with CTBTO. Preliminary results and descriptions of the NET-VISA system are also given in a doctoral dissertation and previous CTBTO reports [Arora et al, 2010(a), Arora et al, 2011, Arora 2012b]. After 2011 the project was transferred to Bayesian Logic, Inc. for further development.

2 Deliverables

This report was originally submitted in fulfillment of Deliverable 4.2 of Solicitation No. 2011-0321, addressing Task 6 of the project: *Complete detailed documentation of the operational software*. The deliverable is composed of several parts,

1. this *Introduction,*
2. a *Scientific Description* which details the mathematics and algorithms in NET-VISA,
3. *a Practical Guide* which gives an overview of the scripts and utilities necessary to run IDCNETVISA.
4. Appendices containing a list of symbols, detailed descriptions of algorithms and the IDCNETVISA database schema.

Section 12 contains the database schema that constitutes Deliverable 4.3 of the solicitation.

3 Statement of the problem

Each day, the sensors of the International Monitoring System (IMS) of the CTBTO collect gigabytes of data which is relayed via secured satellite link to the IDC. An automatic signal processing system sifts through the data to identify features of interest that might be the signals arriving from earthquakes or man-made seismic events, ice calving, blast-fishing bombs, whale sounds, sonic booms, meteorites entering the atmosphere, gas flares, minefield explosions etc. Once a *detection* has been made, the IDC system computes a number of attributes for it, including its arrival time, amplitude, azimuth, and slowness, for example. In some cases additional derived attributes are computed such as seismic phase (Seismic energy travels in *phases*. A single event causes multiple *arrivals* to be registered at a given station, each with a different phase), or the hydro-acoustic energy carried in different frequency bands. These are all features of the arrival of energy from a true event. Nevertheless, they might not actually have been caused by anything real. The "arrival" might just reflect station noise, in which case its attributes tend to be random numbers, or it may be caused by a local uninteresting source such as a gas flare, which are referred to as *clutter* sources. Or it might be from a weak event that occurred near the station and was not picked up at any other station. Or it might stem from a false detection made out of scattered energy immediately following a true arrival. In this case, its attributes are closely correlated to those of the arrival that preceded it. False arrivals such as this are called *coda* arrivals.

The fundamental challenge of creating an event bulletin is to work backwards from a list of arrivals to come up with a set of events that can explain them without creating false events. This is a difficult problem because in order to find the time, location and magnitude of an event, it is necessary to know when, where and how strongly it was detected - in other words, to know which arrivals are *associated* with it. However, in order to associate arrivals with events it is necessary to know when and where the events took place.

In the IDC processing pipeline, detected signals from a single station are examined first to determine an initial phase label (e.g. Pn, P, Lg). Next, the detections from the entire network are clustered together to form events based on their arrival time, azimuth, slowness, amplitude, and phase label. During this stage, phase labels may be modified according to a limited set of transformation rules. Locations of the events are then computed using their associated arrivals. Further attributes of the event like magnitude are then computed from the amplitude measurements. Finally, other global measures as well as other detection attributes like SNR are examined using simple heuristics to determine event quality, and then a decision is made to either keep the event or discard it. With many possible events, false detections almost always are associated with an event; once an event has several associations it does not get discarded, so many false events are reported. When false detections get associated with true events, they distort the solution for the event's position and time.

Perhaps more importantly, events generated this way are always physically possible but are often not *plausible*. An event might, for example, be built from arrivals from stations halfway around the world with no detections from stations nearby. An analyst would immediately discard this event on the basis of negative evidence: it was not seen where it should have been. But because it is physically possible it remains on the automated list.

All of the events in the SEL3 need to be closely examined by highly-trained human analysts before the final data products of the IDC are released. These final bulletins are the

carefully curated Late Events Bulletin (LEB) and its cousin the Reviewed Events Bulletin (REB). The REB is the same as the LEB except that a number of events have been removed which do not meet pre-defined criteria, such as having a minimum number of defining arrivals from primary IDC stations associated with them. In order to produce these lists almost every event from SEL3 must be corrected by hand. Close to half are thrown out or completely rebuilt using detections that had been erroneously assigned to other events.

The ultimate goal of NET-VISA is to produce an automatic bulletin that is an improvement over SEL3 in several ways:

1. it produces fewer false events without missing events,
2. it finds more events which will pass analyst scrutiny,
3. it associates more arrivals correctly with events,
4. it finds lower magnitude events,
5. it locates events more accurately.

In order to do these things, it needs to solve the following problems (not necessarily in this order):

- generate a list of likely events,
- determine which arrivals can be associated with events (even if their phase label has to be changed),
- classify unassociated arrivals as false or coda,
- work out the time, magnitude and location for the events.

Furthermore, it needs to solve these problems using methods which are, as much as possible, physically and experimentally realistic.

4 IDCNETVISA

IDCNETVISA is the version of NET-VISA which is specifically designed to run at IDC and is integrated with the IDC database. It is a computer program written in the PYTHON and c programming languages. The basic scripts used to run IDCNETVISA are IDCNETVISA_LEARN and IDCNETVISA_INFER (accompanied by LEARN.PAR and INFER.PAR which contain parameters necessary to run the scripts). The IDCNETVISA distribution includes a README file which gives up to date instructions for the package including installation and integration with the IDC database and tools and examples of how to run the various scripts.

IDCNETVISA is currently installed on the DEVLAN test platform at IDC. It can be run in parallel to the Global Association (GA) program which produces SEL3. The event bulletin that is output by IDCNETVISA is called the VISA bulletin and is structurally equivalent to SEL3. VISA can be reviewed by analysts using the same Analyst Review Software (ARS)

they use to review SEL3. A bulletin which is equivalent to LEB can therefore be produced from VISA and directly compared to the standard LEB. This bulletin is called VSEL3.

Part II
Scientific Description

5 Method

The word *inference* means the process of reaching a *conclusion* based on *evidence* and *reasoning*. NET-VISA uses *probabilistic inference* to work out the event bulletin (its conclusion) which best explains a set of arrivals at IMS primary and auxiliary stations (evidence). It associates arrivals with events, finding the correct phase label for the arrivals. Arrivals which are not associated with events are classified as either false (including clutter) or coda.

A full set of events, associations, phase labels and false/coda designations proposed to explain a set of arrivals is called a *world*. Reasoning, in the context of NET-VISA, means assigning a *probability* to various hypothesized worlds and searching for the one that has the highest probability and *is consistent with the evidence.* High probability worlds tend to contain events which happen often in nature (such as earthquakes along tectonic plate boundaries). But they can also have unusual events if those events are either supported by high quality evidence or it is difficult to find a better explanation for the data. Once the highest probability world has been found, the VISA bulletin is assembled from the events and associations in it. In a final processing step, standard IDC software (LIBLOC and LIBMAG) is used to make a final location and magnitude estimation for each VISA event.

NET-VISA is composed of two main parts, a *Generative Model* (GM) which gives instructions for evaluating the probability of a world and an *Inference Algorithm* (IA) which performs the search. The GM is essentially a template which formalizes the relationships between factors in event network analysis. The relationships are described in terms of physical laws and Probability Distribution Functions (PDFs). For example, the relationship between the time of an event and the detection of a phase at a given station is predicted using a deterministic function, the IASPEI earth model [Kennett and Engdahl,1991]. Once this prediction has been made, the difference between predicted and observed arrival times is modeled as a Laplace distribution. The parameters of this distribution along with many others in the GM must be found empirically. This is done by *training* the GM using historic arrivals and LEB bulletins. Different parameters are found for each station and each phase. Because the details of the IMS network and the nature of detected signals may change with time, IDCNETVISA needs to be periodically retrained so that the parameters of the PDFs accurately reflect the current network. During training, a *parameter file* (sometimes called the *model file*) is generated containing the model parameters which can then be used in subsequent runs of the IA.

5.1 Bayes' rule

In probability theory, the joint probability of two things, for example an event e and an arrival a , is the probability of one given the other times the probability of the other,

$$P(e, a) = P(e|a)P(a) = P(a|e)P(e).$$

Simple rearrangement of this equation leads to Bayes' rule:

$$P(e|a) = \frac{P(a|e)P(e)}{P(a)}.$$

In Bayesian parlance, the left hand term, $P(e|a)$ is called the *posterior* probability (in this case of the event given the arrival). $P(a|e)$ is the likelihood of the arrival given the event. $P(e)$ is the *prior* probability of the event. In NET-VISA, the event prior represents experience. Although events can occur anywhere on the globe, many years of data suggest that seismic events, for example, tend to occur in known seismic zones. If, for example, two proposed seismic events can explain an arrival with equal likelihood, the event which occurs along a tectonic plate boundary is assigned a higher posterior probability than an event in the middle of a continent. [1]The third term, $P(a)$ is the probability of the *evidence* or data.

The GM gives instructions for computing the prior probability of events and the likelihood of arrivals. Bayes' rule is invoked to compute the posterior probability of events.

If the probability of an event and an arrival are independent of each other (which would be the case if the arrival was a false arrival or was associated to another event), then their joint probability reduces to the product of their individual probabilities,

$$P(e, a) = P(e)P(a).$$

5.2 Conditional independence

NET-VISA worlds and the GM can be drawn as a series of graphs made up of nodes connected by arrows. Inside the nodes are entities (events and arrivals) or attributes of events and arrivals and instructions for computing a value for the node based on the values in the nodes that point to it. This value can either be a deterministic value, computed using a physical law, a conditional probability computed from a PDF, or a sample drawn from the PDF.

[1]Note that due to the particular mission of IDC, it may not be desirable to introduce a geographically-based belief in where events occur. It is possible to give all events equal probability by using a uniform prior for event location. See Section 6.2.2.4.

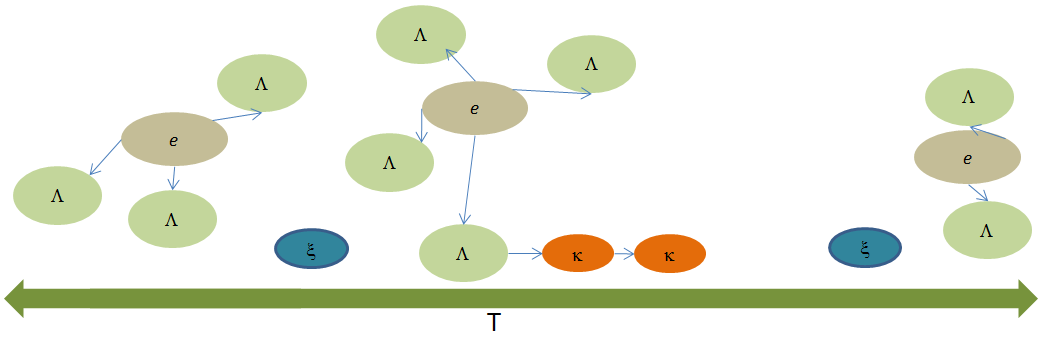


Figure 1 – A hypothetical world. Events are denoted as e, arrivals associated with events are denoted Λ, false events are denoted ξ, and coda arrivals are denoted κ. Arrows indicate dependence. The job of NET-VISA is to figure out the structure of this graph.

A hypothetical NET-VISA world is shown in Figure 1. Arrows indicate *dependence*. The attributes of an arrival entity (its time, amplitude, phase, slowness and azimuth) depend on the attributes of the event it is associated with (time, location, depth and magnitude). Or, if the arrival is a coda arrival, its attributes depend on the attributes of the arrival immediately preceding it. If a node has no arrows pointing to it then it contains a PDF which is not dependent on any other factor. This is the case with false arrivals. They occur randomly and are uncorrelated with events or other arrivals. Following the arrows forward, arrivals are *predicted* by knowledge of events. Going backward, arrivals constitute *evidence* for events; given a set of arrivals, a corresponding set of events may be *inferred*.

If one node points to two or more others, and the value of that node is known, then the values in the nodes that it points to (its *children*) are *conditionally independent* of each other, given the value of the *parent*. Therefore, a set of arrivals at different stations can be treated as if they are independent of each other given that they are all associated to the same event. This is important because otherwise it would be almost impossible to set up the graph, learn the parameters of the PDFs and run inference. If every node was directly connected to every other node, then the number of conditional dependences involved in seismic network analysis would make the problem intractable. Furthermore, since each node is only dependent on its parent nodes, it is possible to build up the graph slowly by only working on one small, independent piece of it at a time.

5.3 MAP world

A number of sets of different events and associations can be found to explain any given set of arrivals. A model like this, in which templates for the relationships between entities can be written out but the exact structure of the world is unknown, is called an *Open Universe Probabilistic Model* (OUPM, well described in [Milch and Russel 2010]). NET-VISA's job is to find the world that has the highest probability that is consistent with a given set of arrivals. This is called the *Maximum A Posteriori* (MAP) world. Once the MAP world has been found, its events become those of the VISA bulletin.

Practically, finding the MAP world consists of:

1. finding events,

2. associating arrivals with events, including figuring out the phase of the associated arrivals,

3. designating leftover arrivals as either false or coda.

Note again that the final assignment of location, depth and magnitude for an event in the VISA bulletin is done using standard IDC tools based on NET-VISA associations.

5.4 Event and arrival scores

Given data, the probability of a world is the joint probability of everything in it: events, associated arrivals, false and coda arrivals. The total probability of a world can be written

$$P(w) = P(\boldsymbol{e}|\boldsymbol{\Lambda})P(\boldsymbol{\xi})P(\boldsymbol{\kappa}).$$

$\boldsymbol{e}$ is the set of all of the events, $\boldsymbol{\Lambda}$is the set of all of the arrivals associated with them, $\boldsymbol{\xi}$ is the set of all false arrivals in the world and $\boldsymbol{\kappa}$is the set of all coda arrivals. (Boldface indicates a set.)

Invoking Bayes' rule, the posterior probability of the events is proportional to their prior probability times the likelihood of associated arrivals

$$P(\boldsymbol{e}|\boldsymbol{\Lambda}) \propto P(\boldsymbol{e})P(\boldsymbol{\Lambda}|\boldsymbol{e}).$$

Combining terms,

$$P(w) \propto P(\mathbf{e})P(\boldsymbol{\Lambda}|\boldsymbol{e})P(\boldsymbol{\xi})P(\boldsymbol{\kappa}).$$

Because events are modeled as being independent of each other and arrivals are also independent of each other (given events or previous arrivals), maximizing the probability of the world comes down to maximizing the contribution to it from each arrival. Given an arrival that could with equal likelihood be associated with more than one event, then it should be associated to the event that has the highest prior, unless, of course, the arrival contributes even more to the world by being classified as false or coda. There is one more way in which an arrival might contribute to the overall probability of a world: an expected arrival might not be detected.

The NET-VISA IA works by first proposing a set of theoretical events. Then, for every arrival in a time window, it tries out the association of the arrival with every event giving it every possible phase label. Each time, a score is computed for the arrival. The score is the likelihood of the arrival times the probability that the event was detected divided by the probability that the arrival is either false or coda (whichever is higher) times the probability that the event was not detected,

$$S(a|e) = \frac{P_d(e)P(a|e)}{(1 - P_d(e))P(a = \xi \text{or } a = \kappa)}.$$

$P_d(e)$ is the probability that an event was detected. Once the scores have been computed for all possible combinations of arrivals and associations, the IA chooses the association that

has the highest score. If the score for that association is still less than one, it means that the arrival has a higher probability of being false or coda and is labeled as such.

Once the associations of arrivals to events has been made thus, then each event is perturbed, generating a new set of events with the same associations as the original events. NET-VISA computes a score for each of these events. The event score is proportional to the posterior probability of the event. It is the prior probability of the event times the product of the arrival score for all of its associations times the probability that the event was missed for every station and phase for which an association was not made. If there are J possible phases and K stations in the network, then the event score is

$$S(e) = P(e) \prod_{j=1}^{J} \prod_{k=1}^{K} \begin{cases} S(a^{jk}|e) & \text{if } a = \Lambda \\ (1 - P_d) & \text{otherwise} \end{cases} .$$

The IA chooses the event which has the highest score. It then iterates between making associations based on arrival scores and refining events based on event scores. If at any point an event score comes out less than one, then the event is removed and all of its associated arrivals are designated false or coda. They may, of course, get picked up by another event in the next round. Once the process has converged, then all of the events which have a score higher than a tunable threshold are output to the VISA bulletin. It is only through the event score that Bayes' rule is invoked in NET-VISA. That is because

$$S(e) \propto P(e)S(\boldsymbol{a}|e) \propto P(e)P(\boldsymbol{\Lambda}|e)$$

The same math is presented more rigorously in Section 7.2).

5.5 Training

A user interacts with the GM through *training*. Although it is assumed that the math of the GM will not change over time, parameters of the PDFs do change since they model many time-dependent factors such as station-specific noise models, detection models, and an ever-changing set of primary and auxiliary stations in the IMS array. This means that the system needs to be periodically retrained with recent data.

The original training and testing of NET-VISA was done using a Baseline Data Set (BDS) made available to the project through the CTBTO's Virtual Data Exploitation Centre (vDec, [Vaidya et al, 2009]). The dataset includes three months of typical IDC data (March 22 to June 20, 2009). The three months were divided into 7 days of validation data (March 22 to March 29), and 2.5 months of training data. The analyst curated Late Events Bulletin (LEB) is used as ground truth.

A retraining schedule is suggested in Section 9

5.5.1 New stations

One of the input parameters into IDCNETVISA.LEARN is a network name. Using the AFFILIATION table of the IDC database, IDCNETVISA determines which stations are in the network and learns a model accordingly. As long as this model is used, data from any stations which were not represented in the training set is ignored. IDCNETVISA needs to be retrained before

it will recognize a new station. NET-VISA uses a hierarchical model to learn station-specific PDFs. If there is not enough data in a training set to learn the details of a station's performance such as its false detection rate, average values across the network are used. This means that, even in the absence of long-term training data, arrivals detected at new stations are able to contribute to NET-VISA events.

5.5.2 Modifications to internal models which do not affect detections

If modifications are made to the way in which internal models such as earth models or SASC corrections are made which do not affect the number of detections made or the events in LEB (although their location and magnitude might be changed), NET-VISA will have to be retrained to reflect these changes. This can be done by applying the new model to historic data and retraining.

5.5.3 Fundamental changes

If more fundamental changes are made, particularly ones which change the detections or produce more LEB events or events with different associations, then retraining NET-VISA on historic data will not be sufficient to capture the effect of the changes. Its performance is expected to decline, but not degrade entirely. Presumably, there will be more work for analysts for a while to correct NET-VISA until enough new data is amassed to retrain the system. It is advised that such changes be introduced slowly to minimize analyst effort while the system is being upgraded.

5.6 Inputs and outputs

The input into NET-VISA is a list of arrivals which have attributes of time, amplitude, azimuth, slowness and phase label. It is assumed that SASC corrections have been applied. The output is a list of events and associations arranged as a seismic bulletin. Events have attributes of time, location, depth and magnitude. Arrivals which are not associated with events are given a label of either *false* or *coda*. NET-VISA does not attempt to correct or re-compute any attributes of arrivals although it may change the phase of an arrival during the association process.

IDCNETVISA takes its input directly from the IDC database and outputs to the VISA bulletin which is structurally equivalent to SEL3. Database schema are in Appendix 12 of this document.

5.7 General program structure

Figure 2 gives an outline of the IDCNETVISA program structure. The basic elements of the package are

1. a README file which gives up-to-date instructions for installing and running the program. The install instructions are tailored for IDC.

2. Installation tools. For example, SETUP.PY installs and builds the NET-VISA programs. These files are described in detail in Part III.

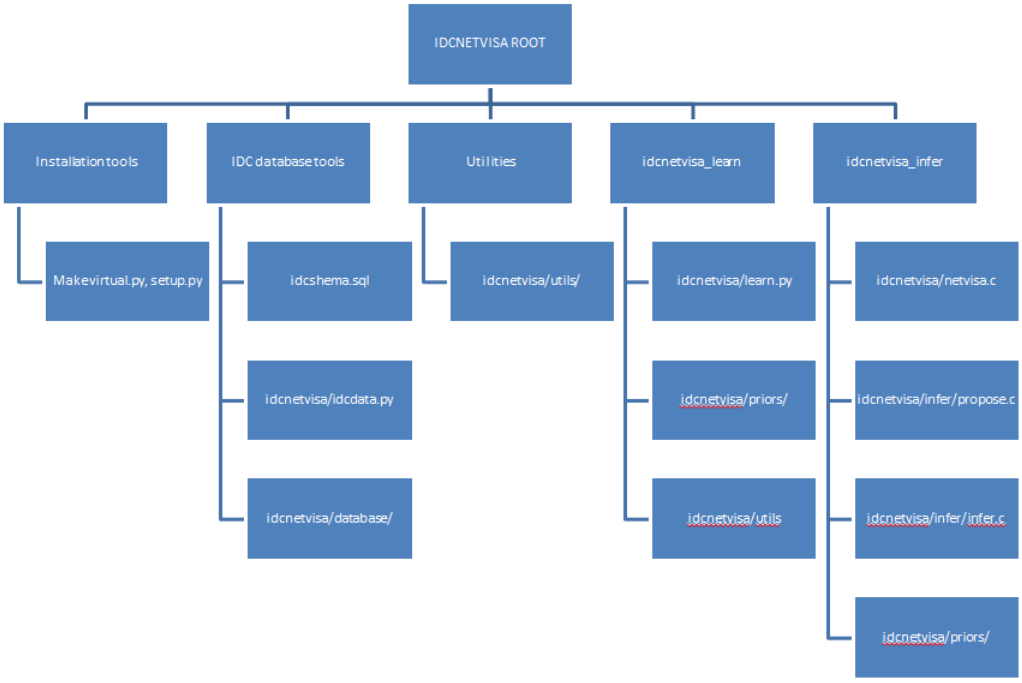


Figure 2 – IDCNETVISA structure

3. Database tools convert information in the IDC Oracle database to appropriate PYTHON inputs and objects and control how the net-visa bulletin is written back to an IDC database table.

4. Visualization tools in IDCNETVISA/UTILS/ are called on by IDCNETVISA_LEARN to make various plots, including plots of probability distributions such the ones displayed in Section 6.

5. IDCNETVISA_LEARN is the training script used to find parameters for a model. Details of each of the conditional distributions of the GM along with instructions for training are stored in the /IDCNETVISA/PRIORS/ directory. Subroutines for training are in /IDCNETVISA/UTILS.

6. IDCNETVISA_INFER is the inference script which runs the Inference Algorithm. It relies on /IDCNETVISA/infer/propose.c to propose events and IDCNETVISA/infer/infer.c to search for the world of highest probability.

In the following descriptions, the path to the appropriate IDCNETVISA file which contains the algorithm described in the section is noted in SMALL CAPITAL text style underneath the section header.

6 Generative Model

[IDCNETVISA_LEARN is used to train the GM. The probability distributions for the GM as well as instructions for training individual components are given in a series of files in the directory /IDCNETVISA/PRIORS/. Training instructions are specified in PYTHON. After training, a MODEL file will be generated and stored in a user-designated location. The PDFs are implemented in C files of the same name which are used by the IA.]

The Generative Model (GM) includes all of the information necessary to compute the probability of a hypothetical world. At the highest level it contains instructions for computing the following:

- the probability of an event,
- the probability of the set of arrivals associated to that event,
- the probability of a false arrival,
- the probability of a coda arrival.

The probability of a world is the joint probability of all of the entities in it, including events, arrivals associated with events, false arrivals and coda arrivals.

6.1 Mathematical conventions

The following conventions are followed.

1. A single event is e.
2. An arrival is A until it has been associated or designated false or coda.
3. An associated arrival is Λ,
4. If an arrival is predicted by an event but not detected, then the value of Λ is set to null, denoted by $\Lambda = \varsigma$.
5. a false arrival is ξ,
6. a coda arrival is κ.
7. A list of events or arrivals is denoted in boldface: $\boldsymbol{e}, \boldsymbol{A}, \boldsymbol{\Lambda}, \boldsymbol{\xi}, \boldsymbol{\kappa}$.
8. Subscripts indicate attributes. For example the time of an event is e_t.
9. Superscripts are used for indexing when absolutely necessary. i is for events, j is for phase, k is for station. For example Λ_t^{ijk} indicates the time attribute of the arrival from the i$^{\text{th}}$ event, with j$^{\text{th}}$ phase measured at station k.
10. Δ_{ik} is the angular distance between an event i and station k.

11. $P(\cdot)$ is used for a PDF.

12. T is an arbitrary time window

Events have the following attributes denoted by the following subscripts:

1. time, t

2. location (latitude, longitude), l

3. depth (km), d

4. magnitude ($\mathrm{m_b}$) , m

Note that only events of $\mathrm{m_b}$ 2 and above are modeled. All others are considered noise.
Arrivals have the following attributes:

1. time of arrival t

2. amplitude, a

3. azimuth (deg), z

4. slowness (sec/deg), s

5. phase label, h

A full list of symbols is given in Appendix B.

6.2 Probability of attributes of events and arrivals

The building blocks for the GM are a set of three graphs summarized in Figure 3. The graphs give the probability of the attributes of an arrival for each of three cases:

1. Case 1: The arrival is associated with an event and its attributes are dependent on the attributes of the event.

2. Case 2: The arrival is false in which case its attributes are independent.

3. Case 3: The arrival is a coda arrival with attributes that are dependent on the attributes of a previous arrival (which might be associated, false or coda).

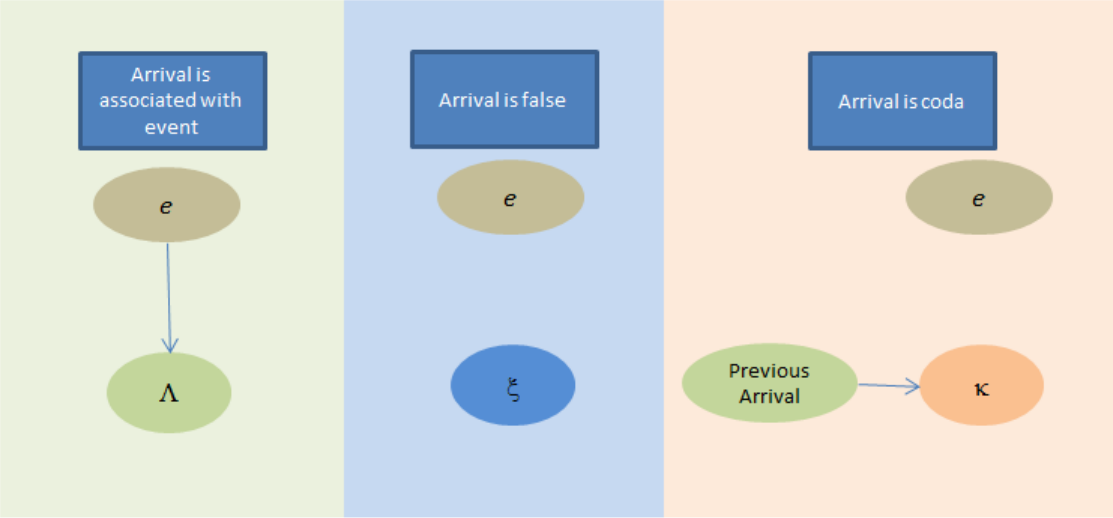


Figure 3 – Relationship between an event and an arrival for three possible cases: 1) The arrival is associated with event in which case the attributes of the arrival depend on the attributes of the event. 2) The arrival is false (or associated with another event) in which case its attributes are independent of everything else in the graph. 3) The arrival is coda in which case its attributes depend on the attributes of the previous arrival.

This section gives the instructions that are in each of the nodes in the graphs in previous sections and training instructions if necessary. The actual instructions are contained in a set of subroutines which are marked under the section headers. The nodes are described starting from the "ground up", that is, nodes which do not have any other nodes pointing to them are described first.

6.2.1 Relationship between an event and an associated arrival

The graph in Figure 4 shows the complete relationship of an associated arrival to an event. Solid lines indicate that nodes are connected via a deterministic relationship. For example, the predicted time of an arrival is directly computed from the time, location and depth of the event and the phase of the arrival (using the IASPEI earth model). Nodes that are connected with dotted lines are related via conditional PDFs. Nodes which have no arrow pointing to them have a PDF which is not conditional on any other node. Although the structure of the graph is the same for each station and arrival phase, the parameters of the PDFs are different so training must be done separately for each.

It is assumed that station-specific Slowness Azimuth Site Corrections (SASC) have already been applied to the attributes during station processing.

6.2.2 Probability of the attributes of an event

6.2.2.1 Event Medium The event medium can be one of rock, water, or air. The rock medium refers to seismic events, i.e. those that originate in the earth's crust. Water refers

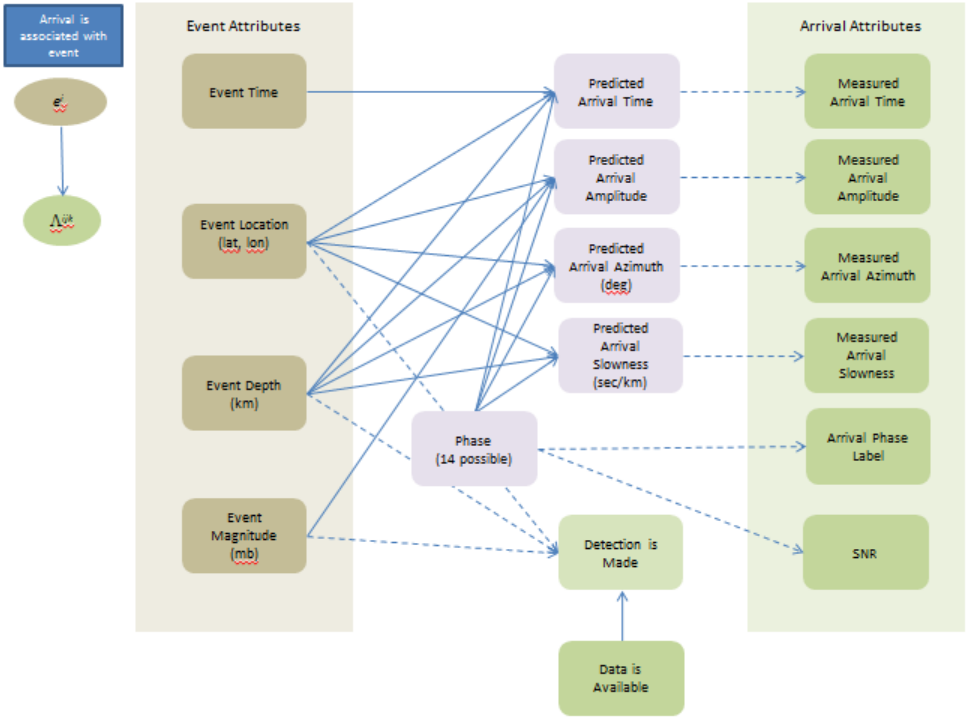


Figure 4 – GM graph for an arrival which is associated with an event. The nodes which are connected by solid lines have a deterministic relationship with each other (in this case derived from the IASPEI earth model). Nodes connected with dotted lines have a probabilistic relationship.

to hydro-acoustic events that occur in the oceans. Note that if an underwater explosion generates seismic waves the event medium would still be considered water. Finally, the air medium refers to infrasound events that occur in the atmosphere.

6.2.2.2 Rate of Events The prior rate of events depends on the medium. This rate is learned empirically from the LEB training data. However, we require a minimum prior rate of 1 event per hour. Typical rates for seismic events is 6 per hour while for hydroacoustic events, the rates are low enough that we use the minimum rate of 1 per hour. For infrasound events, the rate of events is artificially inflated to e^{20}. The rationale behind this *prior inflation* is that the available data on infrasound wave propagation combined with the sparsity of the network makes it hard to separate real infrasound arrivals from noise.

6.2.2.3 Event Time The time of an event is uniformly distributed between 0 and T.

$$P_t(e_t) = \frac{1}{T}$$

Note that, although the event can occur with equal probability any time during the interval T, there is a typical rate at which events tend to occur. It is built into the GM at a later point when the probability of a set of events is evaluated using a poisson distribution with a rate parameter. This is given in Section 6.3.2.1.

6.2.2.4 Event Location

[/IDCNETVISA/PRIORS/EVENTLOCATIONPRIOR]

The probability of the location of an event depends on the event medium. Events in the atmosphere can occur uniformly over the earth's surface, and events in water can occur uniformly in the earth's oceans. Seismic events on the other hand have a prior location that is influenced by the tectonic geography of the Earth. In NET-VISA, the PDF for seismic event location was constructed by fitting a kernel density function to a dataset of 10 years of known natural seismic events obtained from ISC (International Seismological Center). A kernel density function smudges the effect of an earthquake to a spatial region controlled by the kernel *bandwidth*. In addition to this natural seismicity prior a uniform probability was added to the result to allow for explosions occurring anywhere in the globe. Further the location density for seismic events was trained conditional on the event depth. Figure 6 shows this conditional distribution at selected depths.

The equation for the kernel density function $K_{b,x}(\cdot)$ defined with respect to the location x of a historical event is as follows.

$$K_{b,x}(y) = \frac{1 + 1/b^2}{2\pi R^2} \frac{\exp\left(-\Delta_{xy}/b\right)}{1 + \exp\left(-\pi/b\right)} .$$

Here R is the earth's radius, $b > 0$ is the bandwidth of the kernel and Δ_{xy} is the great circle distance between locations x and y. Both b and Δ_{xy} are expressed in radians. Notice that the kernel integrates to 1 over the surface of the earth, and in the limit as $b \to \infty$, the kernel tends to the uniform density of $\frac{1}{4\pi R^2}$. The optimal value of b is chosen by leave-one-out cross

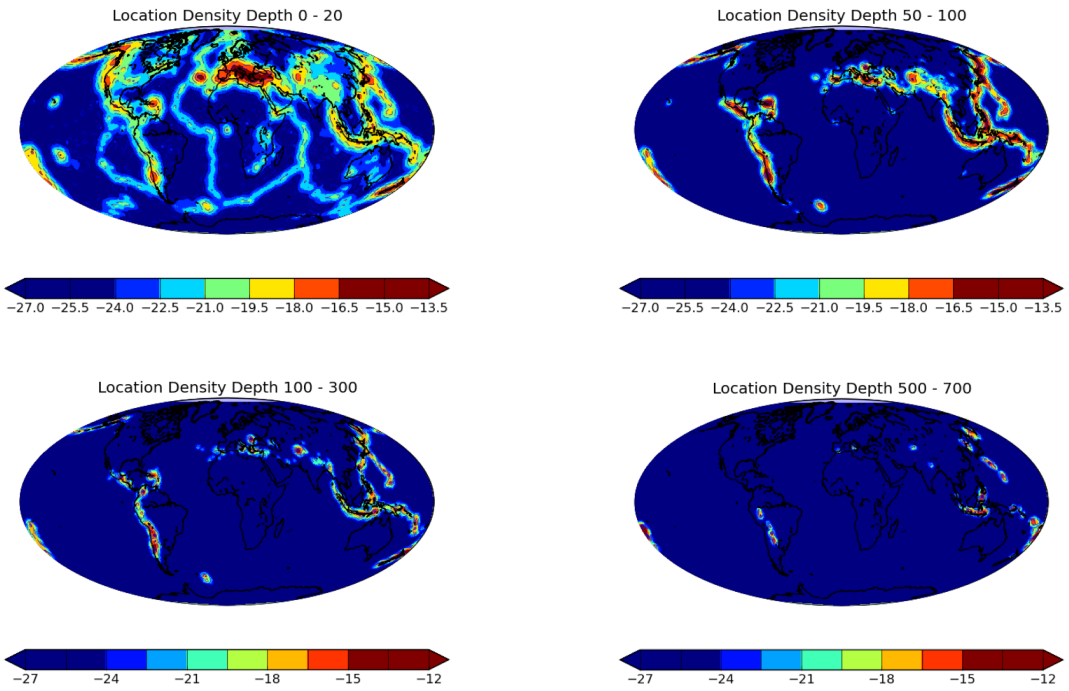


Figure 5 – Event location distribution conditional on depth.

validation (LOOCV) over a random subset of 1000 event locations in the training dataset. Grid search is used to find the value of b within 0 and 2 which maximizes the LOOCV value. An initial grid size of 0.2 was used and then refined with a grid size of 0.05. The best value of b is 0.7.

Note that due to the singular mission of IDC which is to monitor for events anywhere on the globe without geographical bias, it may be undesirable to let the location of an event affect its probability. If this is the case, a uniform distribution can be substituted for Event Location. Instructions for setting a switch that controls this are given in Section 5.5.

6.2.2.5 Event Depth For hydro-acoustic and infrasound events the event depth is fixed to zero. However, for seismic events the depth prior is trained from ISC data up to a maximum depth D which was set at 700km. See Figure 6 for the learned distirubtion.

6.2.2.6 Event Magnitude

[/IDCNETVISA/PRIORS/EVENTMAGPRIOR]

The Gutenberg-Richter law [Gutenber and Richter, 1954] states that the number of seismic events with magnitude m or more is ten times the number of events with magnitude $m+1$ or more. Therefore, the probability of the magnitude of an arbitrary event is $P_m(e_m >= m) = 10P_m(e_m >= m+1)$. This prior knowledge is represented with an exponential distribution with rate $\lambda_m = log(10)$:

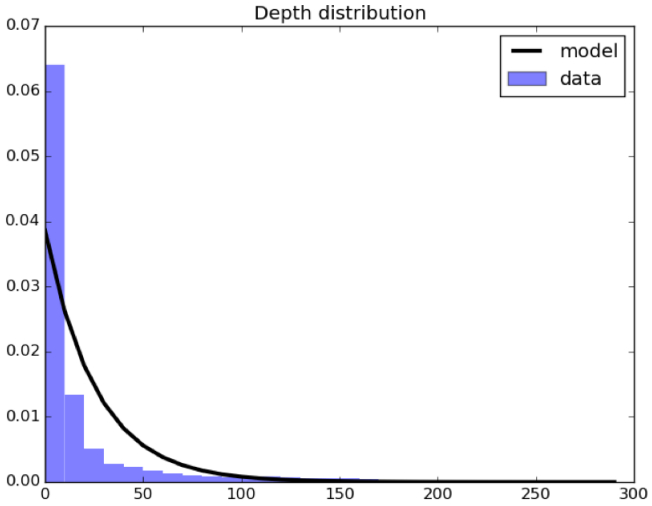


Figure 6 – Seismic event depth distribution.

$$P_m(e_m) = \lambda_m \exp(-\lambda_m \cdot (e_m - 2)).$$

Atmospheric events have no magnitude term, and in-water events have a separate energy term. Figure 7 shows the distribution modeled as a Cauchy distribution learned from historical data. The figure also shows a possible Gaussian model, which was rejected because its tails are not as wide as the Cauchy. The final model is a static Cauchy prior with location 70 and scale 10. As indicated in the figure the energy is measured in units of dB re Joules per meter square per Hz.

Additionally, for hydro-acoustic events a fixed seismic magnitude of 2.0 is assigned by default. Thus, for example, the probability of detection of a hydro-acoustic event at a seismic station would be the same as a seismic event with magnitude 2.0 on the ocean floor at the location of the hydro-acoustic event.

6.2.3 Probability of the attributes of an associated arrival given an event

6.2.3.1 Predicted Arrival Time, Amplitude, Azimuth, Slowness

[/IDCNETVISA/PRIORS/EARTHMODEL]

The predicted arrival time, amplitude and slowness are computed for each of 14 phases using the IASEPI earth model. The predicted azimuth of the arrival is computed using a geographic function G_z which calculates the angle between the event and and the station.

For hydro-acoustic events, the predicted arrival time is based on monthly travel time tables while for infrasound events we use two possible sound speed values depending on whether the distance is less than 20 degrees or not, see Figure 12.

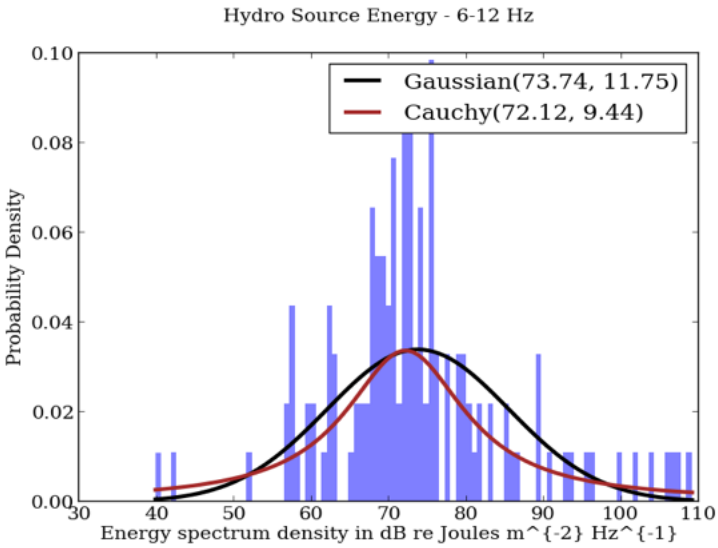


Figure 7 – Hydro-acoustic event source energy distribution.

6.2.3.2 Event Derived Feature (EDF) sets Many of the PDFs that follow are constructed from features that are derived from the attributes of individual events. An EDF is denoted $\mathcal{F}_{\mathrm{x}} \subseteq \mathcal{R}^3 \rightarrow \mathcal{R}$ where the subscript x indicates that the EDF is computed from attribute x. The features of the EDFs are listed in the sections along with the PDFs that they are used to compute. The PDFs are usually weighted functions of the features in the EDF set. The individual features are denoted by ϕ_w. The weights associated with them are denoted μ_x^{wjk}.

6.2.3.3 Data is Available

[/IDCNETVISA/IDCDATA.PY subroutine READDETECTIONS]

Obviously, if a primary IMS station is down, or no data has been requested from an auxiliary station, then arrivals (or the lack of expected arrivals) cannot be taken into account when estimating the probability of an event. After reading a list of detections from the database, NET-VISA performs a quick check to see which stations were operational. If there are no detections in a one hour time window which coincides with the Birth Move (Section B), then a station is considered down (or its data unrequested) and is ignored by NET-VISA.

6.2.3.4 Detection is Made

[/IDCNETVISA/PRIORS/EVENTDETECTIONPRIOR]

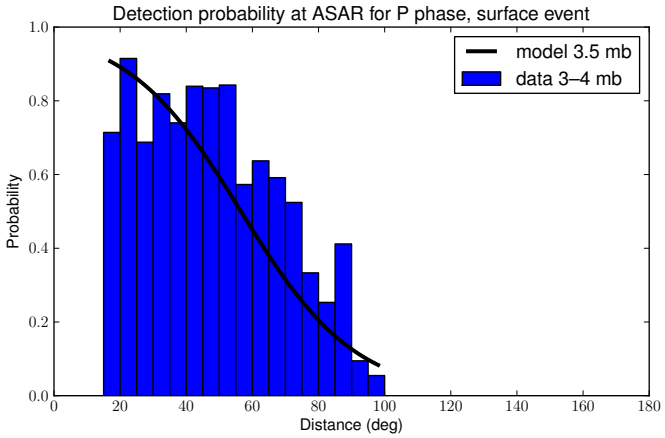


Figure 8 – Probability that a P phase detection is made at station ASAR for an event on the surface as a function of the distance from the event to the station.

The probability for a seismic event, $P_d(e)$, that a detection is made of a particular phase at a particular station depends on magnitude, depth, and distance to the station of the event that caused it. The log probability of detecting the arrival is modeled as a linear function of an EDF computed from e_m, e_d, and Δ_{ik}. The PDF is given by the formula:

$$log\left(\frac{P_d(e)}{1-P_d(e)}\right) = \sum_{w \in \mathcal{F}_d} \mu_d^{wjk} \cdot w(e_m, e_d, \Delta_{ik})$$

Where $F_d \subseteq \mathcal{R}^3 \rightarrow \mathcal{R}$ is a set of features, and μ_d^{wjk} is the weight for the feature w.

Of course, a detection may not be made. The probability of this happening is $1 - P_d\,(e)$.

***Train:** Directly estimating the feature weights μ_d^{wjk} is not possible since many of the station-phase combinations have very little data. To deal with this data sparsity a hierarchical Bayesian procedure [Gelman et al, 2004] is used which posits that the weight for a station-phase is drawn from a global prior for that phase. This global prior is in turn drawn from a non-informative prior, as follows:

$$\begin{aligned}
\mu_d^{wjk} &\sim \mathcal{N}(\mu_d^{wj}, \sigma_d^{wj}) \\
\mu_d^{wj} &\sim \mathcal{N}(0, 100) \\
(\sigma_d^{wj})^{-2} &\sim \Gamma(0.01, 100)
\end{aligned}$$

$\mathcal{N}$ and Γ are the Gaussian and the Gamma distributions parametrized by their mean and standard deviation, and shape and scale respectively. Estimation of parameters follows a coordinate ascent procedure. For each phase, the weights are initialized by setting $\mu_d^{wj} = 0$ and $\sigma_d^{wj} = 1$. Then n$\mu_d^{wjk} : \forall w, k$, $\mu_d^{wj} : \forall w$, and $\sigma_d^{wj} : \forall w$ alternately optimizes until convergence. In each maximization step, the optimal value of μ_d^{wjk} is computed by second order optimization, while the rest of the values have a closed form solution.

The above description for seismic events includes the detection of T phases at hydrophone stations. However, for hydro-acoustic events the detection probability at a hydrophone station depends on event source enegy e_m (see Figure 10) and whether there is a direct path

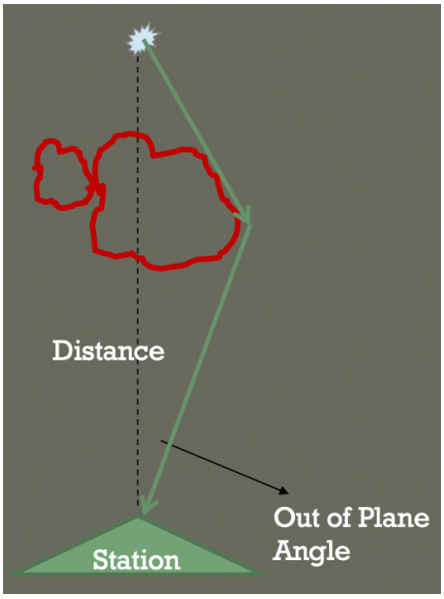


Figure 9 – Out of plane angle.

or not. We compute the out-of-plane angle,δ_{ik}, as the minimum amount in degrees that waves from the source i would have to bend to reach the station k. This angle is shown pictorially in Figure 9.

The detection probability is statically trained, and the resulting model is shown in Figure 10and the equation is as follows:

$$log(\frac{P_d(e)}{1-P_d(e)}) = -7.56 - \delta_{ij} + 0.07e_m$$

For the infrasound phase I, the detection probability is a function of distance only. This distribution is learned statically for both seismic source and infra sources

Measured Arrival Time:

[/IDCNETVISA/PRIORS/ARRIVALTIMEPRIOR]

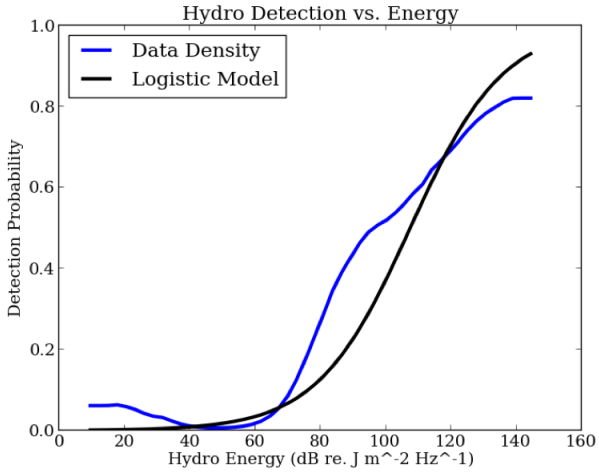


Figure 10 – Hydro-acoustic detection probability.

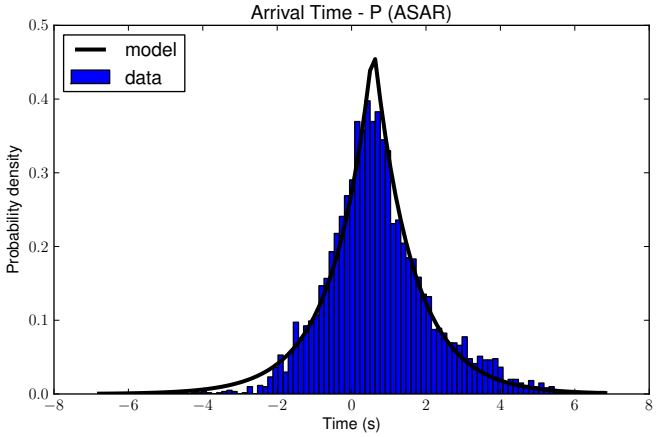


Figure 11 – PDF for P-phase arrival time residual at station ASAR.

The measured arrival time of a phase is the time of the event it is associated to, plus the ISAPEI predicted travel time I_t, plus a time residual r_t which accounts for inhomogeneities in the earth's structure resulting in seismic waves traveling faster or slower than the IASPEI prediction as well as for systematic errors in the picking of arrivals by the IDC system:

$$\Lambda_t = e_t + I_t(e_d, \Delta_{ik}) + r_t$$

As illustrated in Figure 11, the time residual is modeled as a Laplace distribution with mean μ_t and scale b_t. The probability of the arrival time given the time of the event it is associated to is thus:

$$P_t(\Lambda_t \mid e) = \frac{1}{2b_t} \exp\left(-\frac{|\Lambda_t - e_t - I_t(e, \Delta_{ik}) - \mu_t|}{b_t}\right).$$

***Train:** The estimation of all the station-and-phase-specific means and scales, μ_t and b_t, are based on a hierarchical model. In this model, each mean and scale is generated from

a phase-global prior, which is described as the following:

$$
\begin{aligned}
r_t &\sim L(\mu_t^{jk}, b_t^{jk}) \\
\mu_t^{jk} &\sim L(\mu_t^j, b_t^j) \\
(b_t^{jk})^{-1} &\sim \Gamma(1, \beta_t^j) \\
\mu_t^j &\sim L(0, 100) \\
(b_t^j)^{-1} &\sim \Gamma(.01, 100) \\
(\beta_t^j)^{-1} &\sim \Gamma(.01, 100)
\end{aligned}
$$

For each phase j, the mean and scale are initialized as $\mu_t^j = 0$, $b_t^j = 1$ and the rate of the inverse gamma $\beta_t^j = 1$. Similarly, for each station we initialize $\mu_t^{jk} = 0$, $b_t^{jk} = 1$. Next, the values of μ_t^{jk}, b_t^{jk} are iteratively optimized for each station, along with the global variables, μ_t^j, b_t^j, and β_t^j. This is repeated till convergence. Each of these optimization steps has a closed-form solution.

For hydro-acoustic events the arrival time is modeled as a Laplacian centered around the predicted arrival time with a fixed scale of 1 second. For infrasound, the predicted arrival time is also modeled as a Laplacian centered on the predicted arrival time, but the scale depends on whether or not the distance is less than 20 degrees as shown in Figure 13. This split reflects the discontinuity on the predicted infrasound arrival time at 20 degrees as shown in Figure 12.

6.2.3.5 Measured Arrival Amplitude

[/IDCNETVISA/PRIORS/ARRIVALAMPLITUDEPRIOR]

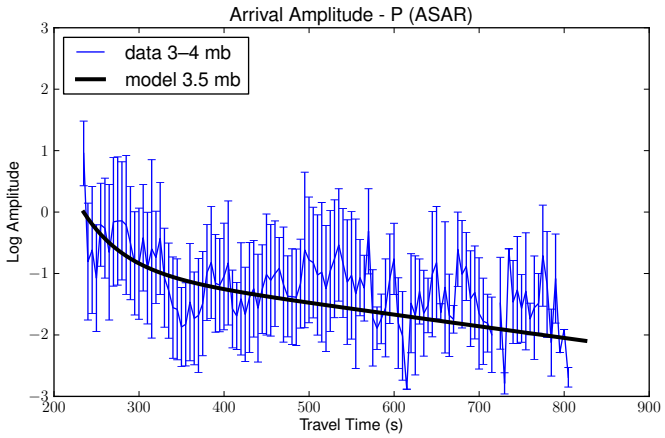


Figure 14 – Arrival amplitude for P phase at ASAR as a function of travel time for an event.

As shown in Figure 14, the the amplitude of an arrival depends on the travel time for a phase as well as the magnitude of the event it is associated with. Amplitude residuals are modeled as a linear combination of EDF features. The log amplitude residual, r_a, is modeled as a zero mean Gaussian PDF:

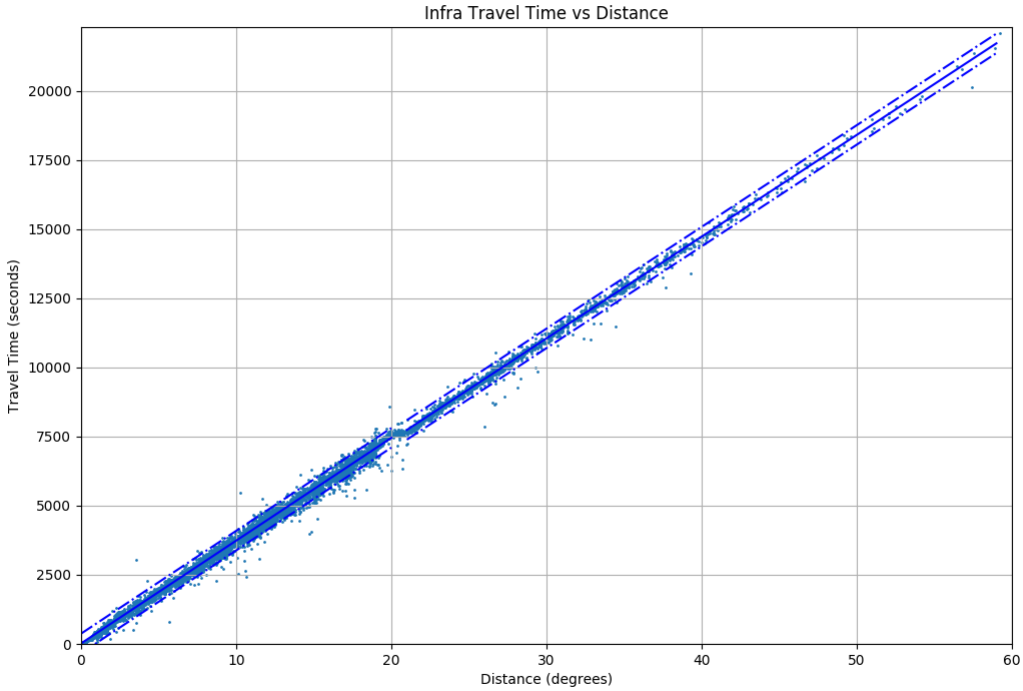


Figure 12 – Infrasound travel time curve

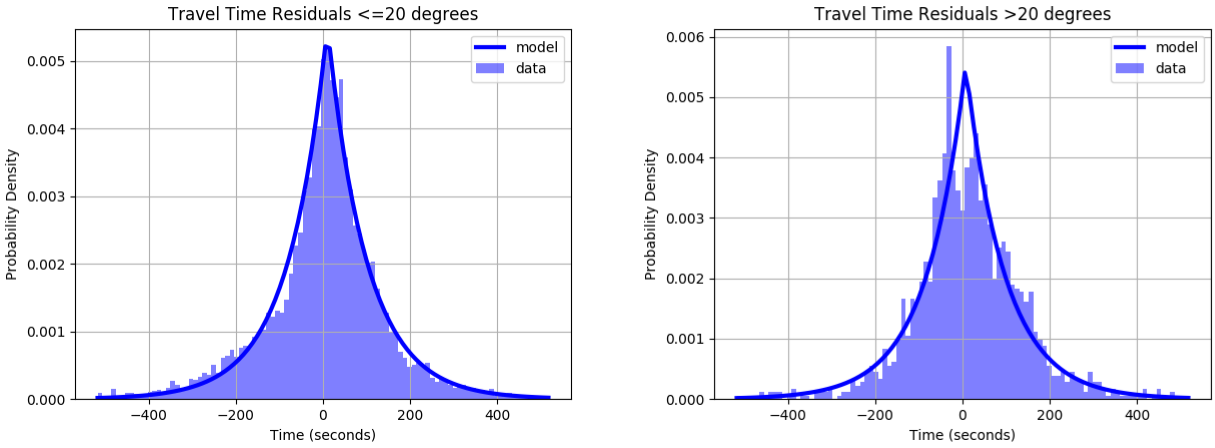


Figure 13 – Infrasound arrival time residual.

$$r_a = log(\Lambda_a) - \sum_{w \in \mathcal{F}_a} \mu_a^{wjk} \cdot w(e_m, e_d, I_T(e_d, \Delta_{ik})).$$

where

$$r_a \sim \mathcal{N}(0, \sigma_a)$$

This implies that

$$P_a\left(\Lambda_a | e\right) = \frac{1}{\sqrt{2\pi}\sigma_a} \exp\left(-\frac{(r_a)^2}{2{\sigma_a}^2}\right) \frac{1}{\Lambda_a}$$

***Train:** A hierarchical model is used to estimate the feature weights which assumes that they are drawn from a global prior:

$$\begin{aligned}
\mu_a^{wjk} &\sim \mathcal{N}(\mu_a^{wj}, \sigma_a^{wj}) \\
(\sigma_a^{jk})^{-2} &\sim \Gamma(100, \beta_a^j) \\
\mu_a^{wj} &\sim \mathcal{N}(0, 100) \\
(\sigma_a^{wj})^{-2} &\sim \Gamma(.01, 100) \\
(\beta_a^j)^{-1} &\sim \Gamma(.01, 100)
\end{aligned}$$

MAP inference is used to find the weights.
Note: infrasound and hydro-acoustic arrivals have no arrival amplitude.

6.2.3.6 Hydro-Acoustic Arrival Energy The total energy measured for the arrival in the 6-12Hz frequency band is modeled as a Gaussian distibution with the mean modeled as a function of distance and source energy. The hydro-acoustic intensity propagating from the event is reduced due to two main factors – absorption (AT), which is primarily due to the presence of magnesium sulfate and boric acid in the oceans, in addition to the effect of water viscosity, and geometric spreading (GS), which is initially spherical, and later cylindrical in the SOFAR channel. These losses are shown below:

$AT = \alpha \cdot range$

$GS = 20 \cdot \log_{10}(3000) + 10 \cdot \log_{10}(range\text{- }3000)$

Here range is in meters. The coefficient for the absorption loss is trained statically from data as shown in Figure 15 and the value chosen is $7 \cdot 10^{-7}$db per meter. The hydro-acoustic loss as a function of distance is shown in Figure 16. We model the arrival energy as a Gaussian distributiom centered around the estimated energy with a standard deviation of 5.

6.2.3.7 Hydro Fraction Time For hydro-acoustic arrivals we measure the fraction of time that the signal spends above the detection threshold versus the total duration of the signal. This is modeled as a beta distribution. This attribute is very helpful in discrimination between T and H arrivals. The former tend to spend less time above the threshold. Note that all hydro features are measured in the frequency band (6-12 Hz).

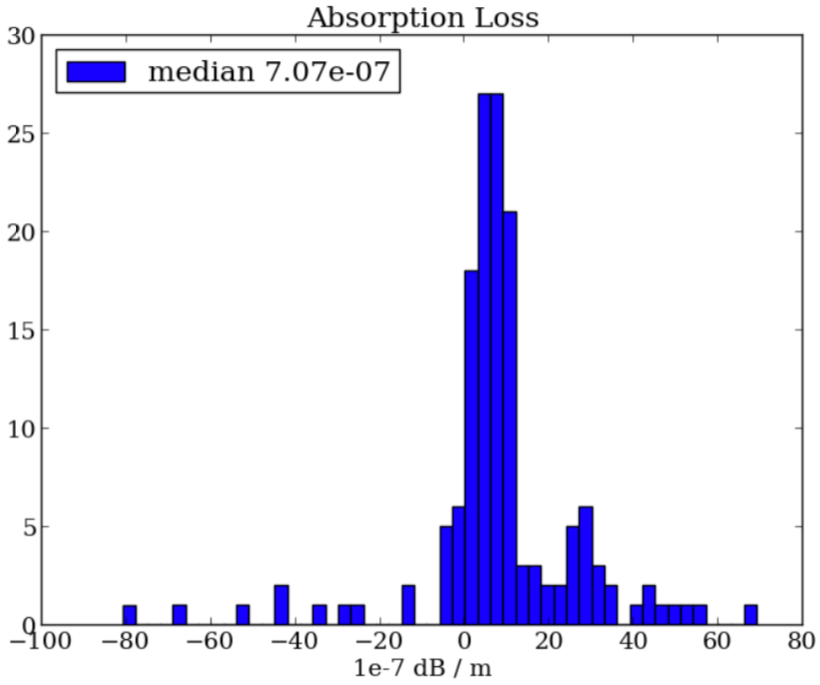


Figure 15 – Hydro-acoustic absorption loss estimate

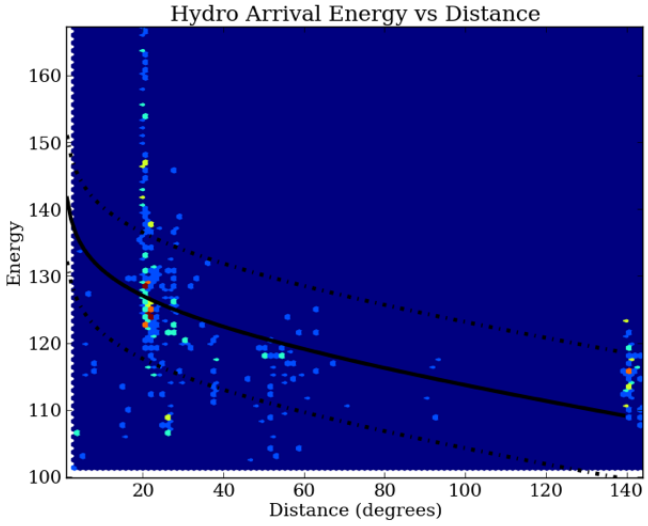


Figure 16 – Hydro-acoustic arrival energy as a function of distance

6.2.3.8 Measured Arrival Azimuth

[/IDCNETVISA/PRIORS/ARRIVALAZIMUTHPRIOR]

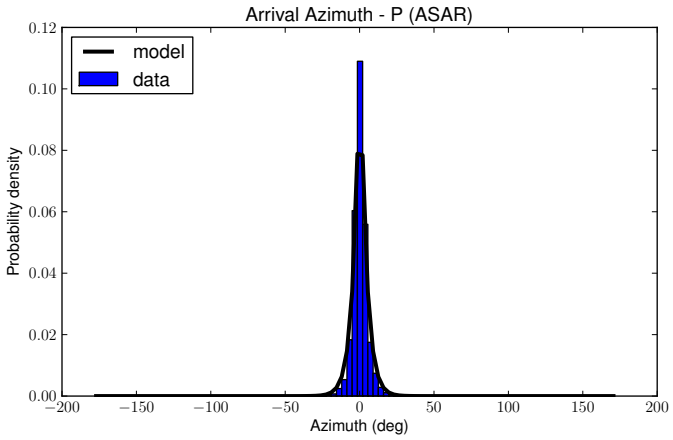


Figure 17 – Difference between predicted and measured azimuth for each phase and station is modeled as a Gaussian function. Here it is shown for P phase at station ASAR.

The PDF for Arrival Azimuth is computed the same way as the PDF for Arrival Time except that a geographical function G_z is used to predict the azimuth of an arrival instead of the IASPEI model. The geographical model depends on the station location, s_l as well as the event location. The function ψ computes the angular difference between the observed angle Λ_z^{ijk} and the predicted angle. The formula for the PDF is:

$$P_z(\Lambda_z \mid e) = \frac{1}{2b_z} \exp\left(-\frac{|\psi(\Lambda_z, G_z(s_l, e_l)) - \mu_s|}{b_z}\right).$$

An example is shown in Figure 17. (It is assumed that in the measured data, SASC corrections have already been applied).

***Train:** The training for arrival slowness is the same as the Training for Arrival Time.

For hydroacoustic arrivals the arrival time residual is fixed at 1 seconds and arrival azimuth residual at 1 degrees. For infrasound the azimuth residual is a function of the event medium as show in Figure 18. As shown in the figure, seismic sources tend to have higher azimuth residuals than infrasound sources.

6.2.3.9 Measured Arrival Slowness

[/IDCNETVISA/PRIORS/ARRIVALSLOWNESSPRIOR]

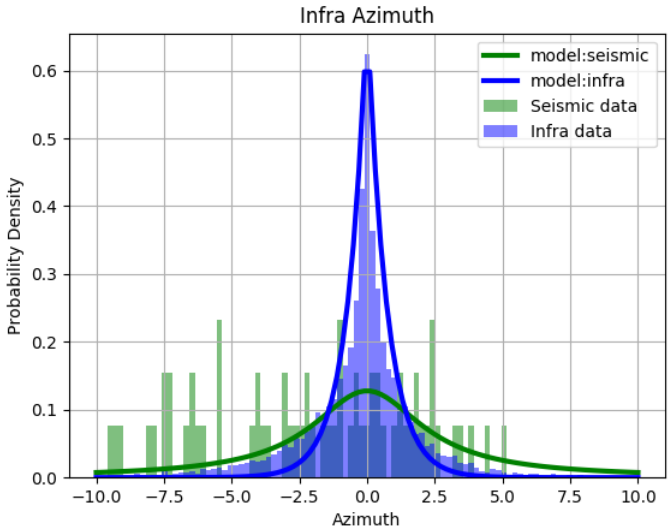


Figure 18 – Infrasound Azimuth Distribution

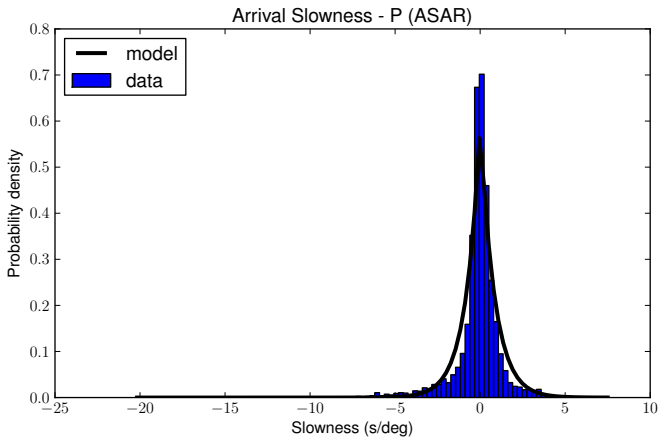


Figure 19 – Slowness residuals for an associated arrival are modeled using a Laplace distribution.

The PDF for Arrival Slowness is computed the same way as the PDF for arrival time using a Laplace distribution (Figure 19):

$$P_s(\Lambda_s \mid e) = \frac{1}{2b_s} \exp\left(-\frac{|\Lambda_s - e_s - I_s(e_d, \Delta_{ik}) - \mu_s|}{b_s}\right).$$

It is assumed that SASC have already been applied to the data.

***Train:** The training for arrival slowness is the same as the Training for Arrival Time.

For infarsound and hydro-acoustic stations the arrival slowness is modeled as a Laplacian around wave propagation speed for their respective mediums. Unlike seismic this slowness values doesn't depend on distance to event.

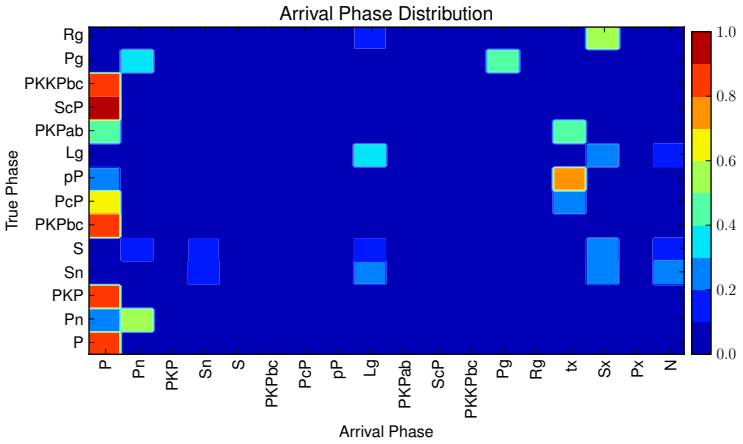


Figure 20 – PDF for associated arrival phase

6.2.3.10 Arrival Phase Label

[/IDCNETVISA/PRIORS/ARRIVALPHASEPRIOR]

The probability of the phase label of an arrival is modeled as a function only of the true phase of the arrival, not the event to which the phase is associated. The phase labels are distributed according to a multinomial distribution:

$$P_h(\Lambda_h \mid e) = p_h(\Lambda_h).$$

Figure 20 shows the PDF with which phase labels are assigned as a function of true phase. With high probability, a P phase will be correctly classified. PcP phases, however, are often miss-classified as tx.

***Train:** The multinomial distributions, p_h , are learned by counting the occurrence of various phase labels as a function of true phase and smoothing the result using "add-one" smoothing. Assuming that there are N examples of phases in the training set, the probability that a phase labeled h actually has true phase j is computed by counting the number of similarly mislabeled phases in the training set n_{hj} and dividing by the total number of examples in the set. The result is "smoothed" by adding one to the numerator and the total number of possible phases to the denominator thus:

$$p_h^j(\Lambda_h^{ijk}) = \frac{n_{hj} + 1}{N + 14}$$

Similar distributions are learned for infrasound and hydro-acoustic stations. Table 1 shows the matrix of all possible true phases and potentially observed phase labels that are allowed by the current model for all combinations of event and station mediums.

6.2.3.11 Infrasound Center Frequency Infrasound center frequency is modeled by a kernel density estimate.

Event Medium	Station Medium	Possible True Phases	Possible observed phases (iPhase)	Maximum distance (degrees)
Seismic, Hydro-acoustic	Seismic	P, Pn, PKP, Sn, S, PKPbc, PcP, pP, Lg, PKPab, ScP, PKKPbc, Pg, Rg	true phase and tx, Sx, Px, N	180
Seismic	Hydro-acoustic	T	true phase and H, N, Tx, Hx	180
Seismic	Infrasound	I	true phase and N	20
Hydro-acoustic	Hydro-acoustic	H	true phase and T, N, Tx, Hx	180
Hydro-acoustic	Infrasound	Not possible		
Infrasound	Seismic	Not possible		
Infrasound	Hydro-acoustic	Not possible		
Infrasound	Infrasound	I	only the true phase I is allowed	60

Table 1 – Possible event medium and station medium, phase combinations.

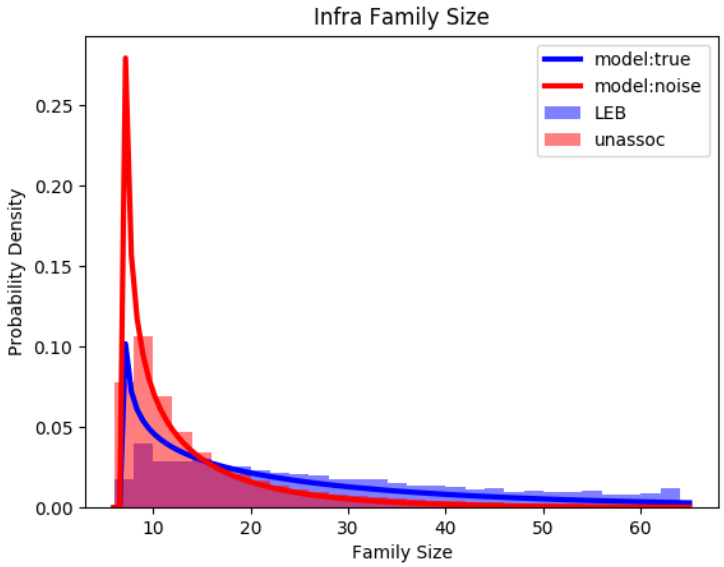


Figure 21 – Infrasound Family Size

6.2.3.12 Infrasound Family Size Infrasound family size, or the number of pixels in a PMCC detection is modeled as a kernel density estimate for both true and false arrivals as shown in Figure 21. This raw distribution is truncated below a family size of 12 for true arrivals.

6.2.3.13 Infrasound Consistency Infrasound consistency attribute is modeled as a kernel density estimate of the observed values as shown in Figure 22.

6.2.3.14 Infrasound Duration / Family Size The ratio of Infrasound duration, or the duration of the signal, to the infrasound family size is modeled as a Gamma distribution for both the real arrivals as well as the false arrivals as shown in Figure 23.

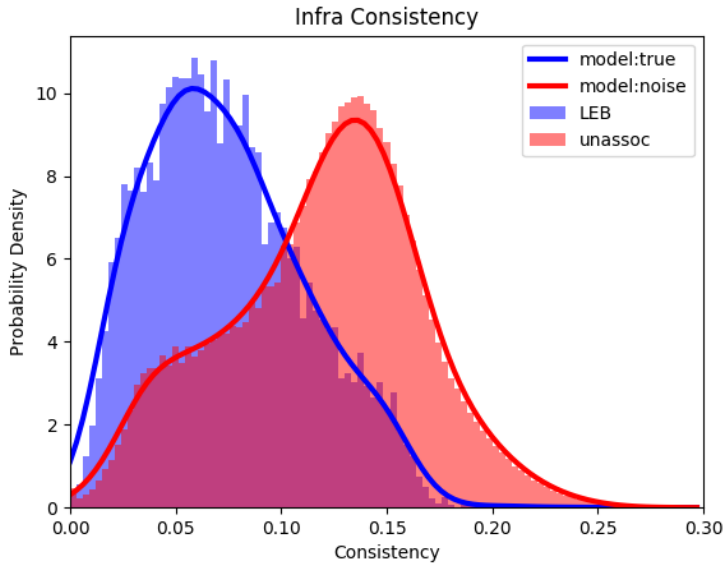


Figure 22 – Infrasound Consistency

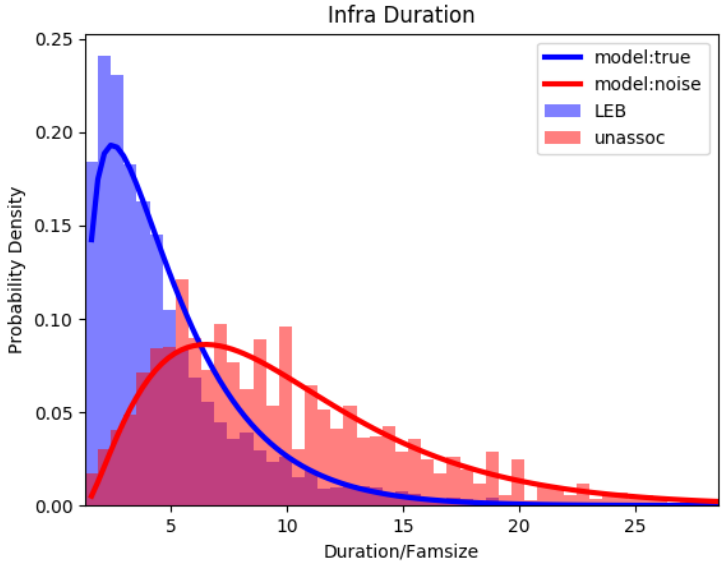


Figure 23 – Infrasound Duration

6.2.4 Probability of the attributes of false arrivals

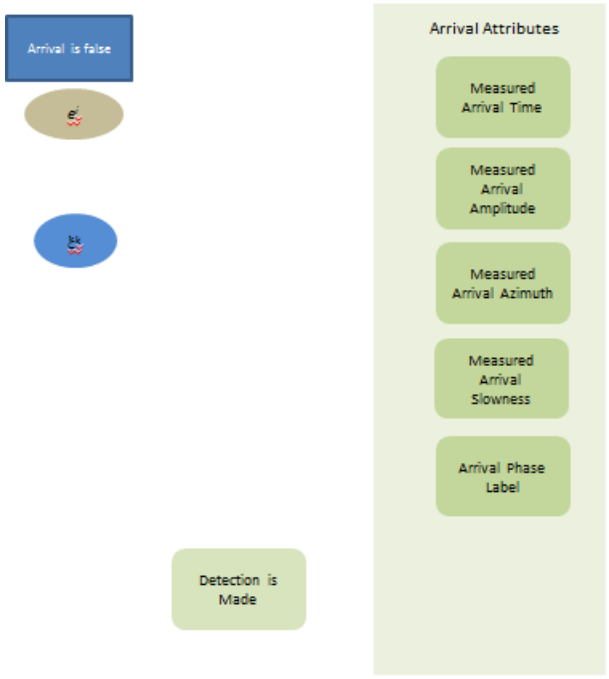


Figure 24 – The attributes of a False arrival are independent of everything else.

The graph giving the relationship between an event and an arrival is much simpler when the arrival is false or uncorrelated to the event, as shown in Figure 24. In this case, most of the arrival attributes, including the probability that a detection is made, have independent PDFs. Only the false arrival slowness distribution depends on the false arrival phase label. The false arrival slowness depends on the site medium. Further, the false detection rate is different at different stations, so the model must be trained separately for each station.

6.2.4.1 Number of false arrivals. For each station we compute a prior rate of false arrivals. Figures 25,27 , and 26 show this rate for seismic, hydro-acoustic and infrasound stations respectively.

6.2.4.2 Measured Arrival Time (false arrival) The time of a false arrival is drawn uniformly at random from range T:

$$P_t(\xi_t) = \frac{1}{T}.$$

Note that this means that a false arrival can occur any time at any station, but some stations have a much higher rate of false arrivals than others. The station-specific rate of false arrivals is taken into account when the score of an arrival is computed. The way that this is done is outlined in section 6.3.4.2.

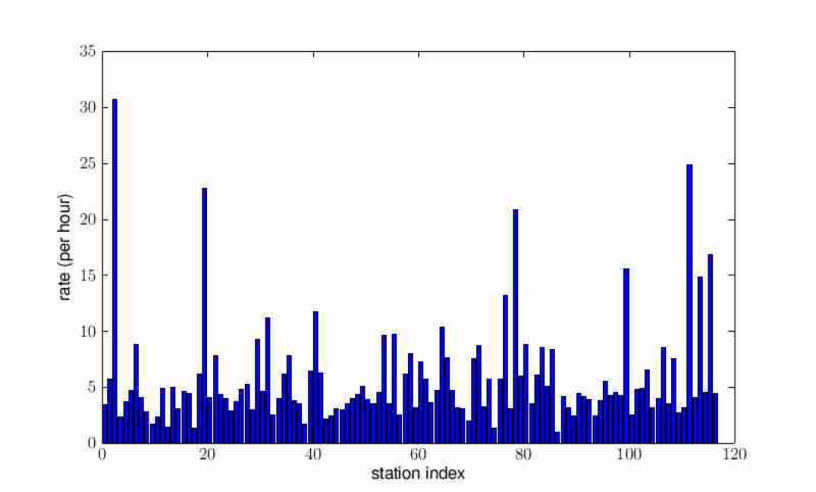


Figure 25 – Seismic stations false arrival rate.

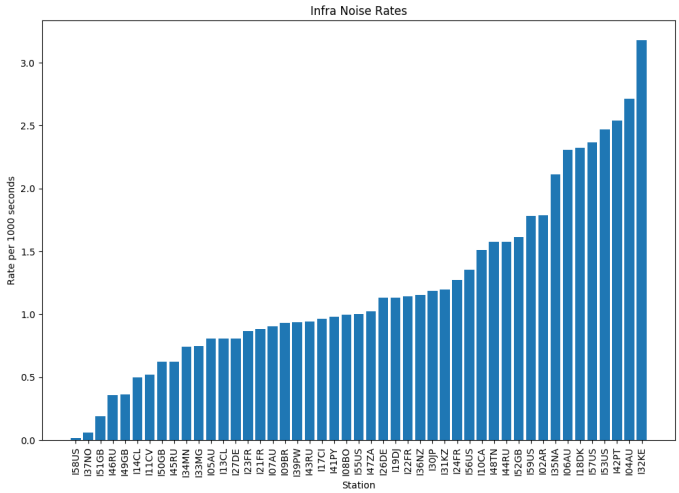


Figure 26 – Infrasound stations false arrival rate.

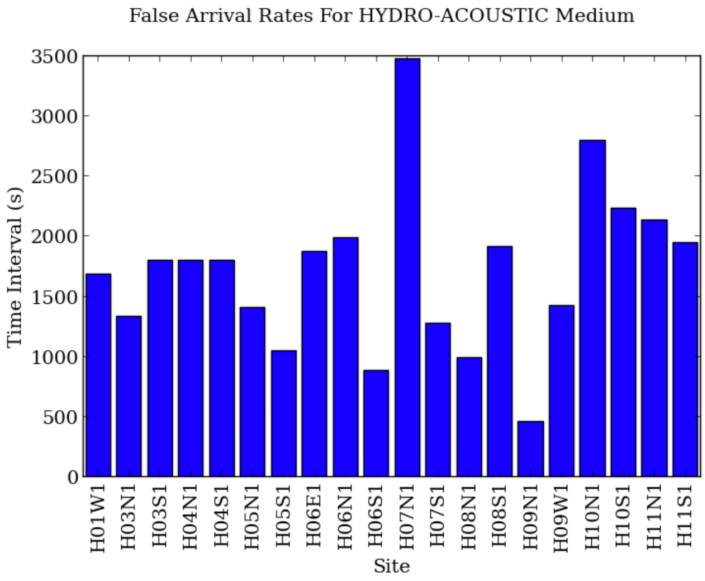


Figure 27 – Hydroacoustic false arrival interval

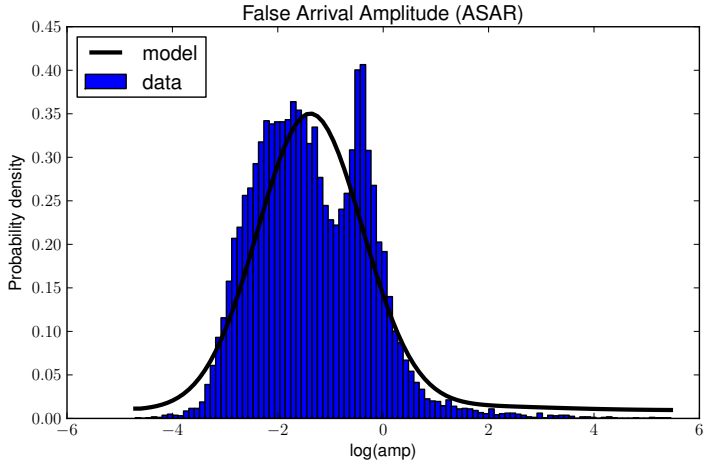


Figure 28 – The distribution of the amplitude of false arrivals from station ASAR.

6.2.4.3 Measured Arrival Amplitude (false arrival) Typically, the amplitude distribution of false arrivals at a given station has two peaks. The peak with higher amplitude most likely corresponds to detections of small (< 2 m_b) events. The lower peak is most likely station noise. The distribution is modeled as a mixture of two Gaussians as shown in Figure 28.

***Train:** The mixture model is found using a standard EM procedure with a flat uniform prior with probability 0.1

6.2.4.4 Measured Arrival Energy (false arrival) The arrival energy for false hydroacoustic arrivals are modeled as a Cauchy distribution as shown in Figure 29.

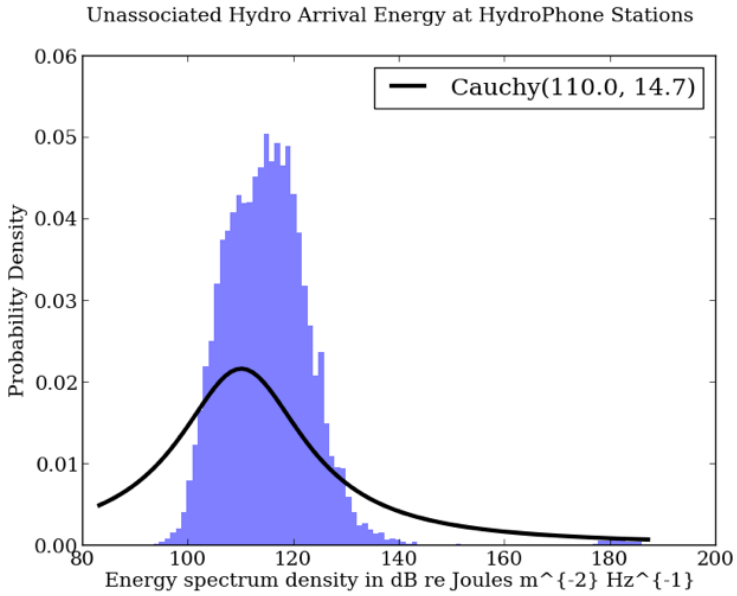


Figure 29 – Hydro-acoustic energy for false arrivals

6.2.4.5 Measured Arrival Azimuth (false arrival) The azimuth of a false event is drawn uniformly at random from range M_z:

$$P_z(\xi_z) = \frac{1}{M_z}$$

6.2.4.6 Measured Arrival Slowness (false arrival) The slowness of a false event is modeled as a Laplacian distribution trained on the data. For the seismic medium the scale of this Laplacian is multiplied by a factor of 10 to prevent it modeling the data too closely, as it is known that many of the unassociated arrivals are from real events. An Example $P_s(\xi_s)$ is provided in Figure 30. Figure 31shows the distribution of all false arrivals slowness values broken down by the iPhase label (or the label assigned by station processing).

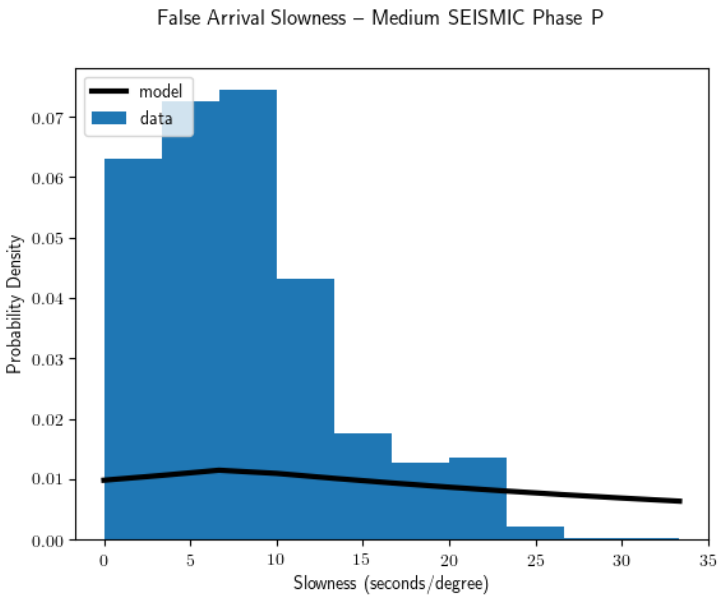


Figure 30 – The distribution of false arrival slowness values for seismic medium with phase label P.

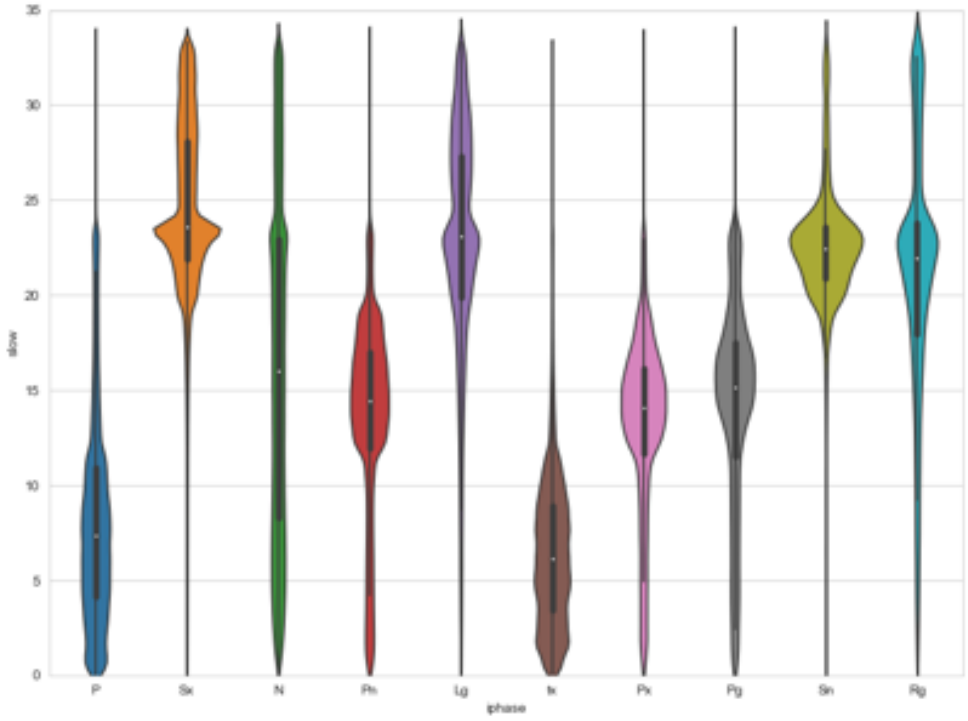


Figure 31 – Arrival slowness distribution given iPhase

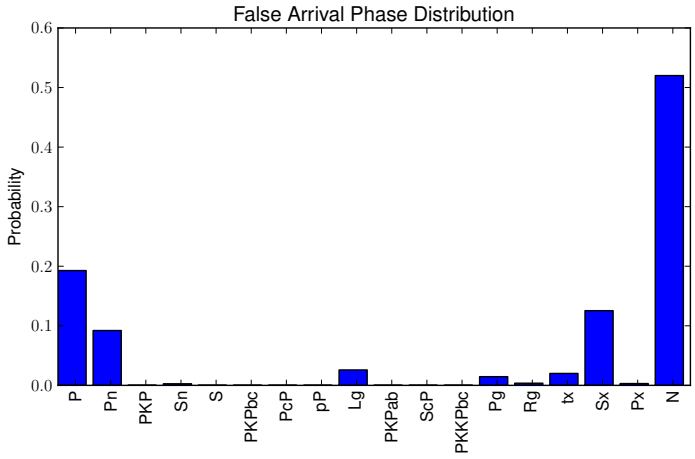


Figure 32 – Distribution of the phase labels of false arrivals.

6.2.4.7 Arrival Phase Label (false arrival) The phase label assigned to the false arrival follows a multinomial distribution which is illustrated in Figure 32. The distribution is the same for all stations.

$$P_h(\xi_h) = p_h(\xi_h).$$

***Train:** The phase label distributions are learned by add-one smoothing. Similar distributions are learned for hydro-acoustic and infrasounnnnd stations as well.

6.2.4.8 Infrasound Clutter Infrasound arrivals at a station have a separate distribution for some of their attributes. For each station we train a so called *clutter* model on the azimuth and center frequency of unassociated arrivals.Some examples of these are shown in Figure 33. This model captures local clutter sources such as oceanic noise and gas flares. For stations that don't have enough data (at least 1000 points over the training interval) to train a clutter model a global 1-d kernel density estimate is used for the center frequency and a uniform distribution is used for the azimuth of false arrivals.

6.2.4.9 Hydroacoustic Clutter Similar to the infrasound clutter model a clutter model is used for hydroacoustic sources as well. During each inference run a recent sample of azimuths for all H phases recorded at each station is used to train a clutter model. This clutter model uses a simple kernel density estimate around each arrival's azimuth with a standard deviation of 1 degree.

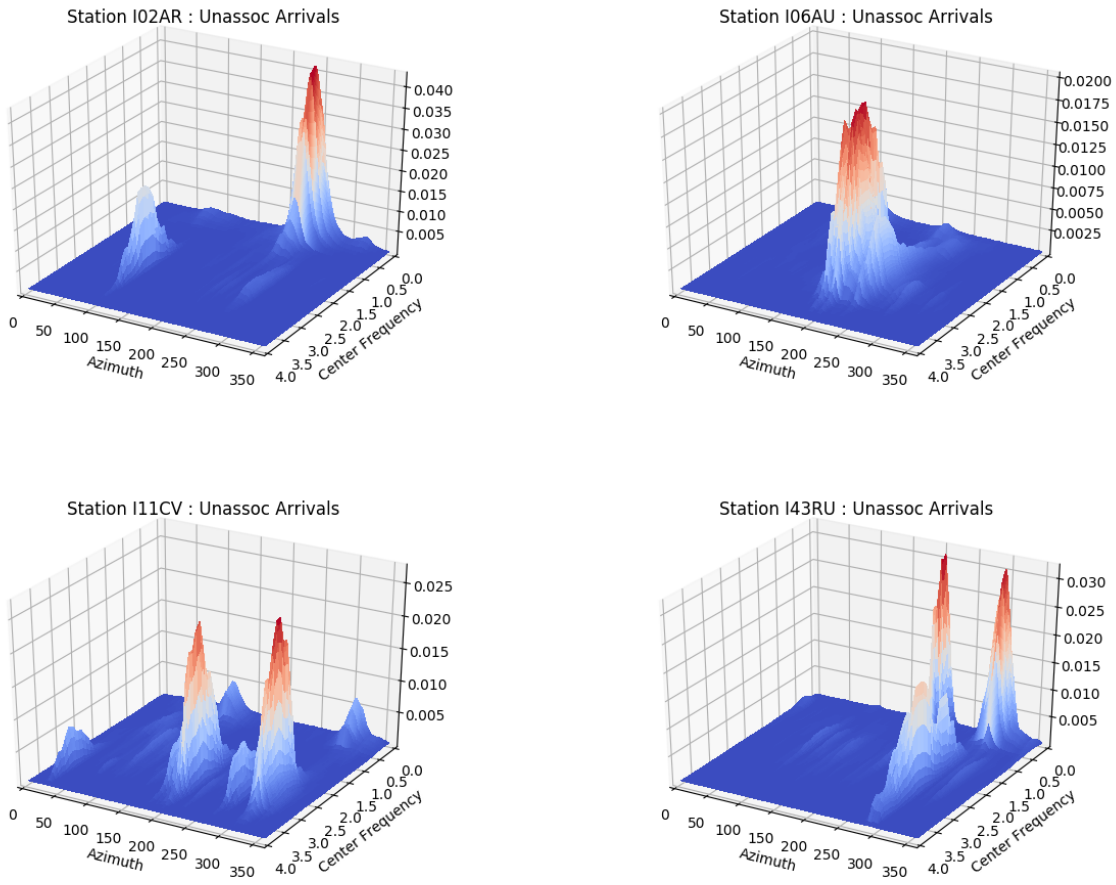


Figure 33 – Infrasound Clutter Examples

6.2.5 Probability of the attributes of coda arrivals

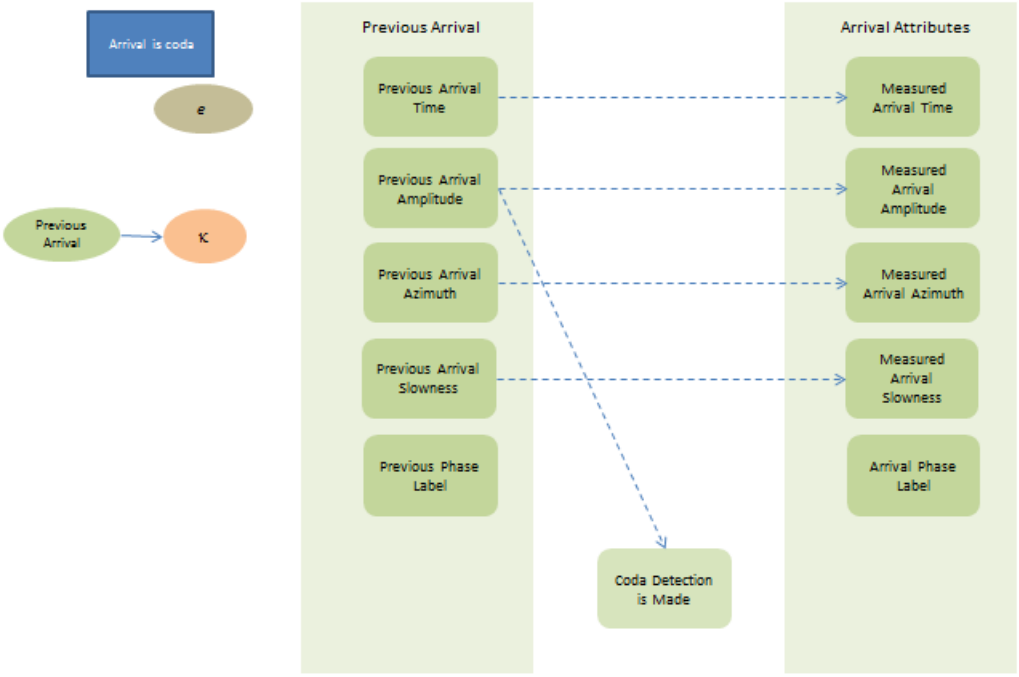


Figure 34 – The probability of a coda arrival is independent of any event, but depends on the arrival which precedes it.

Coda detections are spurious detections that tend to be made by the IDC system out of the scattered energy immediately following an arrival at a seismic station. The graph for coda arrivals is shown in Figure 34. In this case, the probability that a detection is made is a function only of the amplitude of the previous or *triggering* arrival. The rest of the attributes of the coda arrival are correlated to the corresponding attributes of the triggering arrival except for the phase label, which is independent. The coda model is the same for all stations, however it needs to be trained twice, once for coda arrivals which follow true arrivals, and once for coda arrivals following coda arrivals. The PDFs for coda attributes are modeled using the difference between the coda attribute and the corresponding attributes of its triggering arrival.

6.2.5.1 Determining which arrivals in SEL3 are coda arrivals to construct a coda training set One of the problems in modeling coda arrivals is that analysts do not mark and retain them – they simply delete them, so there is no ready-made list of coda and triggering arrivals which can be used to learn probability distributions. Instead, one had to be constructed heuristically. In order to construct such a list, LEB was searched for unassociated arrivals that occurred less than 30 seconds after another arrival with an azimuth and slowness that differed from the first arrival by less than 30 degrees or 10 seconds per degree respectively. The statistics of the coda model were assumed to be the same for all of the seismic stations in the IMS array.

6.2.5.2 Detection is made (coda arrival)

[/IDCNETVISA/PRIORS/SECDETPRIOR]

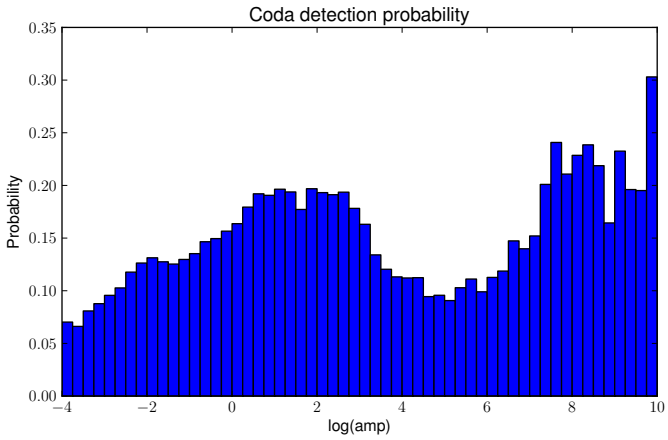


Figure 35 – Probability of a coda detection as a function of the log amplitude of its triggering arrival when the triggering arrival is not a coda arrival.

The arrival that precedes a coda detection is called a *triggering arrival* and denoted asA^q. The probability of detecting a coda arrival is $P_d\left(\kappa\right)$. It is a function of the amplitude of the triggering arrival and unrelated to whether the triggering arrival was a false or coda detection. The probability of detection of a coda arrival is thus:

$$P_d\left(\kappa\right) = P_d\left(A_a^q\right)$$

Note that the probability of detecting a coda arrival is independent of the station. If there is no previous arrival then $P_d(\kappa) = 0$.

As illustrated in Figure 35, the PDF for coda detection tends to have a sweet spot in the middle of its range. If the amplitude of the triggering arrival is low, then there is low probability of making coda detections. However, if the amplitude of the triggering arrival is too high, then the scattered energy after it lasts long enough to affect the Long Term Average (LTA) of energy at a station. Since detections are made using the ratio of the Short Term Average to LTA, when the LTA goes up, fewer overall detections are made. As the magnitude of both triggering arrivals and codas goes up, this effect is eventually overcome.

***Train:** $P_d\left(\kappa\right)$ is estimated using a non-parametric model: there is no mathematical form of the PDF. Instead, the PDF is estimated by making a histogram of the training data and computing the percentage of the total coda arrivals that are observed as a function of the log amplitude of a triggering arrival. The log amplitude of triggering arrivals is discretizing into buckets of size .25 that go between -4 and 10. Any points outside these extreme values are mapped to the nearest bucket.

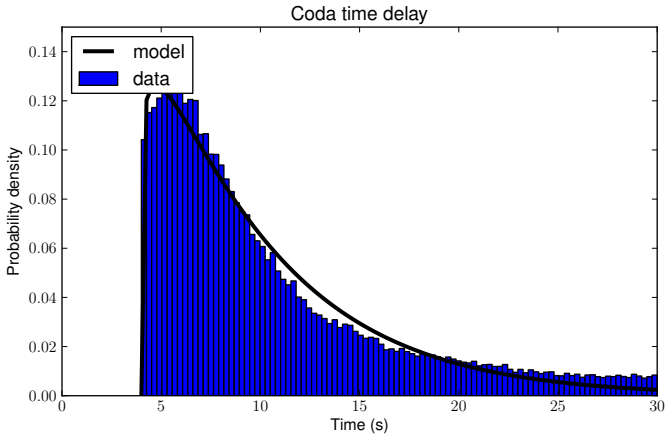


Figure 36 – Coda arrival time as a function of triggering arrival time is modeled as a Gamma distribution.

6.2.5.3 Measured Arrival Time (coda arrival) The distribution of the time delay between the coda arrival and the triggering arrival is modeled as a Gamma distribution as shown in Figure 36:

$$P_t(\kappa_t \mid A_t^q) = \frac{1}{\Gamma(\rho_t)\nu_t^{\rho_t}}(\kappa_t - A_t^q)^{\rho_t - 1} \exp(-\frac{\kappa_t - A_t^q}{\nu_t}).$$

ρ and ν are the shape and rate of the distribution.

***Train:** The parameters of the gamma distribution are determined by Maximum Likelihood Estimation (MLE) using the coda training set.

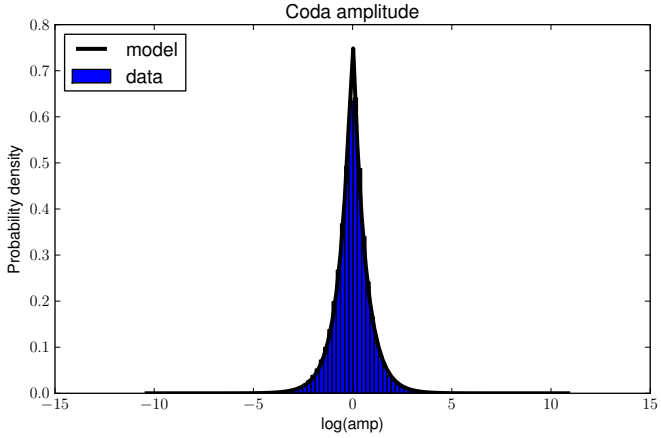


Figure 37 – Measured arrival amplitude for coda arrivals

6.2.5.4 Measured Arrival Amplitude (coda arrival) The distribution for the difference in log amplitude of the coda arrival vs. that of its triggering arrival is modeled as a Laplace distribution:

$$P_a(log\,(\kappa_a) \mid A_a^q) = \frac{1}{2\sigma_a} \exp\left(-\frac{|log\,(\kappa_a) - log(A_a^q) - \mu_a|}{\sigma_a}\right).$$

***Train:** The parameters of the Laplace distribution are determined by Maximum Likelihood Estimation (MLE) using the coda training set.

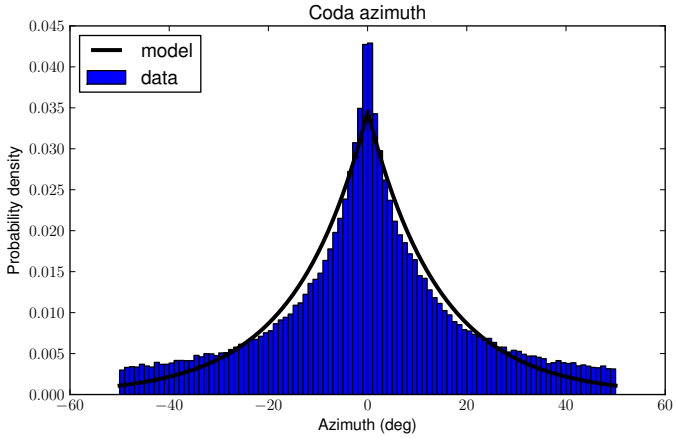


Figure 38 – Coda arrival azimuth as a function of triggering arrival is modeled as a Laplace distribution.

6.2.5.5 Measured Arrival Azimuth (coda arrival) The distribution for the difference in azimuth of the coda arrival vs. the azimuth of its triggering arrival is modeled as a Laplace distribution as shown in Figure 38.

$$P_z(\kappa_z \mid A_z^q) = \frac{1}{2\sigma_z} \exp\left(-\frac{|\kappa_z - A_z^q - \mu_z|}{\sigma_z}\right).$$

***Train:** The parameters of the Laplace distribution are determined by Maximum Likelihood Estimation (MLE) using the coda training set.

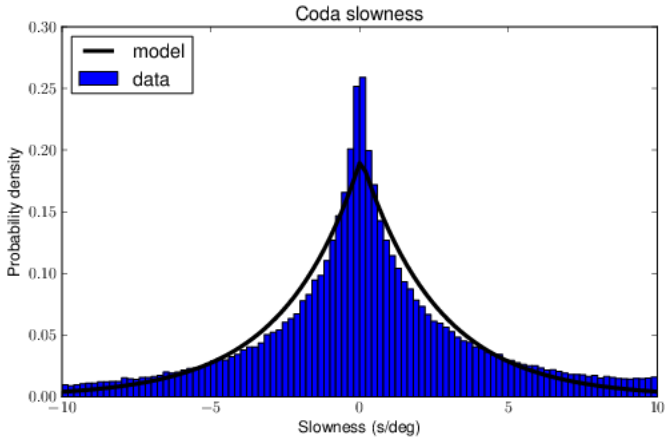


Figure 39 – Coda slowness as a function of slowness of triggering arrival is modeled as a Laplace distribution.

6.2.5.6 Measured Arrival Slowness (coda arrival) The distribution for the difference in azimuth of the coda arrival vs. the slowness of its triggering arrival is modeled as a Laplace distribution as shown in Figure 36.

$$P_s(\kappa_s \mid A_s^q) = \frac{1}{2\sigma_s} \exp\left(-\frac{|\kappa_s - A_s^q - \mu_s|}{\sigma_s}\right)$$

***Train:** The parameters of the Laplace distribution are determined by Maximum Likelihood Estimation (MLE) using the coda training set.

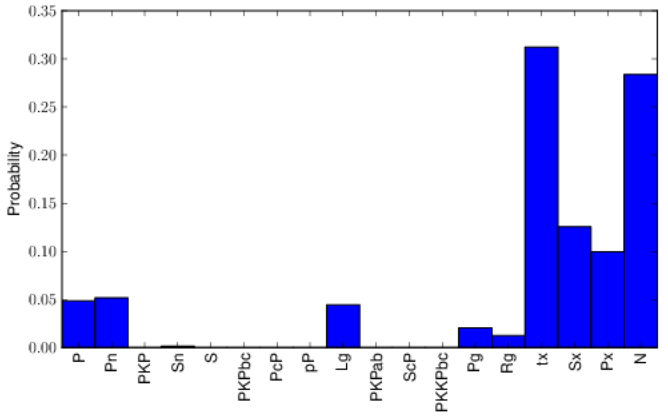


Figure 40 – Coda phase does not depend on phase of triggering arrival.

6.2.5.7 Arrival Phase Label (coda arrivals) (Figure 40)The probability of a coda phase label is modeled as a multinomial distribution, P_h which does not depend on the phase label of the triggering arrival.

$$P_h(\kappa_h) = p_h(\kappa_h).$$

***Train:** The distributions for phase labels of coda arrivals are learned from the training data using add-one smoothing exactly as they are learned for True Arrivals.

6.3 Probability of a world

Referring to Figure 1, the probability of a world is the joint probability of all of the events and arrivals in it. To compute this joint probability, the probability of all of the attributes of the events and the attributes of arrivals needs to be computed using the instructions in Section 6.2. When computing the joint probability of a world, the boldface symbols $\mathbf{e}$,$\mathbf{\Lambda}$,$\boldsymbol{\xi}$and $\boldsymbol{\kappa}$ are used to denote the complete sets of events, associated, false and coda arrivals in the world respectively.

6.3.1 Probability of an event

It is assumed that the time, location, depth and magnitude of an event are independent of each other. This means that the probability of the event itself is the product of probabilities of its attributes:

$$P\left(e\right) = P_t\left(e_t\right) P_l\left(e_l\right) P_d\left(e_d\right) P_m\left(e_m\right).$$

6.3.2 Joint probability of a set of events

6.3.2.1 Probability of the total number of events in a time period It is assumed that events occur according to time-homogeneous Poisson process with a rate parameter, λ_e. If $\mathbf{e}$ is a set of N events, and T is the length of time of the world, then

$$P_N(N) = \frac{\left(\lambda_e \cdot T\right)^N \exp\left(-\lambda_e \cdot T\right)}{N!}.$$

The estimated value of λ_e was determined from historical event frequencies. It is 0.001266 events per second or 4.6 events per hour.

6.3.2.2 Probability of a set of events It is assumed that all of the events in a set of events are independent of each other and they could have, with equal probability, occurred in any order. Therefore, the joint probability of the set is the product of the probabilities of each individual event, times the probability of the total number of events, times the probability that the events occurred in the actual order that they did. This last term is proportional to all of the possible permutations of the set. This reduces to:

$$P\left(\mathbf{e}\right) = P_N\left(N\right) N! \prod_{i=1}^{N} P_t\left(e_t\right) P_l\left(e_l\right) P_d\left(e_d^i\right) P_m\left(e_m\right)$$

6.3.3 Probability of an arrival

The attributes of an arrival are independent of each other although for slightly different reasons depending on the designation of the arrival. The attributes of an associated arrival are

conditionally independent given the event associated with it. The attributes of a false arrival are independent by assumption. And the attributes of a coda arrival are conditionally independent of each other given the preceding arrival. So, although practically the probability of any arrival is the product of the probabilities of its attributes (including the probability that it was detected), the formalism for writing it down is slightly different in the three cases. The general formula is:

$$P\left(A\right) = P_d\left(A_d\right) P_t\left(A_t\right) P_z\left(A_z\right) P_a\left(A_a\right) P_s\left(A_s\right) P_h\left(A_h\right).$$

More specifically, in the case of an associated arrival:

$$P\left(\Lambda|e\right) = P_d\left(\Lambda_d|e\right) P_t\left(\Lambda_t|e\right) P_z\left(\Lambda_z|e^i\right) P\left(\Lambda_a|e\right) P\left(\Lambda_s|e\right) P_h\left(\Lambda_h|e\right).$$

Of course, not all arrivals are detected. The probability of an arrival which was predicted by an event and not detected is the probability that it was not detected:

$$1 - P_d\left(A\right).$$

In the case of a false arrival:

$$P(\xi) = P_t(\xi_t) P_z(\xi_z) P_s(\xi_s^{kl}) P_a(\xi_a) P_h(\xi_h)$$

In the case of a coda arrival given its triggering arrival:

$$P(\kappa|A^q) = P_d(\kappa_d|A_a^q) P_t(\kappa_t|A_t^q) P_z(\kappa_z|A_z^q) P_s(\kappa_s|A_s^q) P_a(\kappa_a|A_a^q)$$

6.3.4 Joint probability of sets of arrivals

6.3.4.1 Joint probability of a set of associated arrivals For every event, there is the possibility that 14 phases were detected at every station in the IMS seismic array. $\mathbf{\Lambda}$ is the complete set of arrivals which can be associated with all of the events in $\mathbf{e}$. If there are N events, there is the possibility of J total number of phases being detected at K total number of stations. Assuming that all of the arrivals are independent of each other, regardless of their designation or association to various events, then the total probability of the list of associated arrivals is the product of all of the arrivals in it, including arrivals that were predicted but not detected. The formula for this is

$$P(\mathbf{\Lambda} \mid \boldsymbol{e}) = \prod_{i=1}^{|N} \prod_{j=1}^{J} \prod_{k=1}^{K} \left[\mathbb{1}(\Lambda^{ijk} = \zeta)(1 - P_d^{jk}(e^i)) + \mathbb{1}(\Lambda^{ijk} \neq \zeta) P_d^{jk}(e^i) P^{jk}(\Lambda^{ijk} \mid e^i) \right].$$

Recall that if an arrival of phase j caused by event i was not detected at station k, it is given a null value: $(\Lambda^{ijk} = \zeta)$. The symbol $\mathbb{1}(\cdot)$ is the *indicator function* which is similar to a Kronecker delta function. If the relationship in the parenthesis is true, then $\mathbb{1}(\cdot)$ =1 otherwise $\mathbb{1}(\cdot)$ =0. The equation is written this way so that NET-VISA takes negative evidence into account: if an arrival is not detected, the fact that it was not detected contributes to the overall probability of the particular world being examined.

6.3.4.2 Joint probability of a set of false arrivals Any station k generates a set of false detections $\boldsymbol{\xi^k}$which correspond to arrivals that cannot be associated to any event. The probability of N_ξ false arrivals being generated at the station during time period T is modeled as a time-homogeneous Poisson process with rate λ_f^k:

$$P_f^k(N_\xi) = \frac{(\lambda_f^k \cdot T)^n \exp\left(-\lambda_f^k \cdot T\right)}{n!}.$$

The rate is the average number of false arrivals during the time period.

Since false arrivals are independent of each other, the probability of the list of all false arrivals at a given station is the product of the probability of all of the false arrivals at that station.

$$P^k(\xi^{kl}) = \exp\left(-\lambda_f^k \cdot T\right) \prod_{l=1}^{l=|\xi^k|} \frac{\lambda_f^k}{M_z M_s} P_a^k(\xi_a^{kl}) P_h^k.$$

The probability of the set of false arrivals at all stations is the product of the probabilities of the sets of false arrivals at all of the stations:

$$P\left(\boldsymbol{\xi}\right) = \exp\left(-\sum_{k=1}^{K} \lambda_f^k \cdot T\right) \prod_{k=1}^{K} \prod_{l=1}^{l=|\xi^k|} p^k(\xi^{kl}).$$

where $p^k(\xi^{kl})$is defined as

$$p^k(\xi^{kl}) = \frac{\lambda_f^k}{M_Z M_s} P_a^k(\xi_a^{kl}) P_h^k(\xi_h^{kl}).$$

6.3.4.3 Joint probability of a set of coda arrivals If there are P coda arrivals in a list of coda arrivals then

$$P\left(\boldsymbol{\kappa}|\boldsymbol{A}\right) = \prod_{p=1}^{P} P_d(\kappa^p|a^q) P(\kappa^p|a^q).$$

6.3.5 Probability of a world

[/IDCNETVISA/INFER/INFER.C]

Given a list of arrivals, **A**, (which may be true, false or coda), the goal of NET-VISA is to find the following that has maximum joint probability given the arrivals:

1. a list of a events, **e**
2. a list of arrivals associated with the events, $\boldsymbol{\Lambda}$
3. a list of arrivals which are false, $\boldsymbol{\xi}$
4. and a list of coda arrivals $\boldsymbol{\kappa}$

It is stipulated that every arrival in $\mathbf{A}$ that is not associated with an event be designated either false or coda. NET-VISA's goal, therefore, is to find

$$\arg \max_{\boldsymbol{e},\boldsymbol{\Lambda},\boldsymbol{\xi},\boldsymbol{\kappa}} P(\boldsymbol{e}, \boldsymbol{\Lambda}, \boldsymbol{\xi}, \boldsymbol{\kappa}, \boldsymbol{A}).$$

where the joint probability of a list of events and lists of associated, true and coda arrival is

$$P(\boldsymbol{e}, \boldsymbol{\Lambda}, \boldsymbol{\xi}, \boldsymbol{\kappa}, \boldsymbol{A}) = P(\boldsymbol{\Lambda}|\boldsymbol{e})P(\mathbf{e})P(\boldsymbol{\xi})P(\boldsymbol{\kappa})\mathbb{1}(\boldsymbol{A} = \boldsymbol{\Lambda} \oplus \boldsymbol{\xi} \oplus \boldsymbol{\kappa}).$$

In the last term, the symbol $\oplus$ indicates exclusive or. It is required to ensure that every arrival is given exactly one classification: associated, false or coda.

7 Inference Algorithm

[IDCNETVISA_INFER runs the inference algorithm on a designated time window of data. Continuous inference (producing a continuous bulletin) is done using IDCNETVISA_INFER_BG.]

7.1 Overview

The IA uses a heuristic search algorithm designed to find the NET-VISA world of highest probability. This means that it must maximize the joint probability of four sequences: the list of events, the list of true arrivals, the list of false arrivals and the list of coda arrivals. Key to doing this is to iterate between associating arrivals to events and refining the events themselves; that is, find the best association of arrivals with events, update the time, location and magnitude of the events accordingly, re-associate the arrivals and then re-update the events. Eventually a stable solution is obtained.

The joint probability of any hypothesized sequence of events $\mathbf{e}$, true arrivals $\boldsymbol{\Lambda}$, false arrivals $\boldsymbol{\xi}$, and coda arrivals $\boldsymbol{\kappa}$ for a given set of observed arrivals $\mathbf{A}$, is

$$P(e, \Lambda, \xi, \kappa, A) = P(e)P(\Lambda \mid e)P(\xi)P(\kappa \mid A)\mathbb{1}(A = \Lambda \oplus \xi \oplus \kappa).$$

In other words, NET-VISA is searching for

$$\arg \max_{\boldsymbol{e},\boldsymbol{\Lambda},\boldsymbol{\xi},\boldsymbol{\kappa}} P(\mathbf{e}, \boldsymbol{\Lambda}, \xi, \kappa, \mathbf{A}).$$

Technical versions of algorithms described below are given in Appendix B.

7.2 Event and Arrival scores

[/IDCNETVISA/INFER/INFER.C]

To simplify computations, each event is considered independent of every other event and is given an *event score*. The event score is defined as the ratio of the probability of two hypotheses: one in which the event exists and the other in which the event does not exist and all of its associated arrivals are classified as either false or coda which is designated $\Upsilon^k(\Lambda^{jk})$. The score for an individual event is:

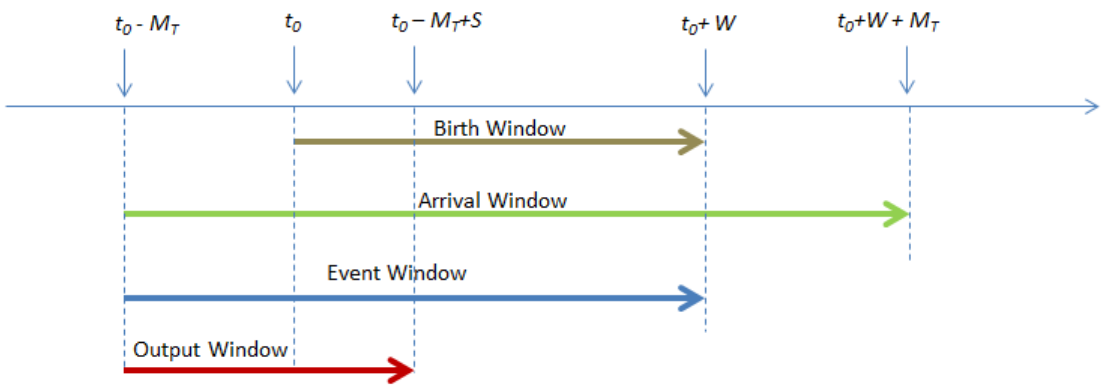


Figure 41 – NET-VISA windows. Events are proposed in the Birth Window. Arrivals in the Arrival Window are associated with events in the Event Window. After the windows are moved forward, events in the Output Window are written to the bulletin.

$$S_e(e) = P(e) \prod_{j=1}^{J} \prod_{k=1}^{K} \left[\mathbb{1}(a^{ijk} = \zeta)(1 - P_d^{jk}(e)) + \frac{\mathbb{1}(a^{jk} \neq \zeta) P_d^{jk}(e) P^{jk}(\Lambda^{jk} \mid e)}{(1 - P_d^{jk}(e)) \Upsilon^k(a^{jk})} \right].$$

This can be decomposed into scores for individual arrivals. The score for an arrival is:

$$S^{jk}(a^{ijk} \mid e) = \frac{P_d^{jk}(e)}{(1 - P_d^{jk}(e))} \frac{P^{jk}(\Lambda^{ijk} \mid e)}{\Upsilon^k(a^{ijk})} .$$

An arrival is classified coda if its probability of being a coda arrival is higher than its probability of being false.

$$\Upsilon(a) = \max(P(\xi), P(\kappa)).$$

7.3 Algorithm

NET-VISA divides the continuous stream of arrival data into windows as illustrated in Figure 41. The Birth Window is of size W , starting from $t_0 = 0$ to $t_1 = t_0 + W$ (in practice, W is 30 minutes long). Events will be proposed in this window. The Arrival window goes from $t_0 = 0$ to $t_2 = t_0 + W + M_T$ where M_T is the maximum travel time for any phase (33.3 minutes, or 2000 seconds). Arrivals in this window are associated with events in the Event Window which goes from $t_{-1} = -M_T$ to t_0.

The NET-VISA process is as follows:

1. Arrivals in the Arrival Window which do not have previous classifications are assigned a label of either false or coda.

2. New events are proposed in the Birth Window (Birth Move).

3. The algorithm iterates between finding the best set of associations and optimizing events using algorithms IMPROVE-ARRIVAL and IMPROVE-EVENT.

4. Events which have an event score less than one are discarded (Death Move).

5. A final association of arrivals with the remaining events is made.

6. The events are optimized one last time.

7. The algorithm moves ahead: $t_0 += S$, $t_1 += S$, $t_2 += S$, ($S = 30$ minutes).

8. Events in the Output Window (which are now older than $t_0 - M_T$) can no longer be modified and are returned by the algorithm.

9. The arrivals associated with these events are returned. False and coda arrivals in the Output Window are returned.

An arrival retains its original classification unless it becomes associated with an event.

7.3.1 Birth Move

[/IDCNETVISA/INFER/PROPOSE.C]

The Birth Move requires the most computation of any algorithm in NET-VISA. Strategies for speeding up NET-VISA by parallelization of the Birth Move are given in [Arora, 2012a]. The algorithm is as follows:

1. For every arrival in the New Arrival window create a seed event by assuming the arrival is a P and inverting the time, azimuth, slowness, and amplitude of the arrival to find the time, location, and magnitude of the event.

2. For every seed event, perform the following:

   (a) Construct sets of events with varying depths and locations around the seed event. The events are constructed in two different ways:

      i. If the inverted distance is more than 20 degrees then the azimuth and slowness of the arrival are perturbed before being inverted.

      ii. If the inverted distance is less than 20 degrees then create events in a 5-degree by 50 second space-time ball around the inverted location. This latter step is designed specifically for finding regional events.

   (b) For each constructed event:

      i. find the best set of unassociated arrivals from the Arrival window which can be associated with that event.

      ii. assign an Event score to the constructed event.

   (c) Select the event with the highest score.

(d) Optimize the best event by repeating the following 100 times:

i. invoke improve-event algorithm

ii. invoke improve-arrival algorithm for all of its associated arrivals with this event as the only potential event

(e) If the score of the event is greater than one, set it and its associated arrivals aside.

3. Return the events that have been set aside

7.3.2 Improve-arrival

The improve-arrival algorithm associates or re-associates arrivals to events. The algorithm is as follows:

1. For each arrival at each station

   (a) For each event

      i. Determine the phase of the arrival as if it were associated with this event by choosing the phase that has the highest score

   (b) Find the event for which the arrival has the highest score

   (c) If the arrival had previously been associated with another event, associate it to the event that gives it the highest score.

   (d) If the arrival was previously unassociated, but achieves a score higher than one, associate it to the event that gives it the highest score. Otherwise, keep its original label of false or coda.

   (e) If an arrival could be assigned as the coda of an event with score more than one then do not assign this arrival to a different event with a lower event score.

7.3.3 Improve Event

The Improve-Events algorithm is used to find a better location, depth and time for events for which associations have already been made. For every event, 100 new events are chosen uniformly at random from a small ball around the event (2 degrees in latitude and longitude, 100 km in depth, 5 seconds in time and 2 units of magnitude). The score of all of these events is computed. The event with the highest score is chosen.

7.3.4 Death-Move

Any event with a score less than one is deleted and all of its currently associated arrivals are marked as false.

7.3.5 Finish

After the Death-Move, the algorithm moves ahead and stable events in the Output Window are written out. The standard IDC libraries LIBLOC and LIBMAG are used to make a final location and magnitude estimate of the events using the NET-VISA associations with NET-VISA locations as a starting point. LIBLOC is run in order to compute error ellipses. Location solutions are damped so that event locations do not change significantly from VISA locations. LIBMAG is run to compute NETMAG and STAMAG parameters as expected by analysts. Finally, the events are passed into the VISA schema where they can be presented to analysts through the standard Analyst Review Station (ARS) software to make the LEB-VISA bulletin.

Part III
Practical Guide for IDCNETVISA

IDCNETVISA is an implementation of the NET-VISA software package which has been developed and optimized for installation at the IDC. It is a package of PYTHON programs which call modules written in PYTHON and c which implement the Generative Model and Inference Algorithm as described in previous sections of this report. Although it is a stand alone package, it requires an appropriate version of PYTHON be installed and integration with the IDC database. Two basic scripts compose the working parts of IDCNETVISA. They are

- IDCNETVISA_LEARN which is used to learn the parameters of the PDFs in the GM using historic data. Once a set of parameters has been learned, it is referred to as a *model* and stored in a model file.
- IDCNETVISA_INFER which runs the IA and produces the VISA seismic bulletin.

Associated with them are LEARN.PAR and INFER.PAR, the parameter files which contain options and instructions necessary to run the scripts. Two additional helper scripts are provided

- IDCNETVISA_INFER_BG is used if inference is to be run on continuously streaming data instead of a single pre-designated time block.
- IDCNETVISA_DEBUG is used to get more details about an event in a NET-VISA, or other, bulletin.

The rest of the files in the top directory are used to install the package. Up-to-date installation instructions are released with each distribution and are contained in the README file. The following descriptions of the scripts and utilities is meant for reference and to be a guide to users who are unfamiliar with IDCNETVISA.

8 Scripts and Utilities

The top level directory in an IDCNETVISA distribution contains the scripts and files necessary to run IDCNETVISA at the command line (or in an automatic queue) and files necessary for installation.

8.1 README

This file contains up to date explicit instructions for installing and running every new distribution of IDCNETVISA. It should be consulted before any attempt at installation is made.

8.2 MANIFEST.IN

`MANIFEST.IN` contains the list of files in the IDCNETVISA distribution.

8.3 PYPROJECT.TOML

`PYPROJECT.TOML` describes build system for the IDCNETVISA distribution.

8.4 REQUIREMENTS.TXT

`REQUIREMENTS.TXT` lists all the Python packages that are needed for the IDCNETVISA distribution.

8.5 SETUP.PY

This script together with `PYPROJECT.TOML` installs the IDCNETVISA package including all of the files necessary to run IDCNETVISA_LEARN and IDCNETVISA_INFER. Before installing IDCNETVISA, the user must be a Python virtual environment with all of the packages mentioned in REQUIREMENTS.TXT already installed.

Another prerequisite is the IDC libraries named CBASE, SBASE, and IDCMODEL. These libraries are expected to be installed in a location pointed to by IDCLIBINSTALLDIR.

Once all the prerequisites are addressed the IDCNETVISA distribution can be created by simply running:

```
pip install .
```

The above command will install IDCNETVISA python package and four PYTHON scripts: IDCNETVISA_INFER, ICDNETVISA_INFER_BG, IDCNETVISA_LEARN, and IDCNETVISA_DEBUG.

8.6 IDCSHEMA.SQL

IDCSCHEMA.SQL creates all of the database tables and synonyms to existing IDC static and idcx tables necessary to read from and write to the idc database. The database schema is described in Section 12.

8.7 MAKEINFRAPRIOR.PY

MAKEINFRAPRIOR.PY creates a static prior for infrasound stations. This command is only required when a new infrasound static prior is desired. The source code comes with a static prior already installed under the IDCNETVISA/STATICPRIOR directory. The usage of this command is as follows:

```
$ python scripts/makeinfraprior.py <conn_str> <network list> <start year> <end year
```

The description of the parameters

- conn_str - a database connect string to query
- network list – a comma separated list of infrasound networks, e.g. CUR_INF,INFRA
- start year – the starting year (inclusive) for learning the models
- end year – the ending year (inclusive) to be used for learning the model
- out dir – a directory where the static priors are generated
- fig dir – a directory where generated model figures are stored

8.8 POPULATE_ISC.PY

POPULATE_ISC.PY populates the table ISC.EVENT with the IASPEI ground truth bulletin downloaded from http://www.isc.ac.uk/gtevents/. The usage of this command is as follows:

```
$ python scripts/populate_isc.py <GT file name>
```

8.9 Installed Scripts

In order to run the python scripts, the following environment must be set up:

- The user must be in a PYTHON virtual environment with NUMPY, SCIPY, MATPLOTLIB CX_ORACLE and BASEMAP installed. Also, the IDCNETVISA package must be installed in this environment. For example:

  ```
  $ source /dvl/shi/net_visa/virtualpy/bin/activate.csh
  ```

(NOTE: it is not necessary to be in a clearcase view)

- LD_LIBRARY_PATH must point to the Oracle dynamic libraries as well as the PYTHON dynamic libs, if necessary. For example:

  ```
  $ setenv LD_LIBRARY_PATH /dvl/software/site/usr/lib64:/cots/oracle/oracle-11.2/lib
  ```

- ORACLE_HOME must point to the correct Oracle installation. For example:

  ```
  $ setenv ORACLE_HOME /cots/oracle/oracle-11.2
  ```

- TNS_ADMIN must point to the location of tnsnames.ora. For example:

  ```
  $ setenv TNS_ADMIN /cots/oracle
  ```

8.9.1 IDCNETVISA_LEARN

IDCNETVISA_LEARN is the basic script for learning the parameters of the probability distributions in the Generative Model. Its output is a set of model files which will be used for inference. Parameters can be included in a file or as arguments at the command line. Parameters are input in standard IDC format. A sample LEARN.PAR file is included in this distribution. To run IDCNETVISA_LEARN for example:

```
$ idcnetvisa_learn par=learn.par start_time=1325376000 end_time=1333238400
```

The required parameters are listed below.

- start_time – epoch time
- end_time – epoch time
- netvisa_model_dir – directory where the model-<start-time>-<end-time> sub-directory will be created, as well as the symbolic link "curr"
- netvisa_sasc_dir – Directory containing SASC corrections
- netvisa_ttime_prefix – prefix of travel time tables
- netvisa_ddrange_file – File containing phase distance-depth ranges
- netvisa_qfvc_file – File containing Q-Factors (Veith Clawson)
- netvisa_radial2d_hydro_dir – Radial2D hydro travel time directory
- netvisa_radial2d_infra_dir – Radial2D infrasound travel time directory
- vmodel_spec_file – Velocity model specification file
- netvisa_use_libloc - defaults to 0. This controls whether the libloc library should be used for training the arrival time residuals. Requires NET-VISA to be linked with the multi-threaded libloc library.
- netvisa_sssc_level - defaults to 0. This controls the SSSC corrections level if the netvisa_use_libloc has been set to 1.
- in_db_account – connect string to the database

- in_db_type – defaults to oracle. This specifies the database backend. The only other supported value (apart from oracle) is postgres.
- At least one of the following network parameters is required:
  - net – a list of networks, example: CUR_PRI,CUR_AUX,CUR_HYD,CUR_INF
  - net_aux – a list of auxiliary networks, example: CUR_AUX
  - hydro_net – a list of hydroacoustic networks, example: CUR_HYD
  - infra_net – a list of infrasound networks, example: CUR_INF

The following are optional parameters:

- in_site_table – defaults to static.site
- in_affiliation_table – defaults to static.affiliation
- in_arrival_table – defaults to idcx.arrival
- in_assoc_table – defaults to leb.assoc
- in_origin_table – defaults to leb.origin
- netvisa_model – list of model nodes to train, default: all model nodes (see section ...)
- netvisa_pdf – outputs figures to a file
- netvisa_gui
- netvisa_store_data - defaults to 1. This parameter controls whether or not a Data.pic file is generated in the model directory. The Data.pic file can be used by the IDCNETVISA_VISUALIZE script to display the model parameters with reference to the data that they were trained on.
- netvisa_store_parquet - defaults to 0. This parameter controls whether or not the data is written out in parquet and csv files in the model directory. This data will be used in a future version of NET-VISA for training and visualization.
- verbose

NOTES:

1. Do not the run the above command from the IDCNETVISA source directory! That would invoke the incorrect version of the script.
2. Change netvisa_model_dir in learn.par to the location where model files are to be stored.

3. Change in_db_account to a connect string on any database of your choice. However, this database must have all the arrivals between start_time and end_time. The archive database is an appropriate choice for this.

4. Change EARTHSPECS to the appropriate directory on your LAN. The provided value in learn.par is fine for the devlan.

5. It takes about 3.5 hours to learn a model using 3 months of data.

The new model will be learned and stored it in the sub-directory model-<start_time>-<end_time>. For Example:

```
$(netvisa_model_dir)/model-1325376000-1333238400
```

The symbolic link $(netvisa_model_dir)/curr will point to the above directory. Keep a note of the directory "netvisa_model_dir". This will be need to be set to run IDCNETVISA_INFER. The progress of IDCNETVISA_LEARN is logged using the standard IDC syslog framework under the name ctbto.netvisa.learn

8.9.2 IDCNETVISA_INFER

IDCNETVISA_INFER is the basic script for inferring seismic events using the IA and generating a seismic bulletin. Parameters can be included in a file or as arguments at the command line. Parameters are input in standard IDC format.

IDCNETVISA_INFER requires the input parameters listed below.

- start_time – start time for inference, epoch time
- end_time – end time for inference, epoch time
- netvisa_curr_model – NET-VISA model parameter are read from here
- netvisa_ttime_prefix – prefix of travel time tables, e.g. prefix+".P" is the P phase travel time table
- netvisa_ddrange_file – distance depth ranges for each phase
- netvisa_qfvc_file – QFVC file
- netvisa_radial2d_hydro_dir – Radial2D hydro travel time directory
- netvisa_radial2d_infra_dir – Radial2D infrasound travel time directory
- netvisa_sasc_dir - SASC corrections directory
- in_db_account - connect string to read the arrivals
- in_db_type – defaults to oracle. This specifies the database backend. The only other supported value (apart from oracle) is postgres.

- out_db_account - connect string to write the events
- out_db_type – defaults to oracle. This specifies the database backend. The only other supported value (apart from oracle) is postgres.

The following parameters are needed for initialization of libloc:

- vmodel_spec_file
- sasc_dir_prefix

The following parameters are needed for initialization of libmagnitude:

- mag_descrip_file
- tl_spec_file
- list_of_magtypes

Optional parameters:

- netvisa_threads - number of threads for birth moves; defaults to 1 This should be set to the number of cpus or cores on the machine for maximum performance. If the number of threads is set higher than the number of cpus, the default will be to use all available cores.
- netvisa_numsamples - defaults to 20
- netvisa_window - defaults to 1800
- netvisa_window_step - defaults to 900
- netvisa_seed - defaults to 123456789
- look_ahead - defaults to 10800 (or 3 hours).
- origin_time_limit - defaults to 1200 (or 20 minutes). Origins that were created these many seconds before the start of the current window will not be modified, and their associations will not change.
- clutter_lookback - defaults to 10800 (or 3 hours). This is the time interval for which we look for infrasound clutter events to update the static clutter model. This parameter is also used to look for recent hydroacoustic arrivals to train the hydroacoustic clutter model as well.
- netvisa_uniform_loc - defaults to 0. Making this 1 gives NET-VISA a uniform location prior while making it a fractional value between 1 and 0 would interpolate between a uniform prior and the learned location prior. Note that the learned location prior already includes a 0.001 uniform location density built into it.

- netvisa_extra_seismic_prior - defaults to 0. Giving a positive number here increases the log prior rate of seismic events per hour by this amount.
- netvisa_initial_temperature - defaults to 0.1. A temperature of zero forces all constraints to be respected while a temperature much higher than 1 ignores constraints.

NOTE: The next five parameters control the birth proposals, changing them can lead to a severe degradation in runtime

- netvisa_prop_delaz_max - defaults to 10.0
- netvisa_prop_delaz_step - defaults to 2.0
- netvisa_prop_delslo_max - defaults to 1
- netvisa_prop_delslo_step - defaults to .2
- netvisa_prop_dist_step - defaults to 1. This controls the step size at which candidate events are proposed in the birth move for hydro and infra arrivals, which don't have a slowness value to estimate distance.
- netvisa_use_libloc - defaults to 0. This controls whether the libloc library should be used throughout the inference. Requires NET-VISA to be linked with the multi-threaded libloc library.
- netvisa_sssc_level - defaults to 0. This controls the SSSC corrections level if the netvisa_use_libloc has been set to 1.
- netvisa_compute_origerr - defaults to 1. This controls whether NET-VISA should compute the values for the origerr table, instead of offloading to libloc.
- verbose – verbose output
- in_site_table – defaults to static.site
- in_arrival_table – defaults to idcx.arrival
- in_amplitude_table – defaults to idcx.amplitude
- out_event_table – defaults to visa.event
- out_origin_table – defaults to visa.origin
- out_origerr_table – defaults to visa.origerr
- out_assoc_table – defaults to visa.assoc
- out_netmag_table – defaults to visa.netmag
- out_stamag_table – defaults to visa.stamag
- out_lastid_table – defaults to visa.lastid

- visaorigin_table – defaults to visaorigin
- visabirth_table – defaults to visabirth
- visaassoc_table – defaults to visaassoc
- pre_origin_table – The origin table for a prior bulletin which is to be integrated into the current bulletin. NOTE: Event identifiers from the prior bulletin will be preserved if those prior events are retained.
- pre_assoc_table – The assoc table for a prior bulletin which is to be integrated into the current bulletin.
- seismic_defining_phases – This is the list of phases that are given to libloc to locate a seismic event and to compute the residuals for the arrivals. This list defaults to – Pg, Pn, Sn, Lg, P, S, PcP, ScP, PKP, PKiKP, PKPbc, PKPab, SKP, SKPbc.
- hydro_defining_phases – This is the list of phases that are given to libloc to locate a hydro (in-water) event, and to compute the residuals for the arrivals. This list defaults to – H.
- infra_defining_phases – This is the list of phases that are given to libloc to locate an infra (in-air) event, and to compute the residuals for the arrivals. This list defaults to - I.

Optional parameters for Event Definition Criteria:

- netvisa_use_event_def - defaults to 0. This controls whether NET-VISA should use event definition criteria to filter out origins. The remaining parameters below are used only if netvisa_use_event_def is set to 1.
- weight_threshold - defaults to 3.55. Events with weight below this threshold will be dropped if netvisa_use_event_def is set to 1.
- Seismic weights:
  - primary_time_weight - defaults to 1.0. This is the weight of primary time defining phase (P,PKP,Pn,Pg) arrivals.
  - secondary_time_weight - defaults to 0.7. This is the weight of all other (i.e., non-primary) time defining phase arrivals.
  - array_azimuth_weight - defaults to 0.4. This is the weight of azimuth defining arrivals at array stations.
  - 3comp_azimuth_weight - defaults to 0.2. This is the weight of azimuth defining arrivals at 3-component stations.
  - array_slow_weight - defaults to 0.4. This is the weight of slowness defining arrivals at array stations.

  - 3comp_slow_weight - defaults to 0.2. This is the weight of slowness defining arrivals at 3-component stations.

- Hydro and Infra weights:
  - hydro_time_weight - defaults to 1.2. This is the weight of time defining arrivals at hydro stations.
  - hydro_azimuth_weight - defaults to 0.6. This is the weight of azimuth defining arrivals at hydro stations.
  - hydro_slow_weight - defaults to 0. This is the weight of slowness defining arrivals at hydro stations.
  - infra_time_weight - defaults to 0.8. This is the weight of time defining arrivals at infra stations.
  - infra_azimuth_weight - defaults to 1.0. This is the weight of azimuth defining arrivals at infra stations.
  - infra_slow_weight - defaults to 0. This is the weight of slowness defining arrivals at infra stations.

NOTES

- Arrivals at auxiliary stations are not included when calculating the weight for an event.

Optional libloc parameters:

- loc_verbose
- sssc_level
- ellip_cor_level
- ignore_big_res
- big_res_mult
- dist_var_wgt
- only_sta_w_corr
- use_tscor ts_region
- num_dof
- conf_level

NOTES

- libloc parameters max_iterations and damp are fixed to 10 and 20000 respectively. They can't be changed.

- To compute coverage ellipse (and not confidence ellipses) at the 90% confidence level, the following two parameters must be set accordingly:
  - dist_var_wgt=1
  - num_dof=99999
- sssc_level and netvisa_sssc_level are different parameters. The former determines whether SSSCs are used in the final location output by NET-VISA. The latter, on the other hand, is only relevant when netvisa_use_libloc=1, and it controls whether SSSCs are used by the NET-VISA location used during the entirety of the inference process and not just at the end.

Optional libmagnitude parameters:

- list_of_mb_magtypes
- mag_verbose
- mag_num_boots
- mag_sub_sta_list
- magtype_to_origin_mb – defaults to mb
- magtype_to_origin_ms – defaults to ms
- magtype_to_origin_ml – defaults to ml
- mag_compute_upper_bounds
- mag_ignore_big_res
- mag_large_res_mult
- mag_only_sta_w_corr
- mag_sub_sta_list_only
- mag_use_tscor
- mag_ts_region

Miscellaneous parameters:

- max_arrival_lag – defaults to 0, or infinite. If specified, it limits the arrivals to those whose lddate - time is within these many seconds. For example, for running with 6 hour lag specify 21600.

- restrict_network – defaults to "". If specified it limits the arrivals to those from this specified list of comma-separated networks, otherwise arrivals from all networks in the learned model are used. Note: INFER can't use arrivals from a network that were not specified during LEARN.
- in_affiliation_table – required if restrict_network is specified.
- skip_empty_interval – skips an interval with no arrivals, default False.

A sample infer.par file is included in this distribution. To run IDCNETVISA_INFER :

```
$ idcnetvisa_infer par=infer.par start_time=1336262400 end_time=1336264200
```

The above command will write events to the event, origin, origerr, assoc, netmag, and stamag tables. The progress of IDCNETVISA_INFER is logged using the standard IDC syslog framework under the name ctbto.netvisa.infer

IDCNETVISA_INFER deletes any events from start_time-2000 to end_time in the database and recomputes them. It infers new events from start_time to end_time (using a birth move) and merges them with the old events from start_time-2000 to start_time. Any existing events from start_time to end_time in the database are deleted. The merged events are written out in a single transaction which combines the deletion of any old or existing events. Thus if infer is to crash, no previous results are lost. It is always safe to re-run infer on a time interval assuming that the needed arrivals are available. infer reads arrivals from start_time-2000 to end_time+2000.

The magic number of 2000 seconds is the maximum travel time of any seismic phase that is modeled by IDCNETVISA. One can set the parameter pre_start_time, which defaults to start_time-2000, to some other value to alter the behavior of infer. For example, setting pre_start_time to start_time would prevent infer from merging with any past events. This is inadvisable unless there is a specific reason.

NOTES:

1. Change netvisa_curr_model in infer.par to the location where model files were stored by the learn command.
2. Change in_db_account to a connect string on any database of your choice. It is not necessary to pick the same database as used by IDCNETVISA_LEARN. However, subsequent calls to IDCNETVISA_INFER should use the same database.
3. Change out_db_account to a connect string for a user who has read and write privileges on all the "out_*" tables.
4. Change EARTHSPECS to the appropriate directory on your LAN. The provided value in infer.par is fine for the devlan.
5. It might be a good idea to query max(time) from visa.origin and pass that as the start_time. (Start where the previous time left off)

8.9.3 IDCNETVISA_INFER_BG

IDCNETVISA_INFER_BG is the wrapper script for IDCNETVISA_INFER. It runs infer in a continuous loop incrementally modifying the start_time. The list of parameters are as follows:

- start_time – initial start_time passed to IDCNETVISA_INFER
- infer_par_file – parameter file passed as-is to IDCNETVISA_INFER
- window – defaults to 1800 seconds (30 minutes)
- delta – defaults to 1500 seconds (25 minutes)
- realtime_delay – defaults to 21600 seconds (6 hours)

By default infer_bg continuously runs infer with a 30 minute window that slides forward by 25 minutes, and it goes to sleep if it is within 6 hours of realtime.

8.9.4 IDCNETVISA_DEBUG

The debug script prints out an event from the specified bulletin, and also the event score. It also prints out all the detections, their residuals, the detection score, and the nearest distance from the best inverted location to the event.

Required parameters

- netvisa_curr_model – The NET-VISA model which will be used to compute the event scores.
- One of
  - orid

  OR
  - start_time
  - end_time
- in_origin_db_account – database from where the origins will be read
- in_origin_table
- in_assoc_table
- in_arrival_db_account – database from where the arrivals will be read
- netvisa_ttime_prefix – prefix of travel time tables, e.g. prefix+".P" is the P phase travel time table

- netvisa_ddrange_file – distance depth ranges for each phase
- netvisa_qfvc_file – QFVC file
- netvisa_radial2d_hydro_dir – Radial2D hydro travel time directory
- netvisa_radial2d_infra_dir – Radial2D infrasound travel time directory
- netvisa_sasc_dir - SASC corrections directory

Optional parameters.

- netvisa_prop_delaz_max - defaults to 10.0
- netvisa_prop_delaz_step - defaults to 2.0
- netvisa_prop_delslo_max - defaults to 1
- netvisa_prop_delslo_step - defaults to .2
- in_site_table – defaults to static.site
- in_arrival_table - defaults to idcx.arrival

8.9.5 IDCNETVISA_MATCH

The match script can compare one or more event bulletins against a reference bulletin. It performs a one-to-one matching between compatible events in the two bulletins in such a way so as to maximize the total number of matched events. Normally two events are considered compatible based on their location and time. However, events can also be considered compatible based on number of identical associated arrivals.

The script can also generate graphs showing the distribution of events and the daily overlap and inconsistency.

Required Parameters

- start_time
- end_time
- bull1_db_account
- bull1_origin_table
- bull1_assoc_table - required if min_assoc > 0
- arrival1_table - required if min_assoc > 0
- bull2_db_account
- bull2_origin_table

- bull2_assoc_table - required if min_assoc > 0
- arrival2_table - required if min_assoc > 0

Optional Parameters

- include_seismic - default True - if set to false then seismic arrivals will be ignored
- include_infra - default True - if set to false then infrasound arrivals will be ignored
- include_hydro - default True - if set to false the hydroacoustic arrivals will be ignored
- max_arrival_lag - default 21600 - the maximum time difference between an arrival and its associated origin
- bull<n>_origin_clause - default "" - a SQL condition added to the where clause in the query to select all the origins
- bull<n>_name - defaults to "" - the name of the nth bulletin. If not specified this defaults to the name bull<n>_origin_table and bull<n>_origin_clause concatenated together
- max_distance_err - defaults to 20 degrees - Maximum distance between compatible events
- max_time_err - defaults to 2000 seconds - Maximum time difference between compatible events
- min_assoc - defaults to 0 - Minimum number of associated arrivals that should match for two events to be considered compatible. If this value is non-zero then the assoc table and the arrival table must be supplied.
- rename_phases - defaults to 1 - Some of the phases need to be renamed before they can be matched. The following table gives the renamed phase list:

Original Phase	Renamed Phase
Pn	P
Pg	P
Sn	S
PKPab	PKP
PKPbc	PKP
PKKPbc	PKP
PKiKP	PKP
PKhKP	PKP

- seismic_arrival_time_match - defaults to 5.0 - the difference in arrival time of seismic arrivals at a station that may be considered identical.

- seismic_arrival_azimuth_match - defaults to 2.0 - the difference in arrival azimuth of seismic arrivals at a station that may be considered identical
- infra_arrival_time_match - defaults to 500.0 - the difference in arrival time of infrasound arrivals at a station that may be considered identical
- infra_arrival_azimuth_match - defaults to 10.0 - the difference in arrival time of infrasound arrivals at a station that may be considered identical
- display_phases - default False - if set to true then matching statistics on phases will be printed
- figure_save_prefix - default "" - the directory and file prefix for the charts. If this is a non-empty value then charts and figures displaying the matched bulletins is written out
- figure_display - default False - whether a figure is to be displayed on the screen
- figure_suffix - default "png" - the suffix of the image file
- print_origerr - default False - if set to true then statistics related to origin error will be printed. If used in conjunction with figure_save_prefix or figure_display, then additional figures related to origin error will be generated as well
- display_events - default False - if set to true then the location of events in each bulletin will also be displayed
- collect_days - default 1 - the number of days of data that are to be collected in daily overlap, inconsistency charts
- compare_runtime - default 1 - if set to true then a chart displaying the running time will also be displayed
- verbose - default False - if set to true then a lot of detailed information about the matching will be printed

Compatibility The parameters min_assoc, max_distance_error, and max_time_err determine the criteria for compatibility between events. For example, if we want to allow two events to be matched based solely on the criteria that they share at least two arrivals, we would set min_assoc to 2 and set the other two to large enough values that they don't play a role in event compatibility.

- min_assoc=2
- max_distance_err=180
- max_time_err=2000

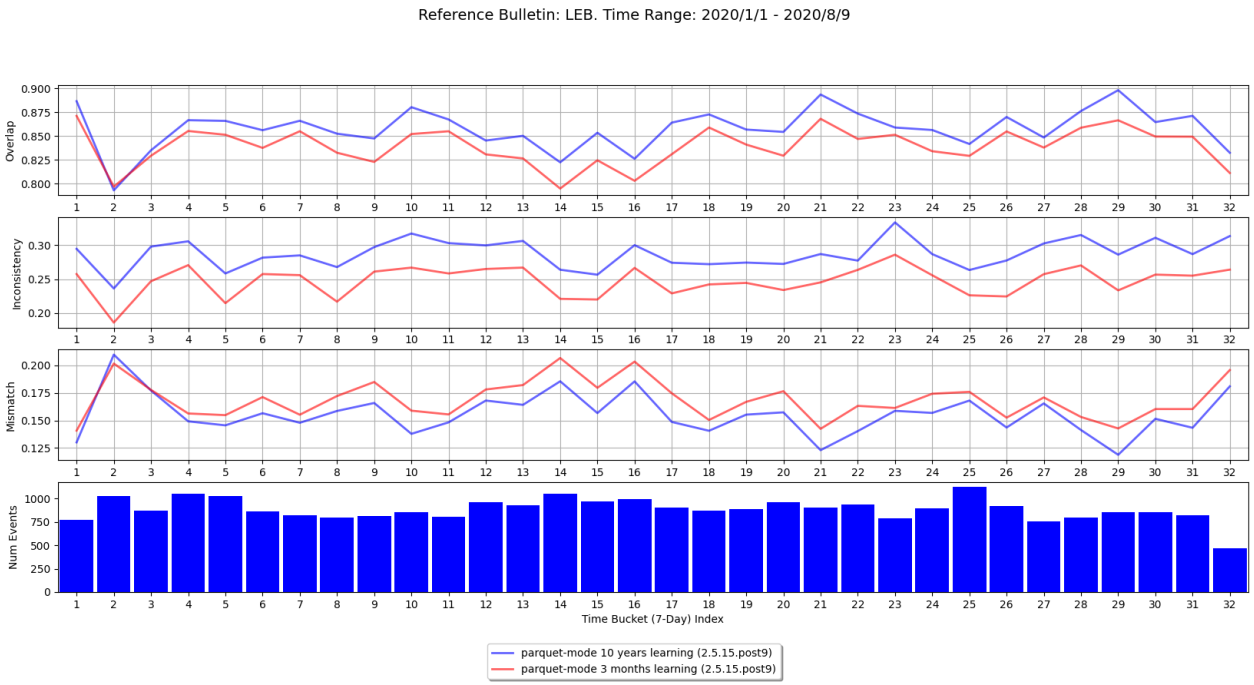
Comparing more than two bulletins The parameters bull<n>_db_account, bull<n>_origin_table, bull<n>_origin_clause, bull<n>_assoc_table, bull<n>_name, and arrival<n>_table may be repeated for n = 3, 4, 5, and 6 if additional bulletins need to be compared.

Key Metrics Reported The script reports the following key metrics to help compare the performance of multiple bulletins. Note that the first bulletin (with bull1_ prefix) is considered the reference bulletin and all subsequent bulletins are considered test bulletins.

- We define **Overlap** as the fraction of reference events that match something in the test bulletin.
- We define **Inconsistency** as the fraction of test events that do not match something in the reference bulletin.
- We define **Split** events as the number of test events that do not match something in the reference bulletin but are compatible to at least one event in the reference bulletin. This number provides a rough indicator on the number of coda events.
- We define **Mismatch** as a weighted average of 1-overlap and inconsistency which gives ten times more weight to the former. More formally, it is defined as

$$\frac{10 \times (1 - overlap) + inconsistency}{11}$$

The following chart shows an example of overlap, inconsistency and mismatch measured over weekly buckets.



8.9.6 IDCNETVISA_ADD_SITE

The add site command allows a user to add a new site to an existing trained model by using the parameters of an existing station. The resulting model can be written out to a new model file.

This command requires the following parameters:

- in_model_file: The existing model.txt file.
- out_model_file: The path to a new model.txt file to be written out. This could be the same as the in_model_file in which case the in_model_file will be overwritten.
- site_map_file: A file that provides the mapping of the new stations to existing stations as well as the location of the new station. This file is a space-separated tabular file with the following columns:

COLUMN	DATA TYPE	DESCRIPTION
existing site	STRING	The name of the existing station in the trained model
new site	STRING	The name of the new station to be added to the trained model.
longitude	REAL	The longitude of the new station.
latitude	REAL	The latitude of the new station.
elevation	REAL	The elevation of the new station in km.
isarray	0 or 1	Whether or not the new station is an array station.
isprimary	0 or 1	Whether or not the new station is a primary station.
isseis	0 or 1	Whether or not the new station is a seismic station.
ishydro	0 or 1	Whether or not the new station is a hydro-acoustic station.
isinfra	0 or 1	Whether or not the new station is an infrasound station.

Notes:

1. Any lines starting with a # character are ignored.
2. A station can only have exactly one of the columns isseis, ishydro, or isinfra with a value of 1. The other two must be zero.
3. For NET-VISA to work correctly with this new station the station must also have an entry in the static.site table and the value of longitude, latitude, elevation should match those provided here.

For example the following file adds three new stations – FOO, BAR, and BAZ. Stations FOO and BAR have the same model as existing station ASAR and BAZ has the same model as station WRA.

```
ASAR FOO 108.2 33.1 0.13 1 0 1 0 0
ASAR BAR 12 -14.4 1.4 1 1 1 0 0
WRA BAZ -120 25.3 0.0 0 0 1 0 0
```

8.9.7 IDCNETVISA_MERGE_MODEL

The merge_model script allows two models to be merged into a new model. The typical usage of this is to merge a local model into a global model. The new model will have all of the parameters from the global model unchanged with the addition of the parameters for any new station in the local model. Note that if a station occurs in both the global and the local model then we will use the parameters from the global model.

This command requires the following parameters:

- in_global_model_file: The location of an existing global model.
- in_local_model_file: The location of an existing local model.
- out_model_file: The location of a new model file that this script creates

8.9.8 IDCNETVISA_VISUALIZE

The visualize command may be used to inspect a previously learned model with graphs showing the data and the learned model. It picks up the model and training data from NETVISA_CURR_MODEL and the rest of the required parameters are a subset of the parameters that are used for learning. Note that learning must be run with the parameter NETVISA_STORE_DATA=1 so that a DATA.PIC file is generated in the model directory. This data file is needed for visualize to work correctly.

Parameters

- netvisa_curr_model - This is the location of a previously trained model and data.
- netvisa_type1 - Set to 1 if type 1 fonts are required on the figures. Defaults to 0.

Standard Parameters

- netvisa_ttime_prefix
- netvisa_ddrange_file
- netvisa_qfvc_file
- netvisa_radial2d_hydro_dir
- netvisa_radial2d_infra_dir
- vmodel_spec_file

8.9.9 IDCNETVISA_VISUALIZE_PARQUET

The parquet-based visualization script can be used to visualize the training data and the learned model if the parquet flag 8.9.1 has been set to 1 during learning. This script starts a web application on a configured port (see server_port below). The user is expected to open a browser on the local host to the specified port to access the application. The script meanwhile will keep running till the user stops it by typing CTRL-C.

Parameters

- server_port - This is the port where the web application will be available. Defaults to 8501.
- netvisa_curr_model - This is the location of a previously trained model with parquet training data stored along with the model. The following files are expected in this directory:
  - model.txt
  - data/arrival.parquet
  - data/assoc.parquet
  - data/phase.csv
  - data/site.csv

Standard Parameters

- netvisa_ttime_prefix
- netvisa_ddrange_file
- netvisa_qfvc_file
- netvisa_radial2d_hydro_dir
- netvisa_radial2d_infra_dir
- vmodel_spec_file
- netvisa_sssc_level
- netvisa_use_libloc

8.9.10 IDCNETVISA_RUNNER

The RUNNER command is automatically executed in the standard NET-VISA Docker images. The purpose of this command is to run learning and inference continuously. By default a new model is trained every week on the preceding 3 months of data. This command requires a root directory which contains:

1. A file called *next_infer_time.out* which only contains a timestamp. This is the timestamp for starting inference and this file is updated by the runner to the next timestamp after each inference interval is completed.
2. A subdirectory with the NET-VISA major and minor version, for example *2.5* which contains:
   (a) a *learn.par* file that will be used to learn a model. The runner will automatically override the following parameters in the *learn.par* file, so they may be omitted:

i. start_time
ii. end_time
iii. netvisa_model_dir - This will be set to the the subdirectory with NET-VISA major and minor version.

(b) an *infer.par* file that will be used to run inference. The runner will automatically override the following parameters in the *infer.par* file, so they may be omitted:

i. start_time
ii. end_time
iii. netvisa_curr_model - This will be set to the same directory as the most recently trained model.

Parameters

- root_dir - Defaults to */netvisa_models*.
- infer_time_file - The file name which contains the timestamp for the start of the next inference window. Note that this file is updated by runner. Defaults to *next_infer_time.out*.
- training_days - The number days of data that are used for training a model. Defaults to 90.
- retrain_after_days - The number of days after which a model is retrained. Defaults to 7.
- infer_interval - The time interval for each interval run. Defaults to 1800 seconds, i.e. 30 minutes.
- infer_skip - The amount of time to move the inference interval forward. Defaults to 1500 seconds, i.e. 25 minutes.
- realtime_delay - Pause when we are within these many seconds of real time. Defaults to 21600 seconds, i.e. 6 hours.
- keep_n_models - Delete models after this many. If zero then don't delete any. Defaults to 10.
- infer_end_time - The end time for inference. If zero then keep running forever. Defaults to 0.

Algorithm In the following, we will assume the default values of all the parameters for the purpose of simplifying the description.

1. Look for a file */netvisa_models/next_infer_time.out*. This file should contain a single timestamp from which infrerence is to be started, let's call this NEXT_INFER_TIME.
2. Determine the current major and minor version of the NET-VISA software. For simplicity we will assume that this is 2.5.

3. Look for a model in */netvisa_models/2.5/* that was trained with an ending time within 7 days of NEXT_INFER_TIME. We assume that the model directory names would be of the form */netvisa_models/2.5/model-<start time>-<end time>*. Where the start and end times refer to the start and end times of the learning run.

4. If the model is not found in Step 3 then do the following:

   (a) Attempt to learn a model using a parameter file in */netvisa_models/2.5/learn.par*, and verify that the model is indeed created.

   (b) After creating the model attempt to delete the 10th oldest model if that directory exists. (Effectively, we will limit the total storage to ten models).

5. Now attempt to run inference starting from NEXT_INFER_TIME for 30 minutes using an inference parameter file in */netvisa_models/2.5/infer.par*.

6. Increase the inference start time by 25 minutes and update the file that was read in Step 1 (*/netvisa_models/next_infer_time.out*).

7. Repeat from Step 1.

If any of the steps above produces an error then the error is printed and the command retries the sequence from Step 1 after sleeping for 5 mintues. In addition, the following environment variables can be used to send out emails on any errors.

- SENDGRID_API_KEY - An API key obtained from Sendgrid for sending emails.
- STATUS_EMAIL_FROM - An email address which will appear in the from field of the email.
- STATUS_EMAIL_TO - The destination emails (comma separated) to whom the status emails will be sent.

If all three of the above environment variables are set then this script will send out an email on the following conditions.

- When the script is run for the first time.
- Whenever the script encounters an error.
- When the script ends.

8.9.11 IDCNETVISA_COMPARE_MODELS

The COMPARE_MODELS command can be used to verify that a newly trained model will produce event scores that are very similar to a previous model. The NET-VISA model is divided into multiple model elements and for each of these elements we estimate the average percentage change. If the confidence intervals for the mean of this estimated change is more than a permissible threshold then a failure is reported. The command returns an exit code of 1 on any failure and 0 on no failures.

In order to estimate the mean of the confidence interval we use a T distribution since both the true population mean and standard deviations are unknown.

Model Elements

- We first consider *event prior* which corresponds to the prior probability of the location of each event and its magnitude. This is further broken down to each of the three event mediums.
- Next we consider *event detection* which corresponds to the conditional detection probability of an arrival at a station and all of the arrival attributes conditioned on the associated event as well as the probabilities of the alternate explanations of noise or coda. This is further broken down to the level of each station.

Parameters

- netvisa_new_model - The new model to analyze.
- ref_origin_table, ref_assoc_table - Location of the reference bulletin whose events will be used to evaluate the two models.
- critical_level - The critical levels of the T distribution for estimating the confidence interval of the estimated change in each model element. Defaults to 5 (percent). Note that a value of 5 implies that we are looking for the 95% confidence interval.
- max_pct_change - The maximum permissible average percentage change in any model element. Defaults to 10 (percent).
- exclude_sta_list - A failure in this list of stations will be ignored and not considered an error.

The rest of the parameters are identical to IDCNETVISA_INFER (see Section 8.9.2). However, the following parameters have a slightly different meaning.

- netvisa_curr_model - The original model that will be used as a baseline to compare the new model with.
- start_time, end_time - The time range for which reference events will be queried to evaluate the two models on.

Sample Output

```
..
Reading detections from 1515348000 to 1515909600 in idcx.arrival
136159 arrivals; 102 guaranteed sites out of 247; avg. uptime 79.1%
136159 detections 30530 P 267 S 4115 T 260 H 4387 I 82001 tx+N
757 events read from leb.origin, leb.assoc with an avg of 9.7 dets per events
..
INFO: medium SEISMIC event_prior changed within 0.0, 0.0. OK
..
WARN: sta ARCES has event_detection values changed within -100.0, 37.6. Ignoring.
```

```
INFO: sta ASAR event_detection changed within -3.0, 1.7. OK
..
INFO: sta YKA event_detection changed within -9.4, -1.0. OK
ERROR: sta ZALV has event_detection values changed within interval -31.1, 3.5 at
critical level 5% and this is outside the permissible max 10.0
4 failures detected.
```

In the above output we first report some statistics on the numbers of arrivals read as well as the number of reference events. Next, for each model element we report the estimated change and whether this change is acceptable, a failure, or ignored. Finally, we output the total number of failures. Note that in the above output the station ARCES was included in the parameter EXCLUDE_STA_LIST and hence the failure for that station is ignored.

8.10 Docker Images

8.10.1 NET-VISA Image

The NET-VISA Docker image includes all the prerequisite software, such as IDC libraries, required to run NET-VISA. It also contains all of the scripts that are described in Section 8.9. The image is based on Red Hat Enterprise Linux (RHEL) 7.9 and it contains a Python 3.9 environment with the NET-VISA software preinstalled.

Image Contents The Python 3.9 environment is located at:

```
/home/misc/cmss/idc/py39env
```

In addition, the IDC libraries (cbase, idcmodel, and sbase) are available at the following location,

```
/home/misc/cmss/idc/install
```

The Python 3.9 environment is activated by default upon entry into the image. The Docker image is configured to run the IDCNETVISA_RUNNER command (see Section 8.9.10) without any arguments.

Image Location and Tags The Docker image is published to the IDC repository, Nexus, at the following location:

```
/ctbto/idc/shi/netvisa/netvisa
```

This image is tagged in multiple ways:

1. At a minimum, the Git SHA is applied to all images.
2. If a tag is present on GitLab, the version number from the tag is also used to label the image.
3. A special rc tag is applied to the latest version of the image.

Image Construction The NET-VISA Image is automatically built by Gitlab Pipelines on every push to master. The build process is defined in the following Dockerfile within the NET-VISA repository:

`build_scripts/Dockerfile.netvisa`

This Dockerfile uses the following script to install NET-VISA:

`build_scripts/build_netvisa_wheel.sh.`

8.10.2 PREREQ Image

The PREREQ image includes all the software required to install NET-VISA, excluding the NET-VISA software itself. It provides the Python 3.9 environment and the necessary IDC software. Refer to Section 8.10.1for the respective locations. In fact, the PREREQ image is used as the base image to build the NET-VISA image.

Image Location and Tags The current location and tag for the PREREQ image is:

`/ctbto/idc/shi/netvisa/netvisa-prereq-rh7-dev:latest`

Image Construction This image is built on top of a base image provided by IDC, located at:

`/ctbto/idc/generic/basedbbuild7_rhel:dev`

The process of creating the PRE-REQ image from the base image is defined in the following Dockerfile within the NET-VISA repository:

`build_scripts/Dockerfile.prereq`

This Dockerfile uses the following script to properly install the Python environment and IDC software:

`build_scripts/build_netvisa_prereq.sh.`

Note: A new PREREQ image is usually not required unless critical changes are made to the IDC libraries or new packages are needed for the Python environment to support NET-VISA. In such cases, the PREREQ image build is triggered manually via GitLab Pipelines by running the job titled:

`build_prereq_rh7_image`

9 Learning Schedule

It is recommended that the NET-VISA model parameters be relearned every three months using at least three months of data as a training set. IDCNETVISA_LEARN automatically verifies its results by holding out a subset of the data. If the new parameters do not outperform the existing set, then the old parameters will be retained.

10 Model Schema

The NET-VISA model parameters are written to a text file, MODEL.TXT, in the model directory. This file is a text-encoded database of the NET-VISA model. In the rest of this section we describe the schema of these tables. As of this writing mainly the learned seismic parameters are recorded in this file.

If a user wishes to inspect the contents of the table, the text file is directly readable. However, the user may optionally read this text file using sqlite3 database software as shown in the following example.

```
$> sqlite3 :memory:
sqlite> .read <full path to model.txt file>
sqlite> .tables
arrival_SNR                 num_false_det
arrival_SNR_header          phase
arrival_amp                 qc
arrival_azimuth             sec_det_azimuth
arrival_phase               sec_det_coda_coda_detprob
arrival_slow                sec_det_coda_coda_phase
arrival_time                sec_det_coda_coda_snr
event_detection             sec_det_detprob
event_mag                   sec_det_logamp
false_arrival_SNR           sec_det_phase_coda_detprob
false_arrival_amp           sec_det_phase_coda_phase
false_arrival_phase         sec_det_phase_coda_snr
false_arrival_slow          sec_det_slowness
hydro_frac_time             sec_det_snr
info                        sec_det_time
num_event                   site
sqlite> .headers ON
sqlite> select * from site where rowid<5;
siteid|site_name|longitude|latitude|elevation|isarray|isprimary|isseis|ishydro|isinfra
0|AAK|74.4942|42.6391|1.645|0|0|1|0|0
1|AFI|-171.7773|-13.9093|0.706|0|0|1|0|0
2|AKASG|29.2242|50.7012|0.16|1|1|1|0|0
3|AKTO|58.0164|50.4348|0.379|0|0|1|0|0
```

10.1 Arrival SNR

The SNR values of true arrivals are binned using bin size specified in arrival_snr_header table.

ARRIVAL_SNR	
phaseid	The phase identifier of the true phase.
binidx	The bin index
value	The probability of SNR value being in this bin.

10.2 Arrival SNR header

This table specifies the SNR bin size that is used in other tables that refer to SNR values.

ARRIVAL_SNR_HEADER	
snr_step	The size of the SNR bins.

10.3 Arrival Amplitude

The arrival amplitude is a linear model per station and true phase. This table specifies the coefficients of the linear model.

ARRIVAL_AMPLITUDE	
siteid	The station identifier.
phaseid	The phase identifier.
intercept	The intercept of the linear model.
mb	The coefficient of the mb value of the linear model.
Q	The coefficient of the Q value (from Veith-Clawson tables) of the linear model.
sigma	The standard error of the linear model.

10.4 Arrival Azimuth

True arrival azimuths are modeled as a per-station and per-phase Laplace distribution with parameters location and scale.

ARRIVAL_AZIMUTH	
siteid	The station identifier.
phaseid	The phase identifier.
loc	The location parameter of the Laplace distribution.
scale	The scale parameter of the Laplace distribution.

10.5 Arrival Phase

The observed label (iphase) of a true arrival is modeled seprately for array stations and single-site (or 3-component) stations.

ARRIVAL_PHASE	
true_label	The true phase identifier.
obs_label	The observed phase identifier (iphase).
arr_prob	The probability of this observed phase at an array station.
ss_prob	The probability of this observed phase at a single-site station.

10.6 Arrival Slowness

True arrival slowness values are modeled as a per-station and per-phase Laplace distribution with parameters location and scale.

ARRIVAL_AZIMUTH	
siteid	The station identifier.
phaseid	The phase identifier.
loc	The location parameter of the Laplace distribution.
scale	The scale parameter of the Laplace distribution.

10.7 Arrival Time

True arrival time is modeled as a per-station and per-phase Laplace distribution with parameters location and scale.

ARRIVAL_AZIMUTH	
siteid	The station identifier.
phaseid	The phase identifier.
loc	The location parameter of the Laplace distribution.
scale	The scale parameter of the Laplace distribution.

10.8 Event Detection

Event detection probability is modeled as a per-medium, per-station, and per-true phase logistic regression. This table lists the coefficients of the regression.

EVENT_DETECTION	
medium	The medium identifier of the event.
siteid	The station identifier.
phaseid	The phase identifier.
intercept	The intercept of the logistic regresion.
mag	The coefficient of the event magnitude feature.
depth	The coefficient of the event depth feature.
dist	The coefficient of the event-station distance feature.
dist0	The coefficient of the dist0 feature.
dist35	The coefficient of the dist35 feature.
dist40	The coefficient of the dist40 feature.
dist12520	The coefficient of the dist12520 feature.
dist12540	The coefficient of the dist12540 feature.
mag6	The coefficient of the mag6 feature.
mag68	The coefficient of the mag68 feature.
md	The coefficient of the event magnitude-depth feature.
oop_angle	The coefficient of the out-of-plane angle feature.
arr_henergy	The coefficient of the hydro arrival energy feature.

10.9 Event Magnitude

The event magnitude is an exponential distribution parameterized with a rate and a minimum value.

EVENT_MAG	
min_magnitude	The minimum allowed magnitude.
mb_rate	The rate parameter.

10.10 SNR of Noise Arrivals

The probability of noise SNR is binned into bins of size given by the table arrival_SNR_header.

FALSE_ARRIVAL_SNR	
binidx	The bin index.
value	The probability of the SNR value being in this bin.

10.11 Amplitude of Noise Arrivals

Noise arrival amplitude is modeled as a mixture of two Normal distributions, which are parameterized by a mean and standard deviation.

FALSE_ARRIVAL_AMP	
siteid	The site identifier.
wt0	The weight of first Normal distribution.
wt1	The weight of the second Normal distribution.
mean0	Mean of first Normal distribution.
mean1	Mean of second Normal distribution.
std0	Standard deviation of first Normal distribution.
std1	Standard deviation of second Normal distribution.

10.12 Phase of Noise Arrivals

The noise arrival phase probability is computed for each station medium and observed phase label.

FALSE_ARRIVAL_PHASE	
medium	The medium identifier for the station.
obs_label	The phase identifier of the observed label (iphase).
prob	The probability of observing this noise label.

10.13 Slowness of Noise Arrivals

Noise arrival slowness values are modeled by a Laplace distribution with a location and scale per phase and per station medium.

FALSE_ARRIVAL_SLOW	
phaseid	The phase identifier.
mediumid	The station medium identifier.
loc	Location parameter.
scale	Scale parameter.

10.14 Hydro Fraction Time

Hydro Fraction time above detection threshold is modeled as two Beta distributions. One for real arrivals and one for noise. The Beta distribution is parameterized by two values a and b.

HYDRO_FRAC_TIME	
true_a	a-value for Beta distribution of real arrivals
true_b	b-value for Beta distribution of real arrivals
false_a	a-value for Beta distribution of noise arrivals
false_b	b-value for Beta distribution of noise arrivals

10.15 Info

The info table provides some basic information about the model.

INFO	
start_time	The start time for training the current model.
end_time	The end time for training the current model.
nvisa_version	The NET-VISA version.
nvisa_location	The location of the NET-VISA software .

10.16 Rate of Events

NUM_EVENT	
seismic_rate	Rate of occurrence of seismic events per second.
hydro_rate	Rate of occurrence of under-water events per second.
infra_rate	Rate of occurrence of atmospheric events per second.

10.17 False Detection Rate

NUM_FALSE_DET	
siteid	The site identifier.
false_rate	The number of false detections per second.

10.18 Phase

The phase table lists all the phases that are supported by the NET-VISA. The phaseid in this table corresponds to the phaseids in many of the other tables.

PHASE	
phaseid	The phase identifier (0-based).
phase	The phase string, e.g. P, S, PKP, N, tx, etc.
istimedef	1 => "is time defining"; 0 => noise or coda.
prop_medium	The propagation medium (0 seismic, 1 hydro, 2 infra).

Notes:

1. All of the time-defining phases have smaller phase identifiers than any of the non-time-defining phases.

10.19 QC

The QC table records the historical noise level at each station. If these values are NaN then we don't have a noise model and any QC noise data is ignored.

QC	
siteid	The site identifier.
p50	The 50th percentile noise level at the station.
p70	The 70th percentile noise level at the station.
p80	The 80th percentile noise level at the station.
p90	The 90th percentile noise level at the station.
p95	The 95th percentile noise level at the station.

10.20 Coda Model

The coda model parameters describe the generative model of coda arrivals. In general there are two kinds of coda arrivals. The PHASE-CODA arrivals are those that immediate follow a true arrival, while the CODA-CODA arrivals are those that follow another coda arrival.

10.20.1 Coda Arrival Azimuth

Coda arrival azimuth is given by a Laplace distribution which is parameterized by location and scale values.

SEC_DET_AZIMUTH	
phase_coda_location	Laplace location for a coda following a real arrival.
phase_coda_scale	Laplace scale for a coda following a real arrival.
coda_coda_location	Laplace location for a coda following a coda arrival.
coda_coda_scale	Laplace scale for a coda following a coda arrival.

10.20.2 Coda Detection Probability

The probability that a coda arrival is detected is a function of the log amplitude of the preceding arrival whether real or itself a coda. We discretize this probability using the log amplitude of the preceding arrival.

SEC_DET_DETPROB	
num_bins	The number of bins of log amplitude value.
min_logamp	The log amplitude corresponding to the first bin.
step	The step size of the bins.

SEC_DET_PHASE_CODA_DETPROB	Coda following a real arrival
bin	The bin number from 0 to num_bins.
value	The probability of detecting a coda.

SEC_DET_CODA_CODA_DETPROB	Coda following a coda arrival
bin	The bin number from 0 to num_bins.
value	The probability of detecting a coda.

10.20.3 Coda Arrival Phase

The coda arrival phase is modeled as a discrete distribution with one bin per phase. As before we learn two distinct distributions.

SEC_DET_PHASE_CODA_PHASE	Coda following a real arrival.
phaseid	The phase identifier, see 10.18.
probability	Probability of generating this phase.

SEC_DET_CODA_CODA_PHASE	Coda following a coda arrival.
phaseid	The phase identifier, see 10.18.
probability	Probability of generating this phase.

10.20.4 Coda Arrival SNR

The coda SNR value is stored as a discrete probability distribution.

SEC_DET_SNR	
step	The step size of the distribution.
num_bins	The number of bins in the distribution

SEC_DET_PHASE_CODA_SNR	SNR distribution of coda following a real arrival
bin	The bin number from 0 to num_bins.
value	The probability of SNR being in this bin.

SEC_DET_CODA_CODA_SNR	SNR distribution of coda following a coda arrival
bin	The bin number from 0 to num_bins.
value	The probability of SNR being in this bin.

10.20.5 Coda Arrival Amplitude

Coda arrival log amplitude is given by a Laplace distribution which is parameterized by location and scale values.

SEC_DET_LOGAMP	
phase_coda_location	Laplace location for a coda following a real arrival.
phase_coda_scale	Laplace scale for a coda following a real arrival.
coda_coda_location	Laplace location for a coda following a coda arrival.
coda_coda_scale	Laplace scale for a coda following a coda arrival.

10.20.6 Coda Arrival Slowness

Coda arrival slowness is given by a Laplace distribution which is parameterized by location and scale values.

SEC_DET_SLOWNESS	
phase_coda_location	Laplace location for a coda following a real arrival.
phase_coda_scale	Laplace scale for a coda following a real arrival.
coda_coda_location	Laplace location for a coda following a coda arrival.
coda_coda_scale	Laplace scale for a coda following a coda arrival.

10.20.7 Coda Arrival Time

The arrival time of a coda is given by a Gamma distribution between the values min_delay and max_delay. The shape and the scale of the gamma is specified in the following table.

SEC_DET_TIME	
min_delay	The minimum time after a preceding arrival.
max_delay	The maximum time after a preceding arrival.
phase_coda_shape	Gamma shape parameter for a coda following a real arrival
phase_coda_scale	Gamma scale parameter for a coda following a real arrival
coda_coda_shape	Gamma shape parameter for a coda following a coda arrival
coda_coda_scale	Gamma scale parameter for a coda following a coda arrival

10.21 Site

The site table records the names and of all the stations for which a model is learned. Each station is assigned a siteid identifier which is used in all the other model tables. Additionally, attributes of the station are recorded.

SITE

siteid	The site identifier (0-based)
site_name	The station name string.
longitude	Longitude.
latitude	Latitude.
elevation	Elevation above sea-level.
isarray	Is this an array station (0 => No, 1 => Yes)
isprimary	Is this a primary station (0 or 1).
isseis	Is this a seismic station (0 or 1).
ishydro	Is this a hydro-acoustic station.
isinfra	Is this an infrasound station.

Notes:

1. Hydro-acoustic stations that are not arrays correspond to the so-called T-stations, which use coastal seismic sensors to record hydro-acoustic waves. In other words, ISHYDRO=1 and ISARRAY=0 correspond to T-stations and ISHYDRO=1 and ISARRAY=1 correspond to hydro-acoustic stations.

11 Parquet Data File Schema

As part of parquet-based learning the learn script generates a number of data files that are subsequently used for training a model. There are four parquet files and two csv files. This section describes the schema of these files.

In these tables the term residual refers to the difference between some measured quantity and the theoretical value of that same quantity.

11.1 Assoc Parquet File (assoc.parquet)

Contains information about all event origins and the arrivals associated to them. Note that this table contains one row per association and as a consequence if an event is associated with more than one arrival then that event will be duplicated in this table.

Column Name	Description
orid	Origin ID of the event origin associated with the arrival
origin_evid	Event ID of the event

origin_lat	The latitude of the origin
origin_lon	The longitude of the origin
origin_time	The timestamp of the origin
origin_ndef	The number of defining phases of the origin
arid	Arrival ID of the associated arrival
arrival_time	Arrival timestamp
arrival_deltim	The uncertainty in the arrival timestamp
arrival_azimuth	The measured azimuth of the arrival
arrival_delaz	The uncertainty in the azimuth measurement
arrival_slow	The measured slowness of the arrival
arrival_delslo	The uncertainty in the arrival slowness
arrival_iphase	The initial phase identification of the arrival
arrival_snr	The measured signal-to-noise ratio of the arrival
arrival_amp	The measured amplitude of the arrival
arrival_per	The measured period of the arrival
arrival_stassid	The station specific arrival identifier labeled by StaPro
assoc_phase	The phase name associated with this arrival
assoc_timedef	Is the associated arrival considered time-defining?
assoc_azdef	Is the associated azimuth considered time-defining?
assoc_slodef	Is the associated slowness considered time-defining?
assoc_delta	The distance between the station and the event origin
assoc_timeres	The time residual of the arrival
assoc_azres	The azimuth residual of the arrival
assoc_slores	The slowness residual of the arrival
theoretical_arrival_time	Predicted time of the arrival given the event origin
theoretical_arrival_azimuth	Predicted azimuth of the arrival
theoretical_arrival_slow	Predicted slowness of the arrival
theoretical_oop_angle	Predicted out-of-plane angle computed for hydro stations
theoretical_henergy_drop	Predicted drop in hydro energy for hydro stations
theoretical_qfactor	Predicted Veith Clawson Q Factor for seismic stations
sta	Station name
site_lon	Station longitude
site_lat	Station latitude
site_elev	Station elevation

11.2 Arrival Parquet File (arrival.parquet)

Contains information about all of the arrivals as well as the immediately preceding arrival at the same station within 50 seconds if any.

Column Name	Description
arid	Arrival ID
arrival_time	Arrival timestamp
arrival_deltim	The uncertainty in the arrival timestamp
arrival_azimuth	The measured azimuth of the arrival
arrival_delaz	The uncertainty in the azimuth measurement
arrival_slow	The measured slowness of the arrival
arrival_delslo	The uncertainty in the arrival slowness
arrival_iphase	The initial phase identification of the arrival
arrival_snr	The measured signal-to-noise ratio of the arrival
arrival_amp	The measured amplitude of the arrival
arrival_per	The measured period of the arrival
arrival_stassid	The station specific arrival identifier labeled by StaPro
arrival_lddate	The datetime when this arrival was loaded in the IDC database the first time
ranking	A sequential rank given to each arrival at a station based on its timestamp
sta	Station name
site_lon	Station longitude
site_lat	Station latitude
site_elev	Station elevation
(optional)	The following columns are optionally populated if there is an event associated with this arrival
orid	Origin ID of the associated event origin
assoc_phase	The phase name associated with this arrival
assoc_timedef	Is the associated arrival considered time-defining?
assoc_azdef	Is the associated azimuth considered time-defining?
assoc_slodef	Is the associated slowness considered time-defining?
(optional)	The following columns are optionally populated if there is an immediately preceding arrival at the same station within 50 seconds
prev_arid	Previous arrival ID
prev_sta	Station name of the previous arrival
prev_arrival_time	Previous arrival timestamp

prev_arrival_deltim	The uncertainty in the arrival timestamp of the previous arrival
prev_arrival_azimuth	The measured azimuth of the previous arrival
prev_arrival_delaz	The uncertainty in the previous arrival's azimuth measurement
prev_arrival_slow	The measured slowness of the previous arrival
prev_arrival_delslo	The uncertainty in the previous arrival's slowness
prev_arrival_iphase	The initial phase identification of the previous arrival
prev_arrival_snr	The measured signal-to-noise ratio of the previous arrival
prev_arrival_amp	The measured amplitude of the previous arrival
prev_arrival_per	The measured period of the previous arrival
prev_arrival_stassid	The station specific arrival identifier labeled by StaPro of the previous arrival
prev_arrival_lddate	The datetime when the previous arrival was loaded in the IDC database the first time
prev_site_lon	Station longitude of the previous arrival
prev_site_lat	Station latitude of the previous arrival
prev_site_elev	Station elevation of the previous arrival
prev_orid	The optional origin ID associated with the previous arrival
prev_assoc_phase	The phase name associated with the previous arrival
prev_assoc_timedef	Is the associated previous arrival considered time-defining?
prev_assoc_azdef	Is the associated previous azimuth considered time-defining?
prev_assoc_slodef	Is the associated previous slowness considered time-defining?
prev_ranking	The sequential rank given to the previous arrival

11.3 Site_Up Parquet File (site_up.parquet)

The site_up parquet file tracks for each station and for each 30 minute time interval whether the station was up during that time interval. For this purpose a site is considered up in an interval if there is any arrival detected within that time interval or in the immediately preceding or succeeding time interval.

In this table, the first column is the START_TIME timestamp and the remaining columns are the names of each station. Each row represents a 30 minute time interval starting at the START_TIME value for that row and the value for a station column is a boolean which indicates whether that station was up during that time interval.

11.4 Event_Det Parquet File (event_det.parquet)

The event_det parquet file tracks event detections (or non-detection) for each origin-station-phase combination. Note that if a particular phase from an event couldn't possibly be detected at a station then there won't be an entry for that event-phase-station.

Column Name	Description
orid	The ID of the event's origin
sta	The station name
phase	The seismic, hydro-acoustic, or infrasound phase that could potentially have been detected at the station
mb	The body-wave magnitude of the event
depth	The depth of the event
delta	The distance between the event and the station
oop_angle	The out-of-plan angle between the event and the station (for hydro station's only)
sta_henery	The hydro energy meaasure at the station (for hydro stations's only)
sitemedium	The site medium
isarray	Is that station an array station?
is_det	Detection flag, 1 if detected, 0 otherwise

11.5 Site and Phase CSV files (phase.csv and site.csv)

Two additional CSV files are generated tracking information about the phase and the stations.

The phase file has the following columns. It has exactly one row per phase that is supported by NET-VISA.

Column Name	Description
phaseid	The unique identifier for each phase
name	Phase name
is_timedef	Indicates if the phase is time-defining
propmedium	Propagation medium for the phase

The site file has the following columns. It has exactly one row per site.

Column Name	Description
siteid	Unique identifier for each site
sta	Station name
lon	Station longitude
lat	Station latitude
elev	Station elevation
isarray	Indicates if the station is an array

ishydro	Indicates if the station is a hydro station
isinfra	Indicates if the station is an infra station
ondate	The latest julian date when the station was operational
sitemedium	The medium type of the station

12 Event Bulletin Database Schema

The tables in this section give the database schema for the event bulletin that is produced by IDCNETVISA. All except the last three tables are standard IDC tables which are documented in IDC literature.

EVID	number (10) not null
EVNAME	VARCHAR2 (15)
PREFOR	number(10)
COMMID	number(10)
LDDATE	DATE

Table 7 – EVENT table

lat	float (24) not null
lon	float (24) not null
time	float (53) not null
orid	number (10) not null
evid	number (10)
jdate	number (8)
nass	number (4)
ndef	number (4)
ndp	number (4)
GRN	number (8)
SRN	number (8)
ETYPE	varchar2 (7)
DEPDP	float (24)
DTYPE	varchar2 (1)
m_b	float (24)
m_bID	number (10)
MS	float (24)
MSID	number (10)
ML	float (24)
MLID	number (10)
ALGORITHM	varchar2 (15)
AUTH	varchar2 (15)
COMMID	number (10)
LDDATE	date

Table 8 – ORIGIN table

ORID	number (10)
SXX	float (24)
SYY	float (24)
SZZ	float (24)
STT	float (24)
SXY	float (24)
SXZ	float (24)
SYZ	float (24)
STX	float (24)
STY	float (24)
STZ	float (24)
SDOBS	float (24)
SMAJAX	float (24)
SMINAX	float (24)
STRIKE	float (24)
SDEPTH	float (24)
STIME	float (24)
CONF	float (24)
COMMID	number (10)
LDDATE	date

Table 9 – ORIGERR table

ARID	number (10) not null
ORID	number (10) not null
STA	varchar2 (6)
PHASE	varchar2 (8)
BELIEF	float (24)
DELTA	float (24)
SEAZ	float (24)
ESAZ	float (24)
TIMERES	float (24)
TIMEDEF	varchar2 (1)
AZREZ	float (24)
AZDEF	varchar2 (1)
SLORES	float (24)
SLODEF	varchar2 (1)
EMARES	float (24)
WGT	float (24)
VMODEL	varchar2 (15)
COMMID	number (10)
LDDATE	date

Table 10 – ASSOC table

MAGID	number(10) not null
NET	VARCHAR2(8)
ORID	number(10) not null
EVID	number(10)
MAGTYPE	VARCHAR2(6)
NSTA	number(8)
MAGNITUDE	FLOAT(24)
UNCERTAINTY	FLOAT(24)
AUTH	VARCHAR2(15)
COMMID	number(10)
LDDATE	DATE

Table 11 – NETMAG table

MAGID	number(10) not null
AMPID	number(10) not null
STA	VARCHAR2(6) not null
ARID	number(10)
ORID	number(10)
EVID	number(10)
PHASE	VARCHAR2(8)
DELTA	FLOAT(24)
MAGTYPE	VARCHAR2(6)
MAGNITUDE	FLOAT(24)
UNCERTAINTY	FLOAT(24)
MAGRES	FLOAT(24)
MAGDEF	VARCHAR2(1)
MMODEL	VARCHAR2(15)
AUTH	VARCHAR2(15)
COMMID	number(10)
LDDATE	DATE

Table 12 – STAMAG table

ORID	number(10) not null
EVID	number(10)
PREFER_LOC	VARCHAR2(1)
CONSTRAIN_OT	number(1)
CONSTRAIN_LATLON	number(1)
CONSTRAIN_DEPTH	number(1)
SRC_DPNT_CORR	number(2)
LOC_SRC_DPNT_REG	VARCHAR2(15)
LOC_SDV_SCREEN	number(1)
LOC_SDV_MULT	FLOAT(24)
LOC_ALPHA_ONLY	number(1)
LOC_ALL_STAS	number(1)
LOC_DIST_VARWGT	number(1)
MAG_SRC_DPNT_REG	VARCHAR2(15)
MAG_SDV_SCREEN	number(1)
MAG_SDV_MULT	FLOAT(24)
MAG_ALPHA_ONLY	number(1)
MAG_ALL_STAS	number(1)
m_b_MIN_DIST	FLOAT(24)
m_b_MAX_DIST	FLOAT(24)
MMODEL	VARCHAR2(15)
COV_SM_AXES	FLOAT(24)
COV_DEPTH_TIME	FLOAT(24)
LDDATE	DATE

Table 13 – EVENT_CONTROL table

keyname	varchar2(15) not null
keyvalue	number(10) not null
lddate	date

Table 14 – LASTID table

Column Name	Data Type	Description
lat	float(24)	Latitude
lon	float(24)	Longitude
depth	float(24)	Depth
time	float(53)	Event time as a UNIX timestamp
orid	number(10)	Origin identifier joins with origin table
evid	number(10)	The event identifier joins with event table
mb	float(24)	Body-wave magnitude
evmedium	number(10)	The medium in which the event occurred (0 -> ground, 1 -> water, 2 -> air).
evscore	float(24)	NET-VISA event log probability minus log probability of detections being caused by noise.
evscore_kill	float(24)	NET-VISA event log probability minus log probability of detections being caused by other events.
score_num	float(24)	Log probability of the event time and overall number.
score_loc	float(24)	Log probability of the event location.
score_mag	float(24)	Log probability of the event magnitude.
detscore	float(24)	Log probability of the detections minus log probability of those detections being caused by noise.
misdetscore	float(24)	Log probability of all the possible detections that are missed.
num_det	number(10)	Number of detections.
poss_num_det	number(10)	The possible number of detections based on the stations that were up and the phases that are detectable at those stations.
lddate	date	Date that the event was stored in the database.

Table 15 – VISAORIGIN table

Column Name	Data Type	Description
arid	number(10)	Arrival identifier joins with arrival table.
orid	number(10)	Origin identifier joins with visaorigin or visabirth table table
sta	varchar2(6)	Station code.
phase	varchar2(8)	Name of phase.
lddate	date	Date that the association was stored in the database.

Table 16 – VISAASSOC table

Column Name	Data Type	Description
lat	float(24)	Latitude
lon	float(24)	Longitude
depth	float(24)	Depth
time	float(53)	Event time as a UNIX timestamp
orid	number(10)	The origin identifier.
evid	number(10)	The event identifier joins with the event table
mb	float(24)	Body-wave magnitude
evmedium	number(10)	The medium in which the event occurred (0 -> ground, 1 -> water, 2 -> air).
evscore	float(24)	NET-VISA event log probability minus log probability of detections being caused by noise.
evscore_kill	float(24)	Not used.
score_num	float(24)	Log probability of the event time and overall number.
score_loc	float(24)	Log probability of the event location.
score_mag	float(24)	Log probability of the event magnitude.
detscore	float(24)	Log probability of the detections minus log probability of those detections being caused by noise.
misdetscore	float(24)	Log probability of all the possible detections that are missed.
num_det	number(10)	Number of detections.
poss_num_det	number(10)	The possible number of detections based on the stations that were up and the phases that are detectable at those stations.
lddate	date	Date that the event was stored in the database.

Table 17 – VISABIRTH table

13 Release Management at the IDC

13.1 Version Numbers

The NET-VISA software is versioned with a sequentially increasing version number to help identify the specific version that was used to build a bulletin. The release numbers have the following convention (as suggested by Semantic Versioning, see semver.org):

MAJOR.MINOR.PATCH

For example version 2.5.14 has major version 2, minor version 5 and patch version 14. Most changes in the software only trigger a change in the patch version while the minor version is bumped up when an incompatibility is introduced in the model file storage format. Further, a release candidate version has an *rc* suffix followed by the release candidate number, for example *2.5.14rc25*.

Further, there is a provision of post-release versions. Such version labels are used to tag changes to the software that don't materially affect either learning or inference. An example

could be *2.5.14.post3*. A post-release suffix could also be added to a release candidate version, for example *2.5.15rc3.post2*.

GIT Tags The git tag for a version has a prefix of *netvisa-* followed by the software version, for example *netvisa-2.5.14rc23* or *netvisa-2.5.15.post2*.

Author The NET-VISA software records its version in the AUTHOR field to help in debugging when problems arise. Given the space constraints for this field, the word *netvisa* is shortened to *nv*, for example *nv-2.5.15rc1*.

Post-release suffixes in the version are not recorded in the author field. For example the author field for an origin produced by NET-VISA version *2.5.15.post2* would be *2.5.15*.

13.2 Testing and Deployment

All changes made to the software and pushed to an upstream branch in git, trigger a testing pipeline in Gitlab which hosts the git repository for NET-VISA. Upon successful completion of the testing a Python wheel and a Docker image containing the NET-VISA software is automatically published to the package mangement service known as Nexus which is hosted internally at the IDC. Next, the NET-VISA software is deployed to a number of environments.

13.2.1 Release Candidate

On every commit to the MASTER branch a new NET-VISA image is created with tag RC as described in 8.10.1. This image is pushed to a special machine, currently DLV122, which is responsible for running the release candidate code. The results of this service are recorded in the VISA@ODB database schema.

This release candidate machine provides early results of the performance of the latest versions of NET-VISA on the operations data. The corresponding bulletin is running only in SEL3-mode using the existing automatic arrivals.

13.2.2 DEVLAN and OPS

When a new release tag or post-release tag is created on the MASTER branch a JIRA ticket is automatically created to push the corresponding NET-VISA version to the DEVLAN, which is the internal development environment for all newly released software. This ticket is then picked up by the IDC staff to publish the NET-VISA Python wheel. Later, IDC staff publish the new NET-VISA release to other environments such as TESTBED and OPS.

In the DEVLAN, NET-VISA is the default associator and hence its results are visible in the SEL1, SEL2, and SEL3 schemas in the IDCDEV database. In OPS (as of this writing NET-VISA hasn't been deployed in full production) the results of NET-VISA is made available in the VSEL3@ODB database schema from where it is picked up by the NET-VISA button in ARS (a tool used by analysts to review automatic bulletins).

13.2.3 QANDEF

For internal review there is another area where NET-VISA results are deployed and this is known as QANDEF. In the database *dba1623.qandef* NET-VISA is running as the primary associator in full 3-pipleine more and its results may be found in the SEL1, SEL2, and SEL3 schemas. The results of GA may be found in the ASEL3 schema. A full 3-pipeline testing involves simulating production pipelines wherein the results of SEL1 lead to the requests for auxiliary stations whose detections are used in the later stages such as SEL2 and SEL3. In theory, DEVLAN is also a full 3-pipeline test but its station and other configuration tends to often deviate from OPS and a resulting bulletin on devlan can't be used to compare to a bulletin from OPS. qandef aims to close this gap by providing full 3-pipeline results for NET-VISA in a configuration as close to OPS as possible.

A List of Symbols

Symbol	Meaning
e	event
A	Arrival that is not associated or designated false or coda
Λ	arrival associated with event
ξ	false arrival
κ	coda arrival
T	Arbitrary length of time
	superscripts
i	i^{th} event
j	j^{th} phase
k	k^{th} station
$\Delta_{ik}.$	Angular distance between the i^{th} event and the k^{th} station
s_l	latitude, longitude location of a station
I	ISAPEI prediction. Subscript indicates attribute of arrival that is being predicted.
G_z	Geographic function which predicts the azimuth of an arrival
r	residual (difference between measured and predicted attributes of an arrival)
ζ	null value; the value of an expected arrival which was not detected is set to null
	PDFs
$P(\cdot)$	Probability Density Function PDF
$N(\cdot)$, μ, σ	Gaussian (Normal) distribution (mean μ and standard deviation σ)
$\Gamma(\cdot)$, ρ, ν	Gamma distribution (shape ρ and rate ν)
β	Rate of an inverse gamma distribution.
$L(\cdot)$	Laplace distribution (mean μ and scale b)
	Event attributes
t	event time
l	event location (latitude, longitude)
d	event depth (km)
m	event magnitude mB
	Arrival attributes
t	arrival time
a	arrival amplitude
z	arrival azimuth
s	arrival slowness
h	arrival phase label
A^q	Triggering arrival for a coda arrival
	Logical operators

Symbol	Meaning
$\mathbb{1}(\cdot)$	Indicator function. If the relationship in the parenthesis is true, then $\mathbb{1}(\cdot)$ =1 otherwise $\mathbb{1}(\cdot)$ =0.
$\oplus$	exclusive or
	Notation for Inference Algorithm
S	Score for an event or arrival
$\Upsilon(\Lambda)$	Probability of an unassociated arrival; the probability that the arrival is false or coda, whichever is larger.

B Algorithms

Algorithm 1 Inference

Inside a window, $t_0 = 0; t_1 = W; t_2 = W + M_T$.

1. A default classification is given to arrivals in t_0 to t_2
2. Add events from BIRTH-MOVE(t_0, t_1, $\{a : t_0 \leq a_t \leq t_2\}$)
3. repeat N times (N = ?)
   (a) for each arrival a, s.t. $t_0 \leq a_t \leq t_2$, IMPROVE-ARRIVAL(a)
   (b) for each event e^i, s.t. $t_0 \leq e^i_t \leq t_1$, IMPROVE-EVENT(e^i)
4. for all events e^i, DEATH-EVENT(e^i)
5. for each arrival a, s.t. $t_0 \leq a_t \leq t_2$, IMPROVE-ARRIVAL(a)
6. for each event e^i, s.t. $t_0 \leq e^i_t \leq t_1$, IMPROVE-EVENT(e^i)
7. $t_0 += S; t_1 += S; t_2 += S$
8. output e^i, Λ^{ijk} for all e^i s.t. $e^i_t < t_0 - M_T$

Algorithm 2 Birth-Move

1. given t_0, t_1, and arrivals A
2. repeat for each a in A
   (a) invert a to obtain a candidate location ι_a
   (b) repeat for each location e in a ball around ι_a
      i. initialize Λ^e
      ii. repeat for each arrival a in A (let k be a's station)
         A. determine the phase j with the max score $S_d^{jk}(a \mid e)$
         B. if $S_d^{jk}(a \mid e) > S_d^{jk}(\Lambda^{jk} \mid e)$ or if $\Lambda^{jk} = \zeta$ and $S_d^{jk}(a \mid e) > 1$ then set $\Lambda^{jk} = a$.
   (c) let e be the event with the max score $S_e(e)$
   (d) repeat 100 times
      i. IMPROVE-EVENT(e)
      ii. IMPROVE-ARRIVAL(a) for all a in A with e as the only potential event
   (e) if $S_e(e) > 1$ then set aside event e and remove arrivals in Λ^{ejk} from A
3. return set aside events

Algorithm 3 Improve-arrival

1. given arrival a at station k
2. repeat for each event e
   (a) determine the phase j with the max score $S_d^{jk}(\Lambda^{ejk} \mid e)$
3. let e be the event with the max score $S_d^{jk}(\Lambda^{ejk} \mid e)$
4. if $S_d^{jk}(a \mid e) > S_d^{jk}(\Lambda^{ejk} \mid e)$ or if $\Lambda^{ejk} = \zeta$ and $S_d^{jk}(a \mid e) > 1$ then set $\Lambda^{ejk} = a$.

Algorithm 4 Improve Events

1. For each event e
   (a) Sample 100 events $e^i$from a uniform ball around the event (2 degrees latitude and longitude, 100 km depth, 5 seconds in time and 2 units of magnitude).
   (b) for each sample event\
      i. Compute its score$S_e\left(e^i\right)$
   (c) Return the event which has the highest score

C Debugging Missed Events

Overview

This appendix provides a comprehensive guide for debugging missed events in the NET-VISA system. Missed events are typically identified through bulletin comparisons and can be investigated using a systematic workflow that includes comparison analysis, detailed scoring diagnostics, and inference verification. This guide covers the complete lifecycle of debugging a missed event from initial detection through root cause analysis.

C.1 Bulletin Matching and Missed Event Detection

The first step in debugging missed events is to perform a bulletin matching between the bulletin being evaluated, which we will assume is in the VSEL3 schema and a reference bulletin, which we will assume is in the LEB schema.. These assumptions of the schema name are purely for simplicity and the instructions here can be easily modified to the user's needs. The IDCNETVISA_MATCH utility matches event lists from two bulletins and identifies discrepancies, including missed events, spurious events, and other differences.

C.1.1 Running the Matching

The bulletin matching is executed using the IDCNETVISA_MATCH command with a parameter file that specifies the comparison configuration.

Command The following command will run the match script for a specified time range and with the verbose flag set to true. The verbose flag provides very detailed information about which events are missed as opposed to simple statistics.

```
idcnetvisa_match par=match.par start_time=1515777000 end_time=1515778800 verbose=1
```

C.1.2 Parameter File Configuration

Below is an example parameter file (comparison.par) for comparing VSEL3 against LEB. In this file the environment variable ORACLE_CONN is assumed to have been assigned to a valid connection string to the oracle database hosting the two bulletins.

Parameter file (match.par)

```
include_seismic=1
include_infra=1
include_hydro=1
bull1_db_account=$(ORACLE_CONN)
bull1_origin_table=leb.origin
bull1_assoc_table=leb.assoc
arrival1_table=idcx.arrival
bull2_db_account=$(ORACLE_CONN)
bull2_origin_table=vsel3.origin
bull2_assoc_table=vsel3.assoc
bull2_origerr_table=vsel3.origerr
arrival2_table=idcx.arrival
min_assoc=2
rename_phases=1
max_arrival_lag=
```

Note: max_arrival_lag is left blank on the last line to ensure that no arrivals are filtered based on when the arrival was built.

C.1.3 Understanding Match Output

Consider the following output from the match script

```
idcnetvisa 2.5.17rc2 /home/misc/cmss/idc/py39env/...
Start time: 1515777000 ; 2018-01-12 17:10:00
End time: 1515778800 ; 2018-01-12 17:40:00
SELECT ...
SELECT ...
SELECT ...
SELECT ...
---------- Bulletin 1: leb.origin origin table: leb.origin ----------
7 origins (21 assocs, 19 def assocs); 6 Matchable Origins
---------- Bulletin 2: vsel3.origin origin vsel3.origin ----------
3 origins (13 assocs, 9 def assocs); 3 Matchable Origins
15342681 --> 15292276 [] [match]
15336857 --> unmat [] (1 assocs)
15341305 --> [] (2 assocs) [miss]
15314322 --> [] (2 assocs) [miss]
```

```
15342745 --> [] (3 assocs) [miss]
--> 15292332 (2 assocs) [spurious]
15314321 --> [] (5 assocs) [miss]
15342962 --> 15292333 [] [match]
Recall 33.3 % Precision 66.7 % F1 44.4 %
Missed 4 Spurious 1 (split 0)
phase cnt:num missed events - 2:2, 3:1, 5:1
phase cnt:num spurious events - 2:1
distance error (km), mean 165.0 std 78.9
by ndef: 2-3: 86.1+-0.0 4-5: 243.9+-0.0
depth error (km), mean 0.0 std 0.0
time error (sec), mean 12.8 std 6.6
mb< 0 : 0 / 2 mb< 1 : 0 / 0 mb< 2 : 0 / 0 mb< 3 : 0 / 1
mb< 4 : 2 / 3 mb< 5 : 0 / 0 mb>= 5 : 0 / 0
ML< 0 : 1 / 3 ML< 1 : 0 / 0 ML< 2 : 0 / 0 ML< 3 : 0 / 2
ML< 4 : 1 / 1 ML< 5 : 0 / 0 ML>= 5 : 0 / 0
seismic 2 / 5 (40.0 %) infra 0 / 1 (0.0 %)
6 ref phases: match, confus, miss -- 83.3 % 0.0 % 16.7 % spurious 28.6 %
match finished
```

In the above output, we see a lot of internal details because of the verbosity level. The key things to look for are the overall statistics of *2/5* ie. 40% of the LEB seismic events were matched while one spurious event was generated. In the following section we see exactly which LEB events were matched or not.

```
15342681 --> 15292276   []                    [match]
15336857 --> unmat    []  (1 assocs)
15341305 -->          []  (2 assocs)  [miss]
15314322 -->          []  (2 assocs)  [miss]
15342745 -->          []  (3 assocs)  [miss]
          --> 15292332  (2 assocs)  [spurious]
15314321 -->          []  (5 assocs)  [miss]
15342962 --> 15292333   []                    [match]
```

LEB origin ids 15342681 and 15342962 were matched while 15336857, 15341305, 15314322, and 15314321were not matched, We also see that 15314321 has 5 associated arrivals. Normally events with 2 or 3 associated arrivals are somewhat difficult to match, so those should not be surprising. In this case, the event with 5 associated arrivals is worth investigating. In the rest of this section we shall look deeper into this event.

C.2 Event Score Diagnostics

Once a missed event has been identified, the next step is to obtain detailed scoring diagnostics using the IDCNETVISA_DEBUG utility. This tool provides comprehensive information about how NET-VISA processed the event, including arrival-level details and scoring metrics.

C.2.1 Running the Score Diagnostic

To obtain detailed diagnostics for a specific LEB event:

Command

```
idcnetvisa_debug par=debug.par orid=15314321
```

Parameter file (debug.par):

```
in_db_account=$(ORACLE_CONN)
in_origin_table=leb.origin
in_assoc_table=leb.assoc
visa_origin_table=vsel3.origin
visa_assoc_table=vsel3.assoc
EARTHSPECS=/dvl/software/shi/config/earth_specs
netvisa_curr_model=/netvisa_models/2.5/curr/
netvisa_threads=12
netvisa_ttime_prefix="$(EARTHSPECS)/TT/iasp91/iasp91."
netvisa_ddrange_file=$(EARTHSPECS)/GA/GA_dist_depth_ranges
netvisa_qfvc_file=$(EARTHSPECS)/MAG/global/qfvc.mb
netvisa_sasc_dir=$(EARTHSPECS)/SASC
netvisa_radial2d_hydro_dir="$(EARTHSPECS)/TT/radial_2D/HYDRO_MONTHLY"
netvisa_radial2d_infra_dir="$(EARTHSPECS)/TT/radial_2D/INFRA_SEASONAL"
verbose=0
netvisa_use_libloc=1
netvisa_sssc_level=1
netvisa_use_qc=0
# libloc
vmodel_spec_file=$(EARTHSPECS)/TT/vmsf/idc.defs
sasc_dir_prefix=$(EARTHSPECS)/SASC/sasc
sssc_level=1
loc_verbose=0
# database tables
in_arrival_table=idcx.arrival
in_amplitude_table=idcx.amplitude
in_site_table=static.site
in_hydro_arr_group=idcx.hydro_arr_group
in_hydro_assoc=idcx.hydro_assoc
in_hydro_features=idcx.hydro_features
in_infra_features=idcx.pmcc_features
in_affiliation_table=static.affiliation
in_qcnetint_table=idcx.qcnetint
```

C.2.2 Understanding Score Diagnostic Output

The IDCNETVISA_DEBUG output provides multiple levels of diagnostic information:

Overall Arrival Statistics:

```
Starting 0.1 hrs from 1515778440 to 1515778620
i.e. 01/12/18 17:34:00 to 01/12/18 17:37:00
Reading detections from 1515756840 to 1515800220 in idcx.arrival
10108 arrivals; 54 guaranteed sites out of 257; avg. uptime 50.6%
10108 detections 2315 P 22 S 283 T 36 H 343 I 6014 tx+N
Loading model /netvisa_models/2.5/curr/
read_HY_info: NONE specified as infrasonic radial_2D file path.
Default travel times will be used.
```

This first section of the report provides information on the number and phase lables of the arrivals and the name of the model that was used to compute the event scores.

Event Summary Section:

```
lon 29.1 lat -26.1 depth 0.0 time 1515778552.6 mb 2.00 medium 0
orid 15314321 evscore -4.9
BOSA Pg:129105178 tres 1.43 azres 4.04 slores -4.09
BOSA Sn:129105181 tres -5.47 azres 21.97 slores -13.60
BOSA Lg:129105183 tres 0.66 azres 70.50 slores -20.72
I47ZA I:129105761 tres -120.19 azres -0.91 slores -68.69
I35NA I:129106975 tres -80.85 azres -6.87 slores -53.49
```

This section provides the basic information about the event in question such as its epicenter as well as the associated phases at various stations. For example, Pg, Sn, and Lg and station BOSA and I phases at a couple of infrasound stations I47ZA and I35NA. We also provide some information about the time residual (tres), azimuth residual (azres) and slowness residual (slores). These residuals provide the first bit of diagnostic information. The event score is **-4.9**. This is the reason that NET-VISA didn't build the event. It doesn't build events with negative scores. To understand why the scores are negative we see in the residuals that while Pg has small residuals the residuals of the Sn and Lg phases are very high. Normally a time residual of less than 2 seconds and azimuth residual of 10 seconds is considered small. The slowness residuals of -13.6 and -20.7 for these phases suggests that these are probably not Sn and Lg phases.

Detection Scores

```
Pg:BOSA arid 129105178 time 01/12/18 17:37:11.224 iphase Pn
tres 1.43 seaz 54.16 azres 4.04 delta 4.23 slow 14.14
sres -4.09 idist 0.8 itime 2.2 detscore 9.6
 Det Attribute Scores { detampsc: 0.7, detazsc: 1.8,
detphasesc: 0.9, detsc: -0.5, detslosc: 1.9,
detsnrsc: 1.1, dettimesc: 4.8, misdetsc: -1.0, }
Sn:BOSA arid 129105181 time 01/12/18 17:37:44.349 iphase Px
tres -5.47 seaz 54.16 azres 21.97 delta 4.23 slow 11.12
```

```
sres -13.60 idist 13.9 itime -140.1 detscore -2.6
 Det Attribute Scores { detampsc: 0.7, detazsc: 0.8,
detphasesc: -0.0, detsc: -1.3, detslosc: -0.2,
detsnrsc: -1.5, dettimesc: 4.3, misdetsc: -0.9, }
Lg:BOSA arid 129105183 time 01/12/18 17:38:07.799 iphase N
tres 0.66 seaz 54.16 azres 70.50 delta 4.23 slow 11.05
sres -20.72 idist 17.3 itime -128.2 detscore 2.9
 Det Attribute Scores { detampsc: 2.5, detazsc: 0.1,
detphasesc: 0.0, detsc: -0.2, detslosc: 0.5,
detsnrsc: 0.9, dettimesc: 4.4, misdetsc: -1.4, }
I:I47ZA arid 129105761 time 01/12/18 18:00:40.0 iphase N
tres -120.19 seaz 54.23 azres -0.91 delta 4.25 slow 308.23
sres -68.69 detscore None
I:I35NA arid 129106975 time 01/12/18 18:50:00.0 iphase N
tres -80.85 seaz 124.93 azres -6.87 delta 12.62 slow 305.19
sres -53.49 detscore None
```

The detection scores further breaks down the score at a per detections level. The Pg at BOSA has a positive score of 9.6 signifying that this provides good evidence for the event.

```
Pg:BOSA arid 129105178 time 01/12/18 17:37:11.224 iphase Pn
tres 1.43 seaz 54.16 azres 4.04 delta 4.23 slow 14.14
sres -4.09 idist 0.8 itime 2.2 detscore 9.6
```

The detection scores are further broken down into the attribute level scores. The amplitude, azimuth phase, slowness, SNR, and time scores can provide some more information. A positive value for this score can suggest that the value of that attribute agrees well with the expected value.

```
 Det Attribute Scores { detampsc: 0.7, detazsc: 1.8,
detphasesc: 0.9, detsc: -0.5, detslosc: 1.9,
detsnrsc: 1.1, dettimesc: 4.8, misdetsc: -1.0, }
```

For each detection, we attempt to invert the detection and measure the distance and time difference from the inverted event and the actual event. These two values are reported in idist and itime respectively. The idist and itime values here suggest that if this detection was inverted we would find an event with 0.8 degrees distance and 2.2 seconds of the LEB event. This is very good agreement and it also suggests that our birth move would have no problems proposing this event. In general idist within 5 degrees and itime within 10 seconds is considered good agreement. If there was other supporting evidence we would have built this event. The score breakdown suggests that the arrival time agrees really well with the event hence a dettimesc of 4.8. Azimuth also has good agreement (detazsc of 1.8).

Now, let's look at Sn at Lg at BOSA.

```
Sn:BOSA arid 129105181 time 01/12/18 17:37:44.349 iphase Px
tres -5.47 seaz 54.16 azres 21.97 delta 4.23 slow 11.12
```

```
sres -13.60 idist 13.9 itime -140.1 detscore -2.6
 Det Attribute Scores { detampsc: 0.7, detazsc: 0.8,
detphasesc: -0.0, detsc: -1.3, detslosc: -0.2,
detsnrsc: -1.5, dettimesc: 4.3, misdetsc: -0.9, }
Lg:BOSA arid 129105183 time 01/12/18 17:38:07.799 iphase N
tres 0.66 seaz 54.16 azres 70.50 delta 4.23 slow 11.05
sres -20.72 idist 17.3 itime -128.2 detscore 2.9
 Det Attribute Scores { detampsc: 2.5, detazsc: 0.1,
detphasesc: 0.0, detsc: -0.2, detslosc: 0.5,
detsnrsc: 0.9, dettimesc: 4.4, misdetsc: -1.4, }
```

Sn has a negative score of 2.6 while Lg has a small positive score of 2.8. So, this suggests that NET-VISA would not have associated the Sn phase while it could have associated the Lg phase.

Finally, the I phases:

```
I:I47ZA arid 129105761 time 01/12/18 18:00:40.0 iphase N
tres -120.19 seaz 54.23 azres -0.91 delta 4.25 slow 308.23
sres -68.69 detscore None
I:I35NA arid 129106975 time 01/12/18 18:50:00.0 iphase N
tres -80.85 seaz 124.93 azres -6.87 delta 12.62 slow 305.19
sres -53.49 detscore None
```

The I phases have a detscore of None which suggests that NET-VISA would not have built the detection at all. In this case we don't even have a score because I phases are not allowed to be associated to arrivals that were given an iphase value N as in this case. So, overall, we couldn't associate the I phases and the Sn and Lg didn't give enough supporting evidence for NET-VISA to form this event.

References

[Arora, 2012a] Arora, N.S., S. Russell and H.A. Kuzma. 2012. Test Report: NET-VISA Performance on Large Aftershock Sequences. CTBTO report submitted. Solicitation No. 2011-0321, NET-VISA Software Engineering and Operational Testing.

[Arora et al, 2012] Arora, N.S., Given, J., Tomuta, E., Russell, S., and Spiliopoulos, S., "Analyst Evaluation of NET-VISA (Network Processing Vertically Integrated Seismic Analysis) at the CTBTO." In Proc. Monitoring Research Review (MRR 2012), Albuquerque, New Mexico, 2012.

[Arora 2012b] Arora, N. 2012. Model-based Seismic Monitoring, Doctoral Dissertation, University of California, Berkeley, Dept. of Computer Science.

[Arora et al, 2011] Arora, N., Russell, S., Sudderth, E. 2011. NET-VISA: Vertically Integrated Seismological Analysis for Network Processing: Final Technical Report. CTBTO Technical Report.

[Arora et al, 2010(a)] Arora, N., S. Russell and E. Sudderth. 2010. Evaluation criteria for net-visa and sig-visa prototypes for detection, identification, and association of seismic data. CTBTO Technical report.

[Arora et al, 2010(b)] Arora, N., S. Russell, P. Kidwell and E. Sudderth. 2010. Global seismic monitoring as probabilistic inference. *In* Lafferty, J., Williams, C.K.I., Shawe-Taylor, J. and Zemel R.S. *eds*. **Advances in Neural Information Processing Systems, 23**: 73-81.

[Draelos et al 2012] Draelos, T.J., S. Ballard, C.J. Young, M. Gonzales and R. Brogan. 2012. A Seismic Event Detection and Signal Association Algorithm. Geophysical Research Abstracts. Vol 14, EGU2012-6831. EGU General Assem$_{\text{b}}$ly, 2012.

[Douglas 1967] Douglas, A. 1967. Joint epicentre determination. Nature, 215:47 (48).

[Gelman et al, 2004] Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin. 2004. **Bayesian Data Analysis**. Chapman & Hall.

[Gutenber and Richter, 1954] Gutenberg, B. and C.F. Richter. 1954. **Seismicity of the Earth and Associated Phenomena**. Princeton University Press, Princeton, N. J., 2nd Ed.

[Le Bras et al,1994(a)] Le Bras, R., H. Swanger, T. Sereno, G. Beall and R. Jenkins. 1994. Global association. Technical Report ADA304805, Science Applications International Corp, San Diego, CA, URL http://handle.dtic.mil/100.2/ADA304805.

[Le Bras et al, 1994(b)] Le Bras, R., H. Swanger, T. Sereno, G. Beall, R. Jenkins, and W. Nagy. 1994 Global association: Design document and user's manual. Technical Report SAIC-94/1142, Science Applications International Corp, San Diego, CA.

[Mackey et al, 2009] Mackey, L., A. Kleiner, and M.I. Jordan. 2009. Improved automated seismic event extraction using machine learning. Eos Trans. AGU, **90**(52), 2009. Fall Meet. Suppl., Abstract S31B-1741.

[Meyers et al, 2007] Myers, S.C., G. Johannesson and W.A. Hanley. 2007. Bayesian hierarchical method for multiple-event seismic location. Geophysical Journal International, **171**: 1049-1063.

[Milch and Russel 2010] Milch, B. and S. Russell. 2010. "Extending Bayesian Networks to the Open-Universe Case." In Rina Dechter, Hector Geffner, and Joseph Y Halpern, Eds., Heuristics, Probability and Causality. A Tribute to Judea Pearl. College Publications.

[Kennett and Engdahl,1991] Kennett, B. L. N. and E.R. Engdahl. 1991. Traveltimes for global earthquake location and phase identification. Geophysics Journal International, **105**:429 -465.

[Vaidya et al, 2009] Vaidya, S., E.R. Engdahl, R. Le Bras, K. Koch, and O. Dahlman. 2009. Strategic initiative in support of CTBT data processing: vDEC (virtual Data Exploitation Centre). In CTBTO International Scientific Studies.

CTBTO PREPARATORY COMMISSION | preparatory commission for the comprehensive nuclear-test-ban treaty organization

Manual

IDC Database Schema – ODB

12 January 2021

IDC-ENG-MAN-102

Distr.: INTERNAL USE ONLY

Summary

This document describes the Database Schema for Seismic, Hydroacoustic, Infrasound monitoring technologies (referred as S/H/I hereinafter) along with the Database Schema for Radionuclide (referred as RN hereinafter) in use as of January 12, 2021 at the International Data Centre (IDC) in Vienna, Austria. This document specifically outlines the structure and content of database ODB.

Disclaimer

The IDC database is undergoing active improvements and changes. The database schema provided here has been generated semi-automatically and reflects the information that was available on January 12, 2021. Subject matter experts are continuing to provide feedback and updates into the IDC database and where necessary, updates are being made. This document will be updated as required.

Contents

1. About this document....5
1.1. Purpose....5
1.2. Scope....5
1.3. Audience....5
1.4. Related Information....5
1.5. Using this Document....5
1.6. Conventions....7
1.7. Dates and Times....10
1.8. Oracle Data Types....10
2. S/H/I Entity Relationships....11
2.1. Overview....11
2.2. Table Categories....11
2.3. Fundamental Tables....11
2.4. Reference Tables....18
2.5. S/H/I Application Software Tables....21
3. S/H/I Objects Description....33
3.1. S/H/I Table Descriptions....33
3.2. S/H/I Column Descriptions....132
3.3. S/H/I Trigger Descriptions....335
3.4. S/H/I Index Descriptions....341
3.5. S/H/I Synonym Descriptions....348
3.6. S/H/I Foreign Key Descriptions....354
4. Radionuclide Entity Relationships....355
4.1. Overview....355
4.2. Radionuclide Tables....355
5. Radionuclide Objects Description....370
5.1. Radionuclide Table Descriptions....370
5.2. Radionuclide Column Descriptions....502
5.3. Radionuclide Trigger Descriptions....669
5.4. Radionuclide Index Descriptions....766
5.5. Radionuclide Synonym Descriptions....777
5.6. Radionuclide Foreign Key Descriptions....782
6. Functions, Procedures and Packages....788
6.1. Functions....788
6.2. Procedures....788
6.3. Packages....788
References....789
Glossary....791

List of Figures

Figure 1: Sample entity relationship....9
Figure 2: Overview of the fundamental SHI tables....12
Figure 3: Event table relationships....14
Figure 4: Detail tables related to the *ARRIVAL table*....15
Figure 5: Measurement table relationships....16
Figure 6: Event characterization table relationships....17
Figure 7: Waveform table relationships....18
Figure 8: Referential core table relationships....19
Figure 9: Network table relationships....20
Figure 10: Channel table relationships....21
Figure 11: Relationships of tables used in automatic processing....22
Figure 12: Event screening table relationships....23
Figure 13: Tables involved in Analyst Review of Time-series Data....24
Figure 14: Map table relationships....25
Figure 15: Tables used by distributed processing applications....26
Figure 16: Continuous Data Subsystem table relationships....27
Figure 17: Message Subsystem table relationships....28
Figure 18: Subscription Subsystem table relationships....29
Figure 19: Message and Subscription Subsystem support tables....29
Figure 20: Data archiving subsystem tables....30
Figure 21: Tables used by System Monitoring Applications....31
Figure 22: Tables used for Performance Monitoring....32
Figure 23: Tables used to support schema....32
Figure 24: Radionuclide equipment....355
Figure 25: Raw sensor data....356
Figure 26: Sample information and pulse height data....357
Figure 27: Gamma calibration pairs....358
Figure 28: Processing parameters and nuclear data for gamma spectra....359
Figure 29: Calculated *gamma* calibration data....360
Figure 30: Results and categorization data for particulate gamma spectra....361
Figure 31: Processing parameters and nuclear data for SPALAX gamma spectra....362
Figure 32: Results and categorization data for SPALAX gamma spectra....363
Figure 33: Beta-Gamma calibration pairs....364
Figure 34: Calculated beta-gamma calibration data....365
Figure 35: Processing parameters and nuclear data for SAUNA beta-gamma coincidence spectra....366
Figure 36: Results and categorization data of beta-gamma coincidence spectra....367
Figure 37: Analyst comment data....368
Figure 38: Multiple analyst review data....369

1. ABOUT THIS DOCUMENT

1.1. Purpose

This document describes the database schema in use as of 12 January 2021. This document presents the relationships between tables, table descriptions, and definitions of the table columns. This document does not describe the specific location and general use of these tables at the IDC or how to manipulate them to obtain information.

1.2. Scope

This document describes the S/H/I and RN schema used in the IDC databases. The schema includes relationships between tables, table descriptions, and definitions of the table columns.

This document does not describe the specific location and general use of these tables at the IDC or how to manipulate them to obtain information. In most cases it provides a recommendation for the format of each column in external file representations of the tables, however, utilities are available for exporting the table content in a variety of formats and users are free to select the most appropriate for their requirements.

This document can be generated towards several databases of the IDC, this one specifically outlines the structure and content of database ODB.

1.3. Audience

This document is intended for software engineers, scientists, processing engineers, and anyone who needs to interact with the S/H/I and RN databases at the IDC.

1.4. Related Information

This document supersedes all other documents describing the IDC S/H/I and RN Database Schema dated prior to 12 January 2021. Legacy versions of this document were labelled under the code IDC-5.1.1.

The following documents provide information and instructions for retrieving data from the IDC databases:

- Database Tutorial [IDC-5.1.2]
- Configuration of PIDC Databases [IDC-5.1.3.Rev1]

See References on page 789 for a listing of all the sources of information consulted in preparing this document.

1.5. Using this Document

This document is part of the core system documentation of the IDC. It is intended to serve as a User Guide. It provides information relevant to understanding IDC processing complementing the IDC Processing of Seismic, Hydroacoustic, and Infrasonic Data [IDC-OPS-MAN-001.Rev1] and IDC Processing of Radionuclide Data [IDC-OPS-MAN-050.Rev3] documents.

This document is organized as follows:

- **S/H/I Entity Relationship**

  This chapter provides an overview of the main S/H/I tables and describes the relationships between these tables.

- **S/H/I Objects Descriptions**

  This chapter describes each table in the S/H/I database schema (in alphabetical order). It includes information about the category to which the table belongs, the columns included in the table, Oracle storage types for each column, keys (primary, alternate, and foreign), and column brief descriptions.

  It also provides detailed descriptions of the columns of the S/H/I database schema including the tables in which the columns may be found, a full description of the column, Oracle storage type and external format, NA value, unit, and acceptable range of values.

- **Radionuclide Entity Relationships**

  This chapter provides an overview of the radionuclide database tables through an organizational description of the tables. This chapter describes the relationships between the radionuclide database tables.

- **Radionuclide Objects Descriptions**

  This chapter describes each table in the radionuclide database schema (in alphabetical order). It includes information about the columns included in the table, Oracle storage types for each column, and keys (primary, alternate, and foreign).

  It provides detailed descriptions of the columns of the radionuclide database schema including the tables in which the columns may be found, a full description of the column, Oracle storage type and external format, NA value, units, and acceptable range of values.

- **References**

  This section lists the sources cited in this document.

- **Glossary**

  This section defines the terms, abbreviations, and acronyms used in this document.

1.6. Conventions

This document uses a variety of conventions, which are described in the following tables. Table 1 shows the conventions for entity-relationship diagrams. Table 2 lists typographical conventions. Table 3 explains certain technical terms that are not part of the standard Glossary, which is located at the end of this document.

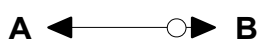
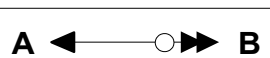
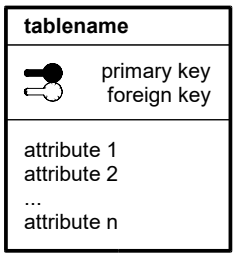
Description	Symbol
One A maps to one B	
One A maps to zero or one B	
One A maps to many Bs	
One A maps to zero or many Bs	
Database table	

Table 1: Entity-relationships symbols

The table name is always shown at the top of the table. Keys are shown below the table name. The primary key of a table is indicated with a filled symbol ⚷. Optional foreign keys shown by white symbol ⚲. Keys consisting of multiple columns are shown with a key symbol next to the first column of the key and the remaining columns are listed beneath with no symbols. All column names are shown below the key section in the diagrams.

Element[1]	Font	Example
• database table	UPPERCASE	DATAREADY DATAREADY.status
• database columns	lowercase	status
• processes, software units and libraries • user-defined arguments and variables used in parameter (par) files and program command lines • titles of documents	*italics*	*parsesubs* *delete-remarks object*

• computer code and output
• filenames, directories and web sites
• text that should be typed in *exactly* as shown
• SQL code
	`consolas`	`>(list 'a 'b 'c)`
		`ars.scm`
		`edit-filter-dialog`

Table 2: Typographical conventions

[1] Formatting may differ when the element is used as a section heading or it is hyperlinked.

Term	Description
field	database column
Keys: Foreign	primary key in a different table
Keys: Primary	set of database columns that uniquely define a row in a database table (unique key)

Table 3: Technical terms

Relationships between tables are usually indicated by naming the column or columns through which two tables are related. Table 4 explains the syntax used. In many cases the column names that must be compared in the two tables are not identical, or a column value in one table must be compared to more than one column value in another table. The delimiters in the syntax are the dash (-) and the slash (/). A dash (-) separates groups of column names from the two tables, and slashes (/) separate the columns that comprise the composite key. Other symbols, such as equal (=), vertical bar (|), ampersand (&), and brackets, specify how the columns are compared. Expressions within brackets in relationships between tables are evaluated first. The order of operations is: =, |, and &.

Syntax	Definition
col	This is the simplest case where the column names (col) of the keys in the tables at each end of the relationship are the same. Both keys consist of a single column.
col1/col2	A slash (/) is used when a key is comprised of multiple columns. Here, the keys in both tables are the same and consist of two columns, col1 and col2.
col1-col2	A dash (-) is used when the column names of the keys in the two tables are not the same. Col1 is the name of the key column in one table. Col2 is the name of the key column in the other table. Each key consists of a single column. Only one dash may be used, and the dash separates the keys of the two tables. A dash can be combined with slash (/) to show that the keys consist of multiple columns and that one or more of the columns have different names in the two tables as in col1/col2-col3/col4 (both parts of the key are different in the two tables), or col1/col2-col1/col3 (only the second part of the key is different in the two tables).
col1-col2/col3=value	An equals sign (=) is used when a component of a key must be set to a particular value. Here col1 is the name of the key column in one table. Col2 and col3 are the names of the key columns in the other table, and col3 must be set to the shown value. See the arid-tagid/tagname=arid relationship between ARRIVAL and WFTAG and orid-tagid/tagname=orid relationship between ORIGIN and WFTAG.
col1-col2 | col3	A vertical line (|) is used to show that the key in a table could be one of a set of columns. This is different from a key that consists of multiple columns. Here col1 corresponds to either col2 or col3. See the magid-mbid|msid|mlid relationship between NETMAG and ORIGIN.

Syntax	Definition
col1-col2 & col3	An ampersand (&) is used to show that a key in one table must have a value between the values of two keys in another table. Here the value of col1 must be between the values of col2 and col3. See the sta/chan/time-sta/chan/time&endtime relationship between WFDISC and SENSOR.
col1-col2 & col3	An ampersand (&) is used to show that a key in one table must have a value between the values of two keys in another table. Here the value of col1 must be between the values of col2 and col3. See the sta/chan/time-sta/chan/time&endtime relationship between WFDISC and SENSOR.
(col1)-(col2)	Brackets () are used to show that the keys within them have different formats and a conversion must be made to make the comparison. Here col1 corresponds to col2, but col1 and col2 have different storage formats (usually an epoch time versus a date). See the sta/chan/(time)-sta/chan/(ondate&offdate) relationship between SITECHAN and SITEAUX.

Table 4: Syntax Used to Indicate Database Table Relationships

The relationship shown in Figure 1, with TABLE1 columns on the left and TABLE2 columns on the right, demonstrates the possible relationships between the columns of two different tables. The syntax of Table 4 is used to interpret the relationships between the columns of TABLE1 and TABLE2 in Figure 1. col8 in TABLE2 has no matching column in TABLE1 and must be equal to value in this relationship. All other columns have one or more corresponding columns in the other table. Following the syntax, col1 in TABLE1 must have the same value as col1 in TABLE2, col2 in TABLE1 must have a value between col4 and col5 in TABLE2, and col3 in TABLE1 must match either col6 or col7 in TABLE2 for the one-to-many relationship indicated by the entity-relationship symbol (Table 1) to be true.

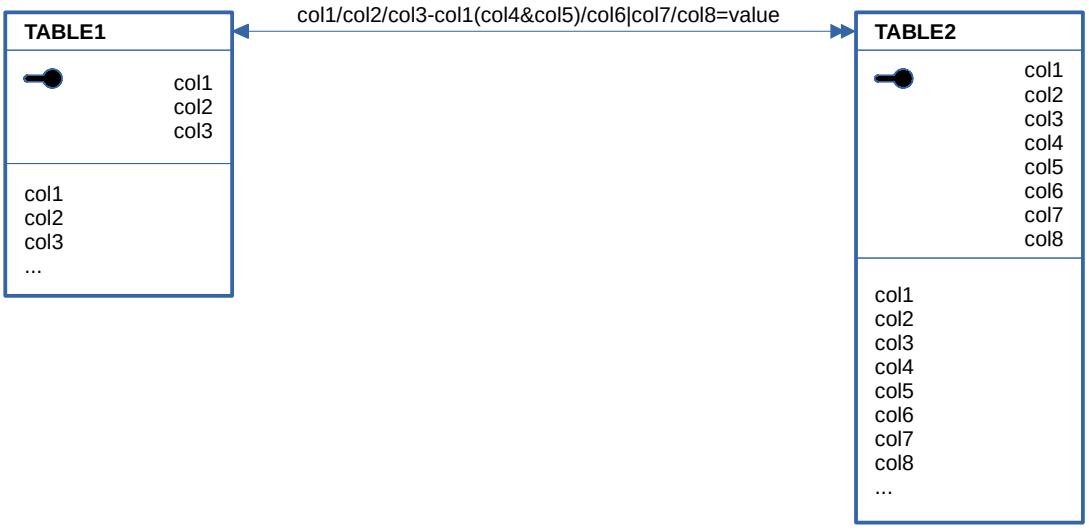


Figure 1: Sample entity relationship

1.7. Dates and Times

The time column used throughout the S/H/I schema is stored as epochal time, the number of seconds since January 1, 1970. Epochal time conventionally has a precision of one second, but, in the context of the IDC Database Schema, this is extended to a precision of one millisecond. Often time is matched by the more readable field, jdate. This "Julian date" represents a day in the form yyyyddd; for example, 1981231 where 1981 is the year (yyyy) and 231 is the day of year (ddd); this is not to be confused with the more conventional use of Julian dates in Astronomy, for example.

1.8. Oracle Data Types

The IDC database uses five of the available Oracle data types:

- **char**

  This fixed data format is only used for a very limited number of character-based columns in the S/H/I schema.

- **varchar2(n)**

  Almost all character data in the database are defined to be varchar2(n) where n is the maximum number of characters in the string. varchar2 does not store trailing blanks.

- **number(n)**

  All integer fields in the database are defined to be number(n) where n is the maximum number of digits allowed in the field. Number may also be used without specifying the maximum number of digits.

- **float(n)**

  Oracle supports the float(n) data type where n is the maximum number of binary digits. Float allows the approximation of single and double precision floats commonly used in scientific programming. The decimal point may be specified anywhere from the first to the last digit (or not at all). Most real numbers in the database are single precision float(24), except for fields such as time, endtime, and other time fields that are double precision float(53).

- **date**

  The only columns in the database that are declared to be the Oracle date data type are the lddate, moddate, last_mig_date, offdate, ondate, and initialdate columns, which store the day and time a record was inserted into the database or last modified.

2. S/H/I ENTITY RELATIONSHIPS

2.1. Overview

This chapter contains entity-relationship diagrams (ERDs) that show the relationships between database tables for the IDC S/H/I database schema following conventions described in section 1.6.

2.2. Table Categories

The database tables of the schema are grouped into natural categories. The first category contains the fundamental tables used with S/H/I data. These tables are used by many applications and are frequently updated. The fundamental tables contain a smaller set called the core tables. The core tables, which have remained essentially unchanged for many years, have been widely adopted by other seismological organizations, and therefore their definitions are virtually immutable. The relationships between the fundamental tables are shown in Figures 2 through 7.

Figures 8, 9 and 10 show the relationships between tables in the second category, referential tables for S/H/I data. These tables are read by many applications, but seldom if ever are the contents altered – they are considered to be "static". Most of the data in these tables pertain to International Monitoring System (IMS) stations and seismic networks.

Relationships between tables used by only a few application programs are shown in additional figures. These are further grouped according to the software configuration items of the software architecture. Figures 11 and 12 have tables associated with automatic processing; Figures 13 and 14 have tables associated with interactive processing; and Figure 15 has tables associated with management of distributed processes. Figures 18 through 20 are related to the data services software; Figures 21 and 22 are used by system monitoring software; Figure 23 shows tables that facilitate use of the database.

2.3. Fundamental Tables

The fundamental tables have been part of seismic processing at the IDC from its inception. They hold data characterizing time-series signals as well as the collections of signals that are associated with a particular origin or event.

Figure 2 summarizes the fundamental tables. In this figure, only the table names and keys are shown. Those tables that are part of the core set are labeled 3.0 core and 3.1 core, indicating previous versions of the CSS schema (see Anderson 1990, Swanger 1991, Swanger 1993). Figure 3 shows the tables that have data related to signal arrivals and the association of arrivals to origins and events. Additional data about signal arrivals are recorded in the tables shown in Figure 4, and Figure 6 shows tables holding data about source size.

Tables holding data used for event characterization are shown in Figure 7. Finally, Figure 8 shows the relationships of tables that have information about the time-series data, which are themselves saved in conventional files.

2.3.1. Summary of Tables and Keys

Fundamental tables related to stored time series, detected signals, and relationships between signals and events are shown in Figure 2. This summary provides the table names, keys, and relationships, but each table is presented in its entirety in a subsequent figure.

Data from the IMS stations (WFDISC) are processed for detections (DETECTION) from which arrivals and signal characteristics are extracted (AMPLITUDE, ARRIVAL, STASSOC). The signals are used to infer event locations (ASSOC, EVENT, NETMAG, ORIGERR, ORIGIN, PARRIVAL, STAMAG).

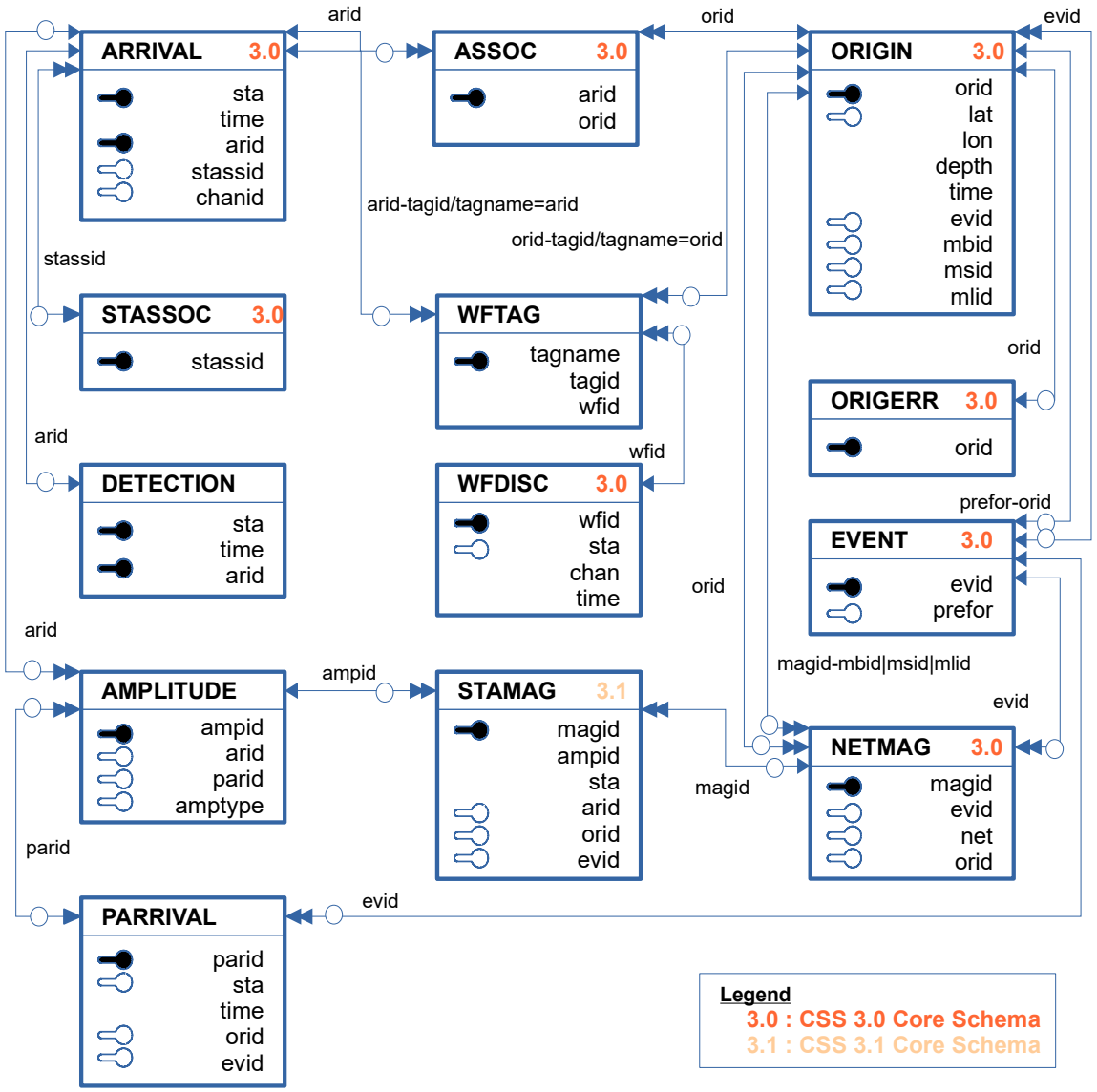


Figure 2: Overview of the fundamental SHI tables

2.3.2. Events

Figure 2 shows tables involved in forming origin hypotheses and events. Each table is part of the core set. Tables with information about event arrivals, location, origin time, and error are

included. During automated processing, groups of arrivals from several stations are associated with presumed events. The initial origin estimate for an individual station is contained in the STASSOC table, which is related to the ARRIVAL table through stassid. Specific arrivals listed in the ARRIVAL table are associated with origins in the ORIGIN table through the ASSOC table. An event may have several different origins, each of which is a different estimate. The preferred origin is indicated in the EVENT table.

2.3.3. Measurements

The tables shown in Figure 3, 4, and 5 contain additional data about seismic and hydroacoustic signals and the magnitudes, locations, and characteristics of events. Most tables in this category lie outside the core set and are used by a smaller number of applications.Figure 4 shows tables that have detailed data about arrivals. Figure 5 shows tables related to the location, origin time, and size of an event.

The event characterization tables in Figure 6 include SPVAR, COMPLEXITY, SPLP, THIRDMOM, TIMEFREQ, and CEPPKS. The SPVAR table contains arrival-based information and is related to the ARRIVAL table through arid. The other tables contain origin-based measurements and are related to ORIGIN through orid.

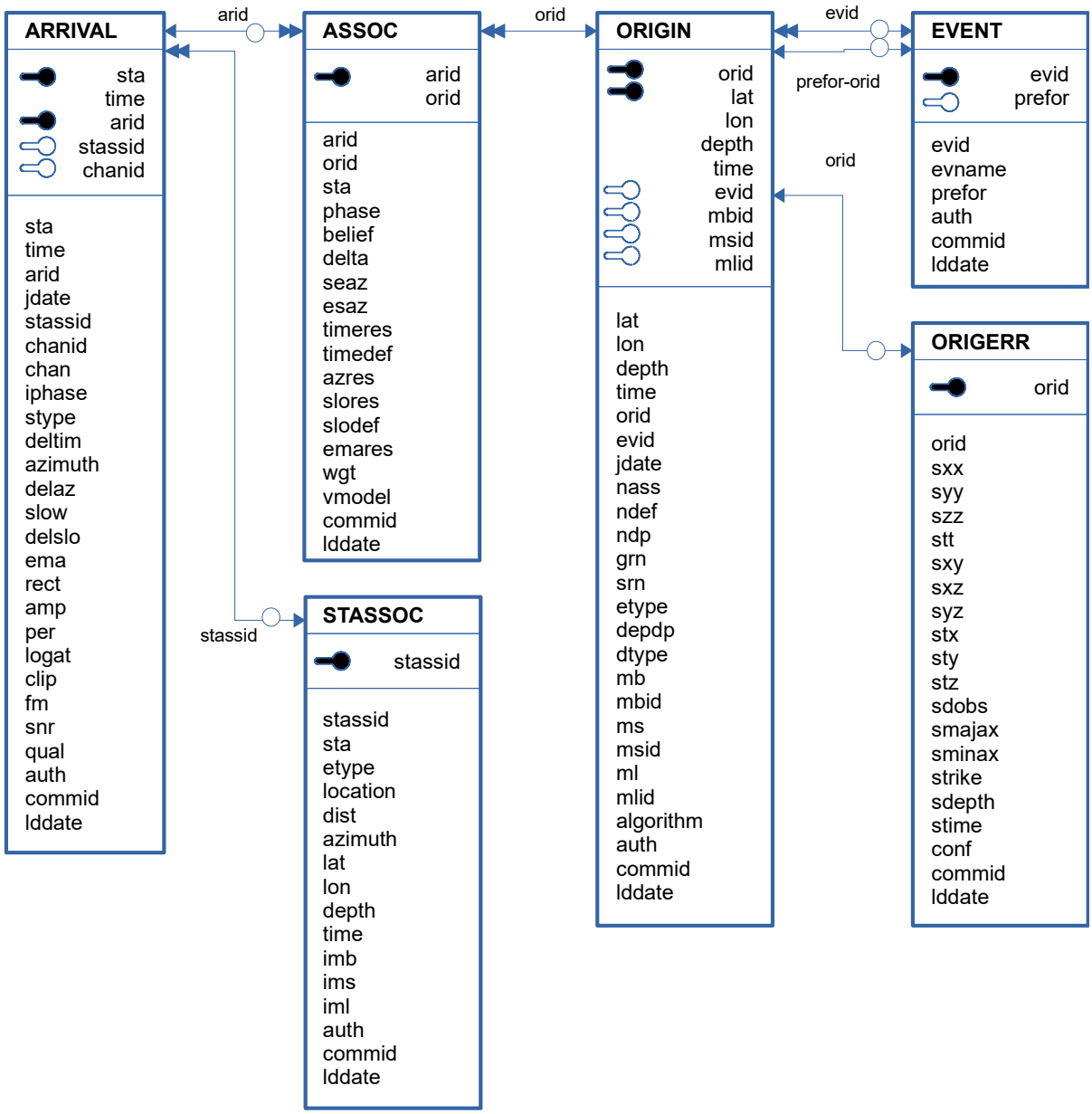


Figure 3: Event table relationships

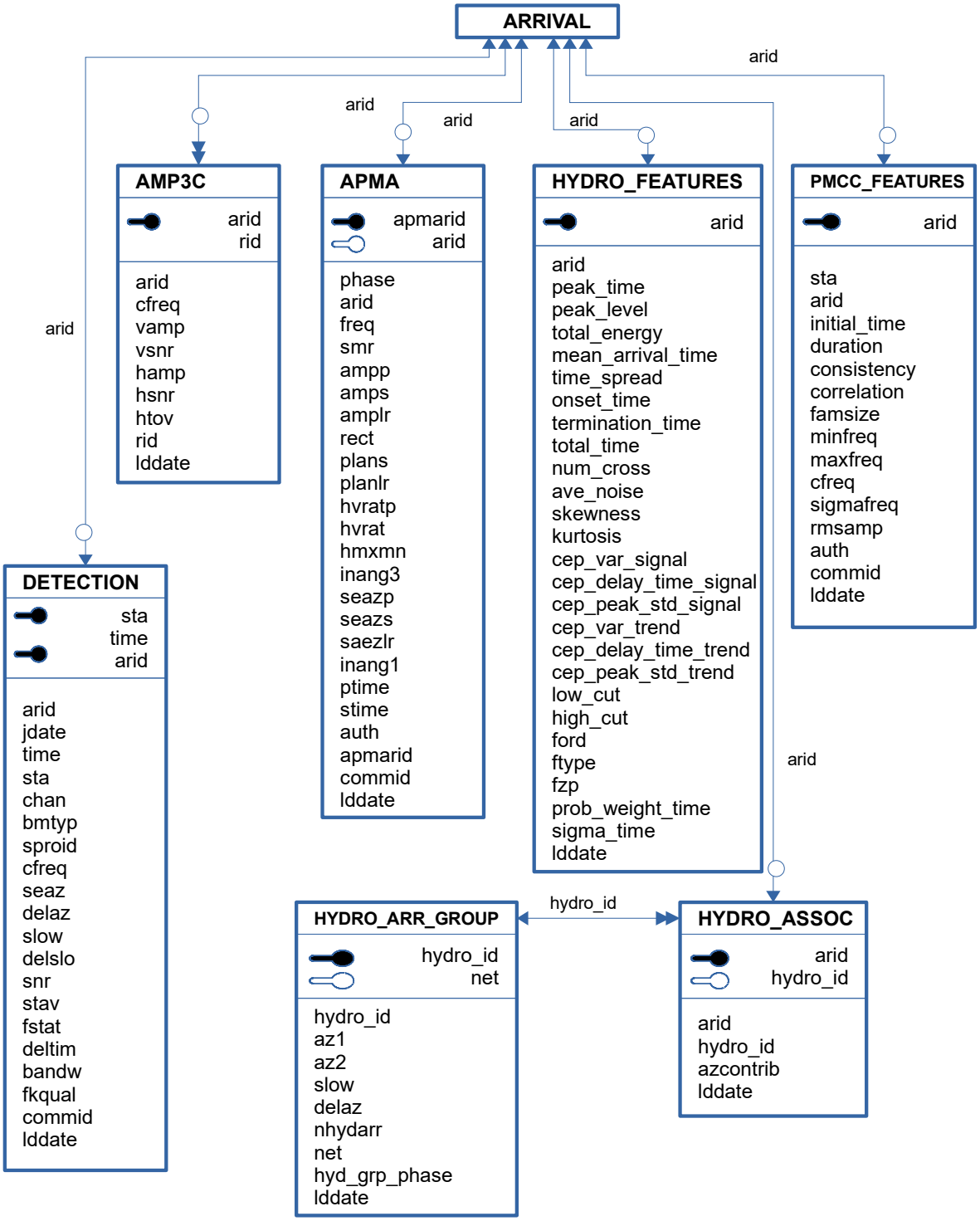


Figure 4: Detail tables related to the ARRIVAL table

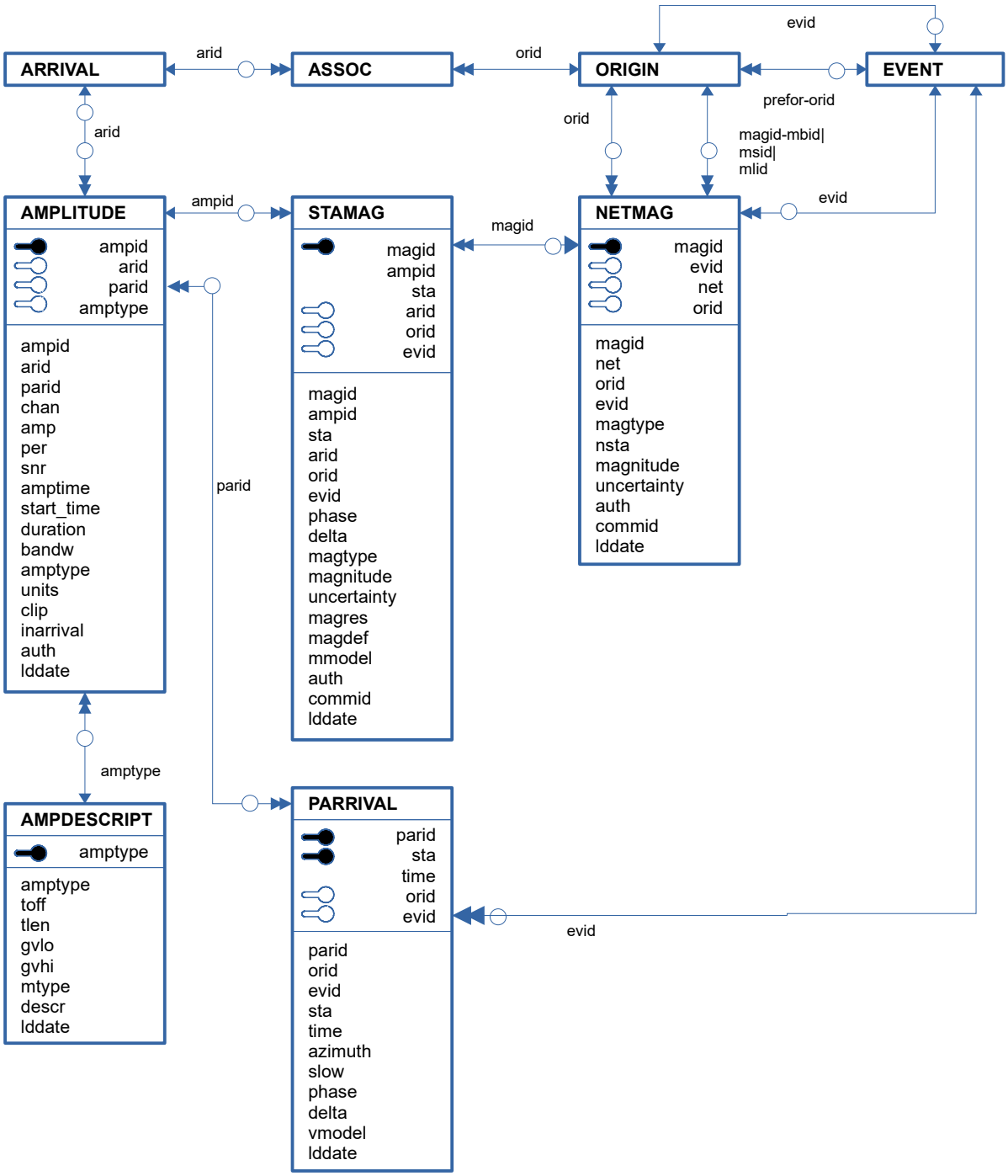


Figure 5: Measurement table relationships

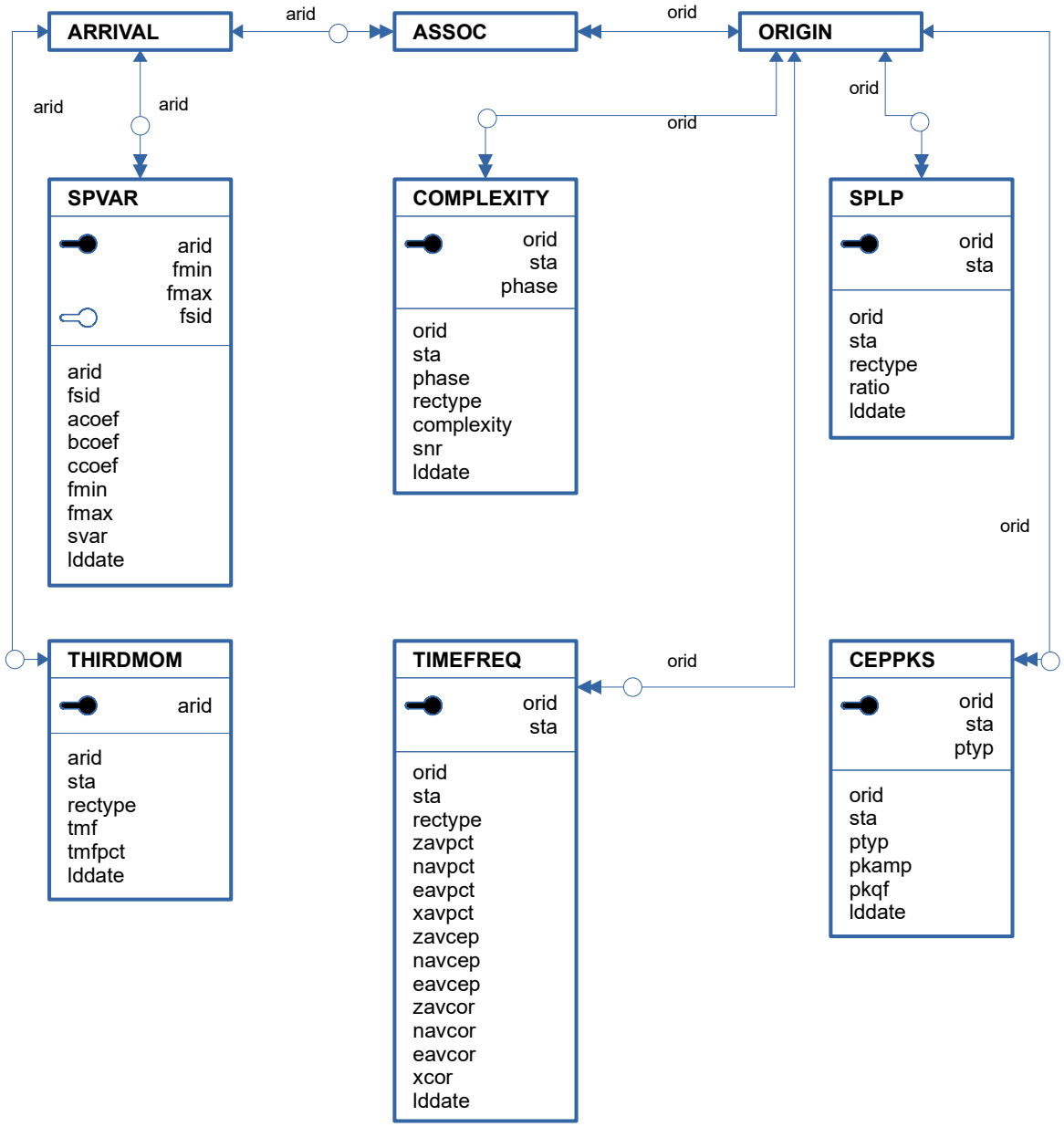


Figure 6: Event characterization table relationships

2.3.4. Waveforms

The waveform tables include WFDISC and WFTAG (see Figure 7). The WFDISC table is related to most other tables through the WFTAG table. However, sta, chan, and time are more frequently used to link the ARRIVAL and WFDISC tables directly.

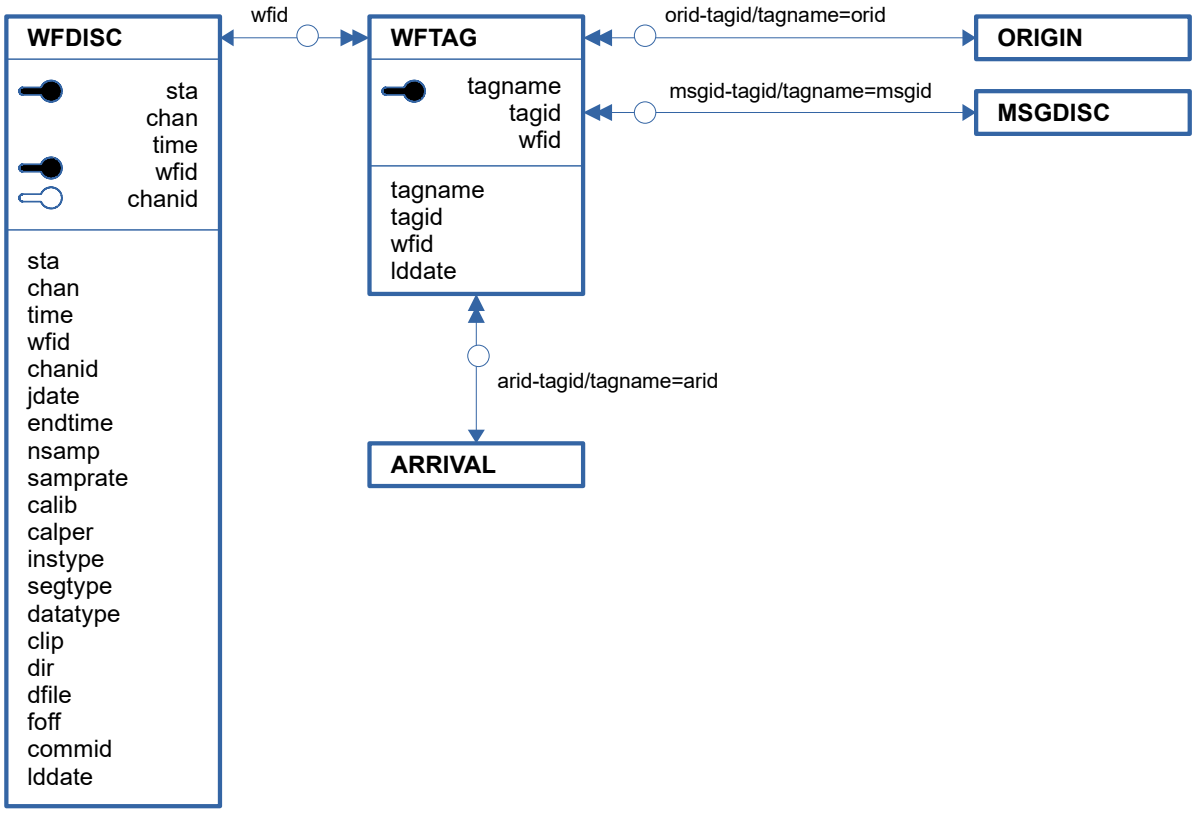


Figure 7: Waveform table relationships

2.4. Reference Tables

The reference tables are relatively static and primarily contain look-up information. Figure 8 is an overview of the tables in this category. As with Figure 2, no column names are provided, and tables in the core set are noted. Tables shown by name only are core tables that were previously shown in Figure 2.

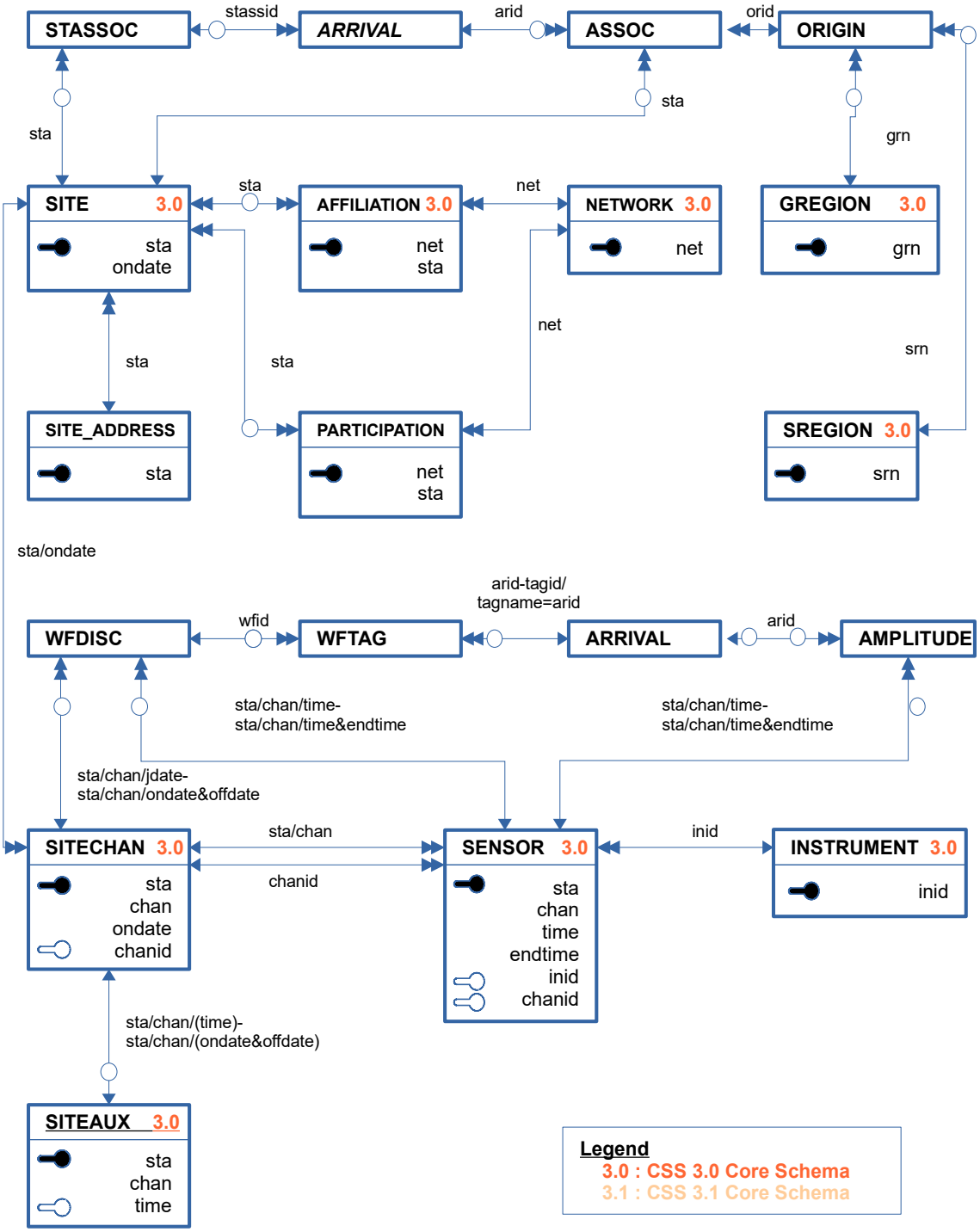


Figure 8: Referential core table relationships

2.4.1. Network

Figure 9 shows tables related to networks, which are collections of stations across wide geographic areas. Network tables include information on the names of the networks, the stations included in them, the participation of the stations in the network, and geographic and seismic region information.

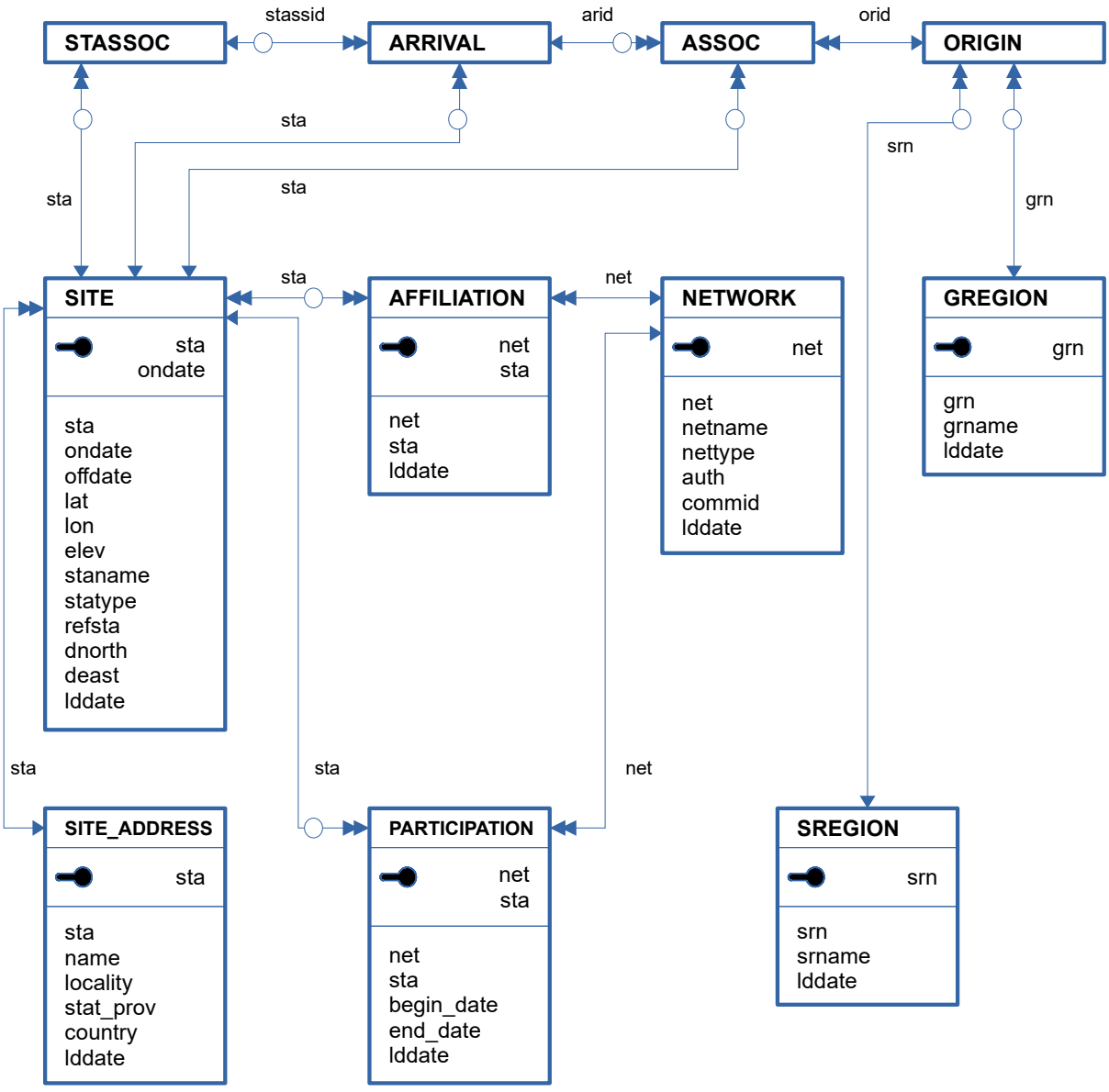


Figure 9: Network table relationships

2.4.2. Channel

Figure 10 shows tables that contain specific information about the data channels of the IMS stations. Instrument response information is identified in the INSTRUMENT table. The

SENSOR table is linked to it through inid and from there to WFDISC and AMPLITUDE through sta/chan/time. General data characteristics for any data channel are in SITEAUX.

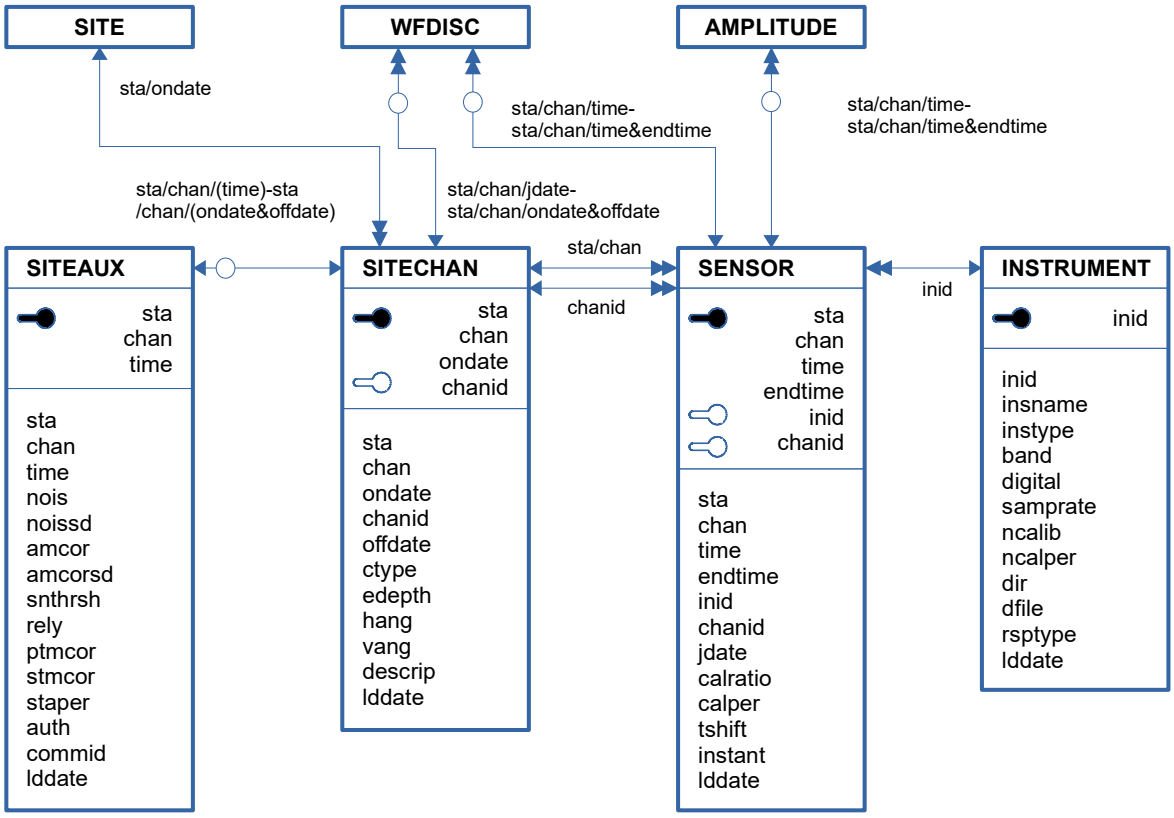


Figure 10: Channel table relationships

2.5. S/H/I Application Software Tables

The tables shown in this section are used by fewer applications than those previously described. The tables are organized by computer software configuration items (CSCIs).

2.5.1. Automatic Processing

The EVENT_CONTROL table is used by the event location and magnitude programs to preserve the values of key parameters that analysts set while reviewing the bulletin (see Figure 11), and it is also used in post-analysis processing. The SEISGRID and SEISINDEX hold historical seismicity data. The tables are also used by event quality control software.

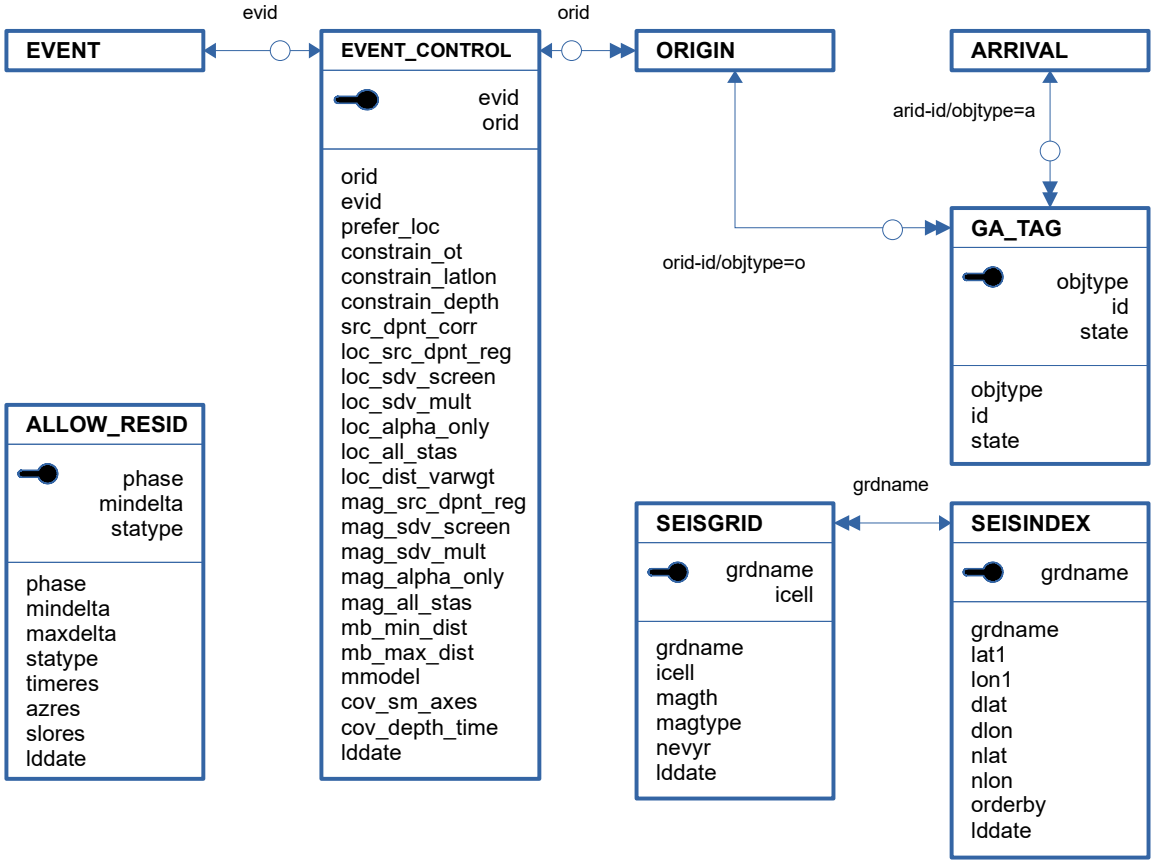


Figure 11: Relationships of tables used in automatic processing

Event Screening Subsystem uses the tables shown in Figure 12.

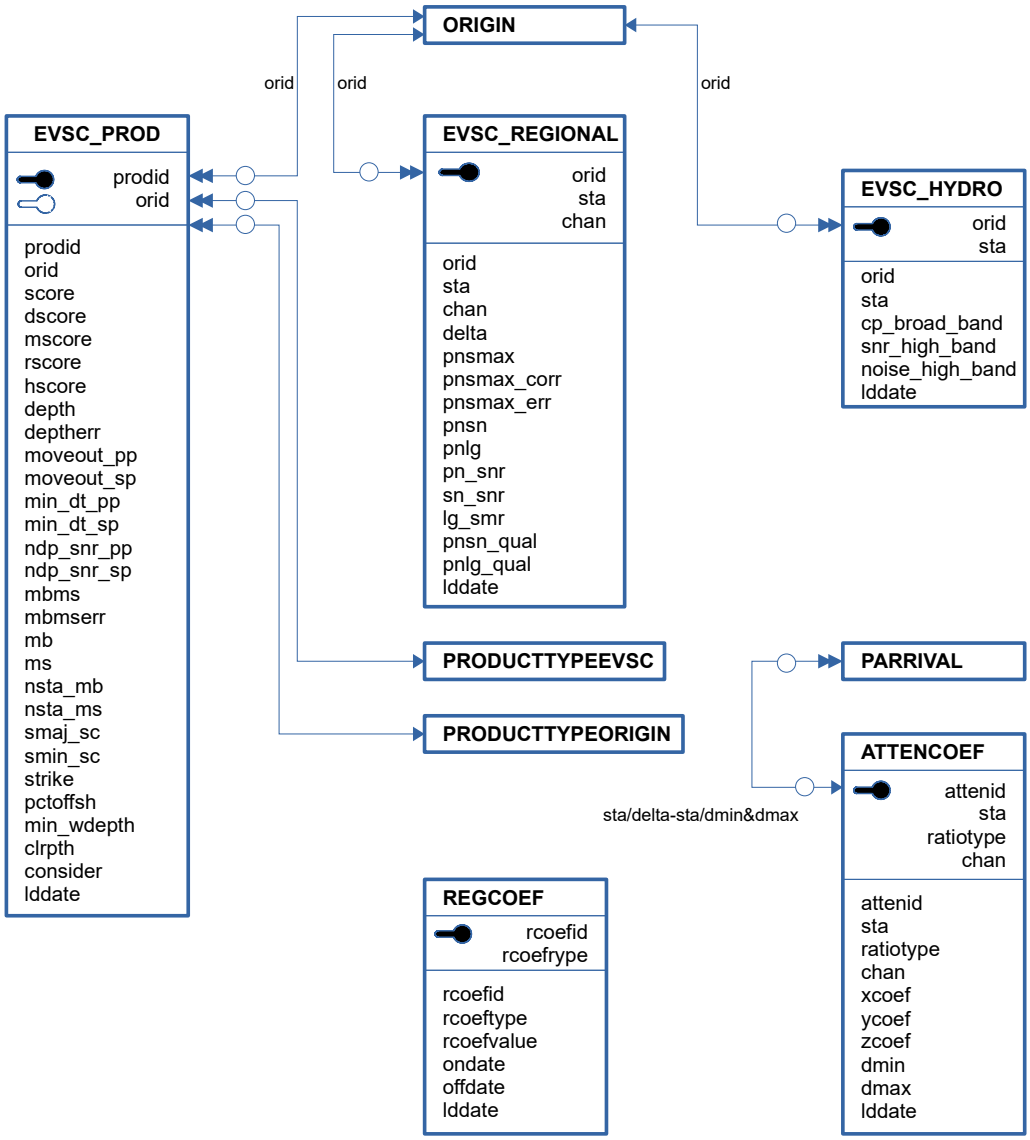


Figure 12: Event screening table relationships

2.5.2. Interactive Processing

The tables in this group support the work of S/H/I data analysts, but they are not linked to any of the core tables. Figure 13 shows tables involved in scheduling the work of analysts and recording progress. Figure 14 shows tables for the *Map* application. The tables have information about the base maps, as well as overlays and colours. Interactive analysis frequently includes calculation of frequency spectra and frequency-wave number spectra.

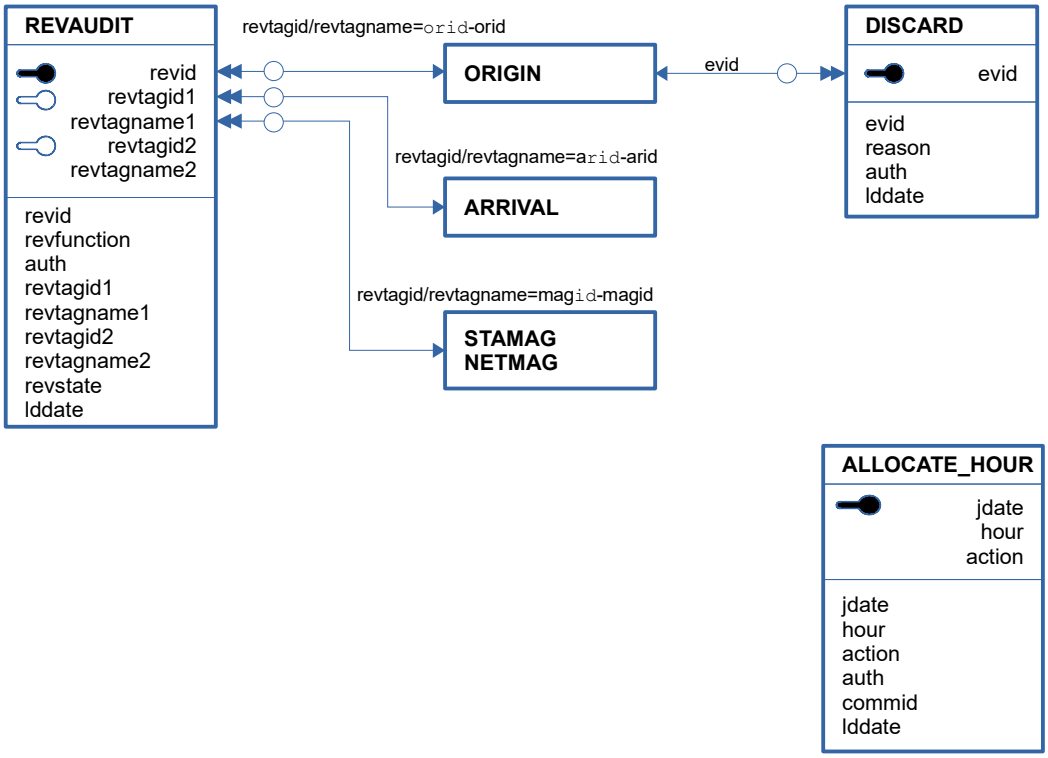


Figure 13: Tables involved in Analyst Review of Time-series Data

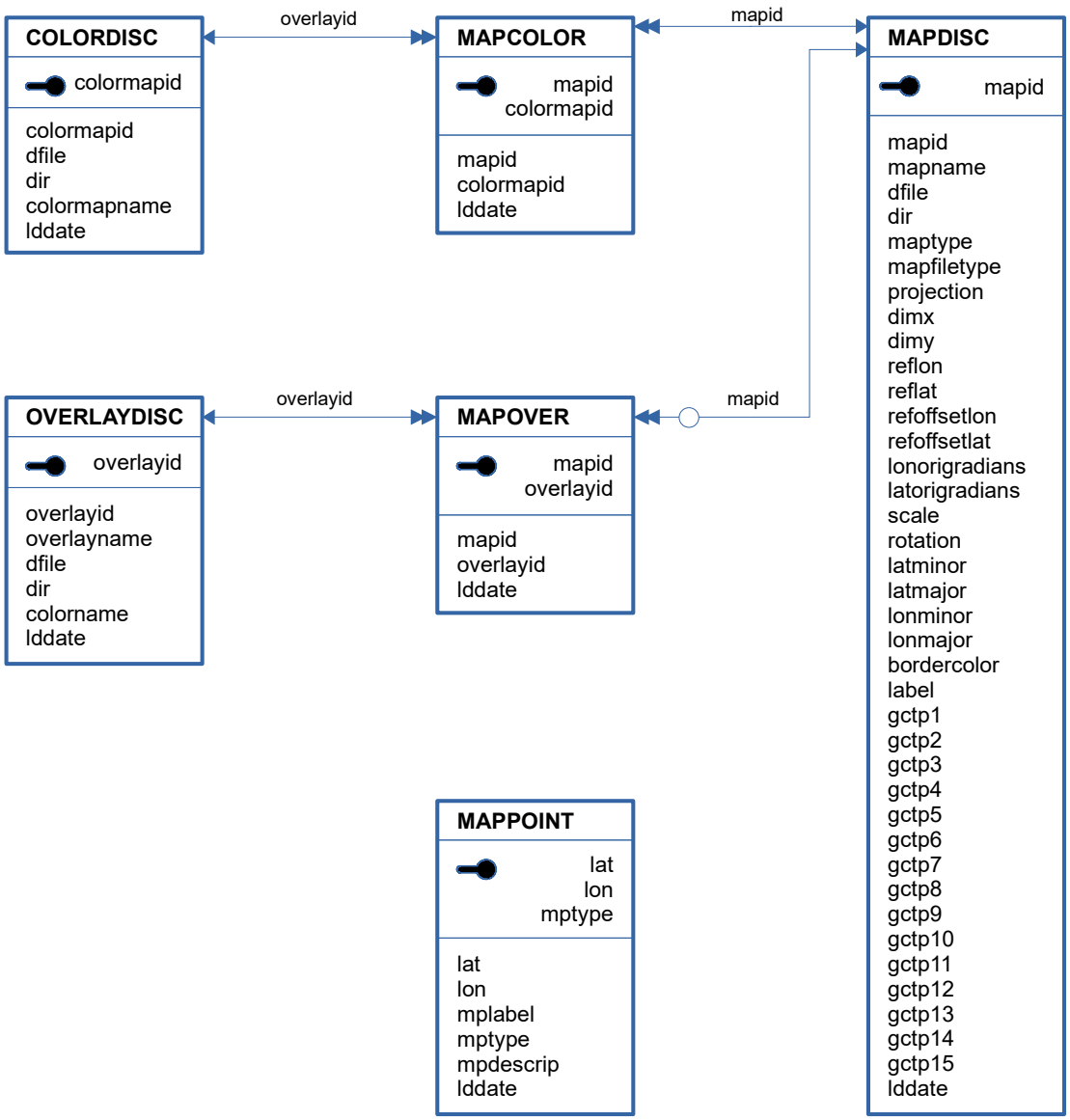


Figure 14: Map table relationships

2.5.3. Distributed Processing

The Distributed Processing ("DACS") CSCI has software that manages jobs and orchestrates the workflow [IDC-SPC-ENG-401.Rev0]. The INTERVAL and TIMESTAMP tables (shown in Figure 15) are used by this software as well as by some of the data services applications.

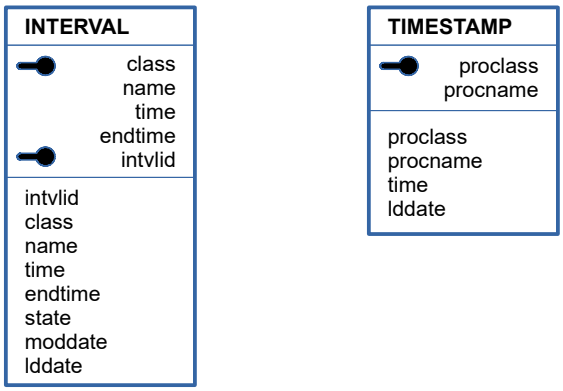


Figure 15: Tables used by distributed processing applications

2.5.4. Data Services

The Data Services CSCI has software to receive and forward time-series data received in Continuous Data (CD) Protocol (see CD Tools User Guide, IDC-ENG-SPC-100.Rev.1, IDC-ENG-SPC-101.Rev.3). CD Tools is a software system that contains a number of components (or tools), which can interact with continuous seismoacoustic data from IMS stations (or from any entity forwarding data in CD-1.0 or CD-1.1 format). Tools are available for receiving, storing, parsing, sending, archiving, and reporting.

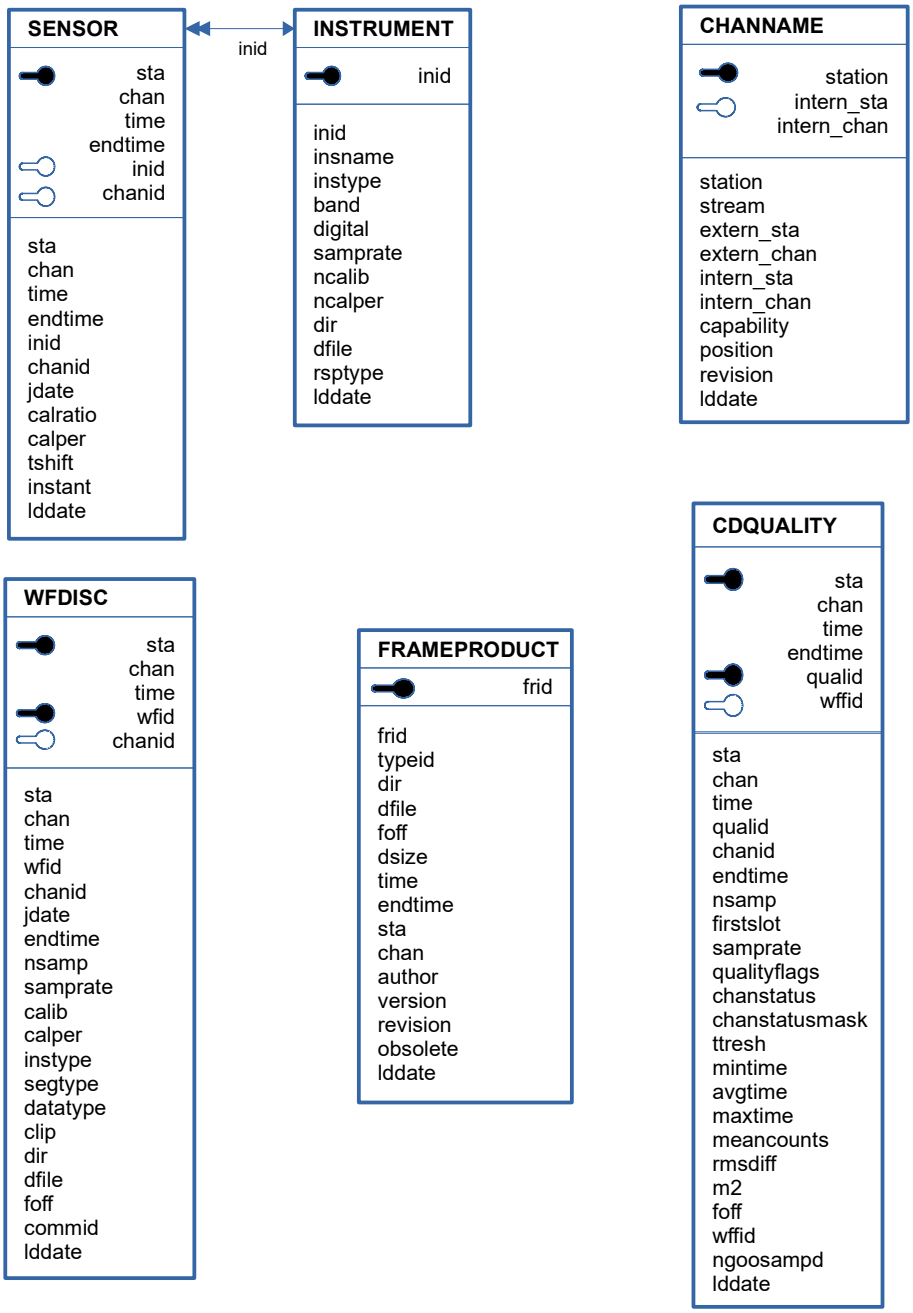


Figure 16: Continuous Data Subsystem table relationships

Figure 17 shows relationships between tables of the Message Subsystem. These tables are used by the Verification Data Messaging System (*VDMS*) and applications requesting seismic data from stations in the auxiliary network.

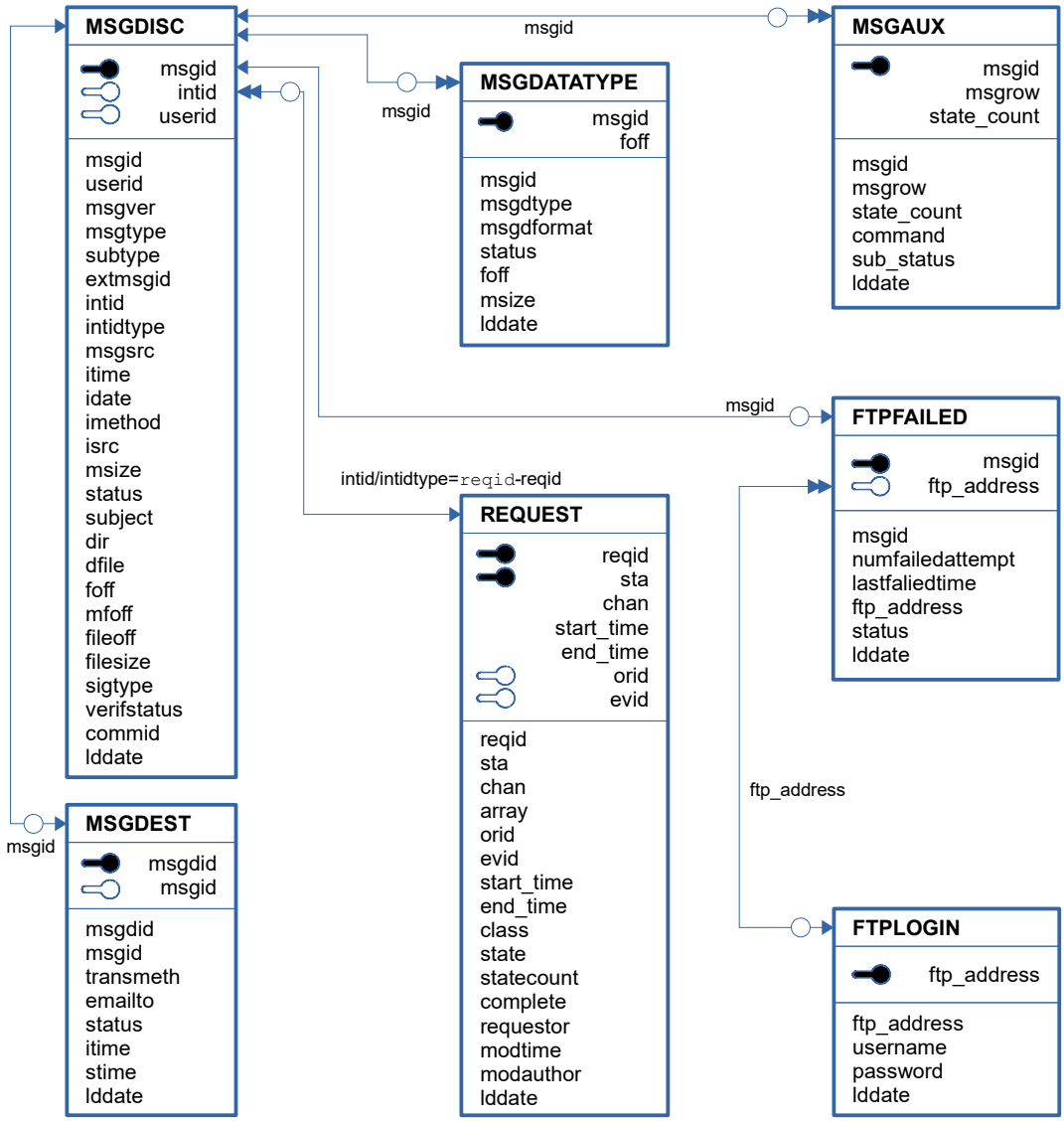


Figure 17: Message Subsystem table relationships

Figure 18 shows tables used by the Subscription Subsystem, which notes when products are ready, maintains subscriptions, and tracks subscription processing.

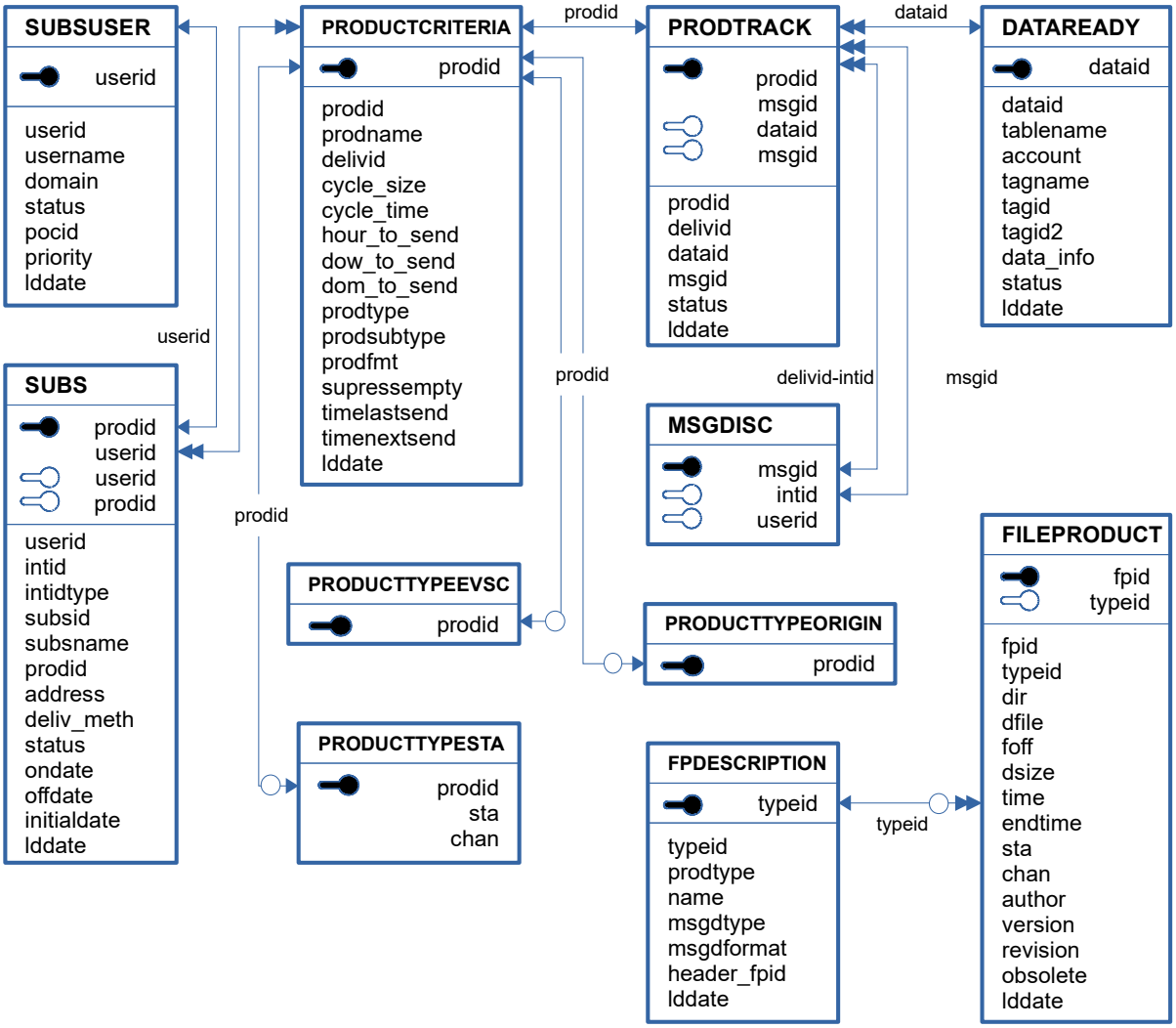


Figure 18: Subscription Subsystem table relationships

The DATAUSER table supports both the Message and Subscription Subsystems.

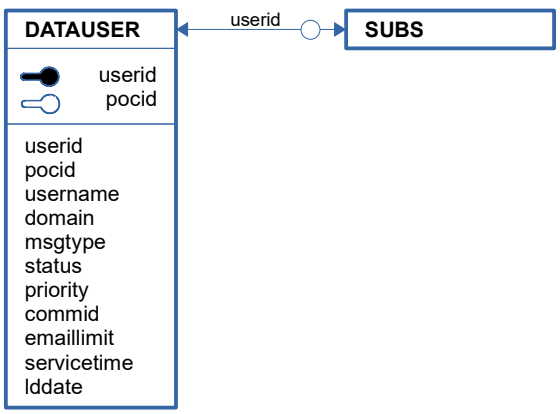


Figure 19: Message and Subscription Subsystem support tables

The tables of the data archiving subsystem contain information used by the software subsystems that migrate database tables between databases and that migrate time-series data to the mass-storage device (see Figure 20).

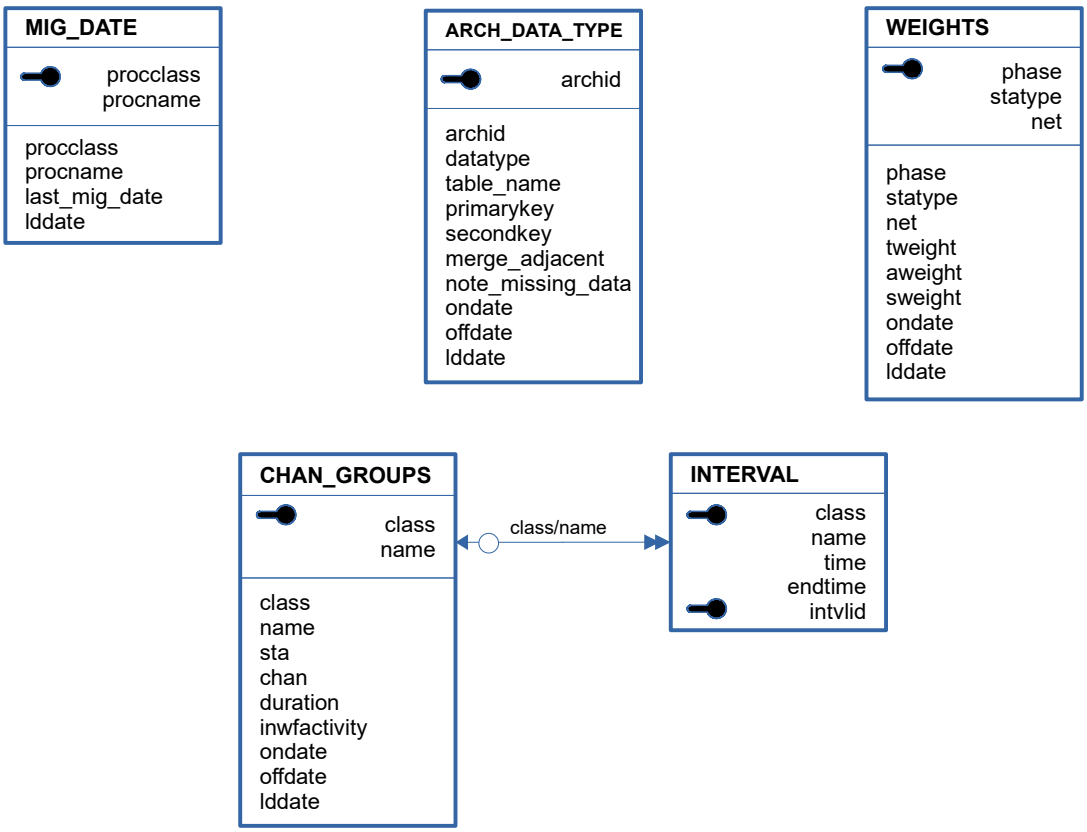


Figure 20: Data archiving subsystem tables

2.5.5. System and Performance Monitoring

The tables of the System Monitoring CSCI track the state of the hardware and software as well as the quality of the scientific results. *Xlogger* is used to record problems discovered in the system. The SITEPOLL table is a list of auxiliary seismic stations and channels that are regularly "polled" to determine their availability. System and performance monitoring applications tables are displayed in Figure 21.

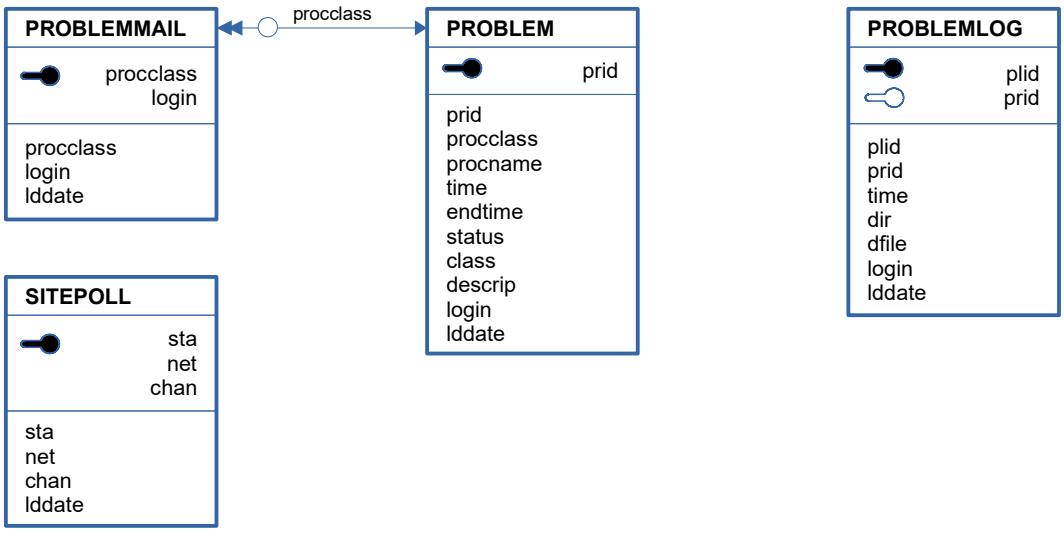


Figure 21: Tables used by System Monitoring Applications

Figure 22 displays tables that hold data pertinent to monitoring scientific performance. The BULL_COMP table (used by the *BullComp* application) compares event origins first estimated by automatic analysis against analyst-reviewed origins. In the figure, the ORIGIN table on the left exists in the database account for automatic processing, and the ORIGIN table on the right exists in the account for interactive processing. The tables at the bottom of the figure are used by other evaluation applications.

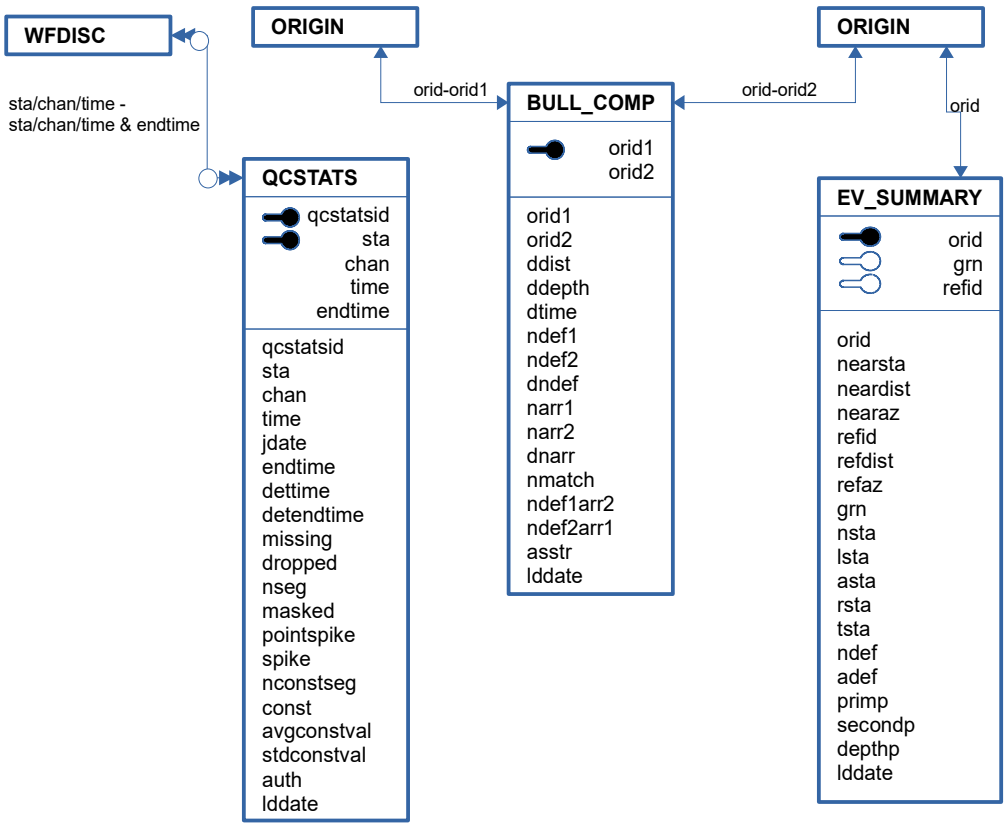


Figure 22: Tables used for Performance Monitoring

2.5.6. Database Support

Several tables facilitate use of the database (see Figure 23). The LASTID table is used by all applications that require unique values for surrogate keys. NA_VALUE specifies "not-available" values for many columns of the schema.

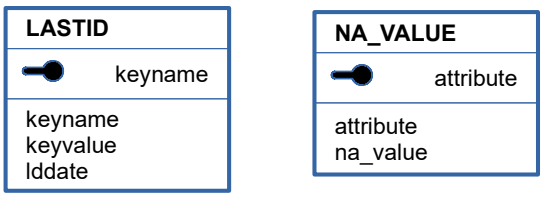


Figure 23: Tables used to support schema

3. S/H/I OBJECTS DESCRIPTION

3.1. S/H/I Table Descriptions

3.1.1. AFFILIATION

Owner(s)	STATIC
Category	Core, Reference
Description	The AFFILIATION table groups stations into networks.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	net	*varchar2(8)*	Network identification string	N
PI	sta	*varchar2(6)*	Station code	N
	lddate	*date*	Load date	Y

Primary key	PK_AFFILIATION(net, sta)
Indexes	AFFNETXX(net), AFFSTAX(sta, net), PK_AFFILIATION(net, sta)

3.1.2. AFFILIATION_EXCL

Owner(s)	STATIC
Description	Obsolete.

Pk/Fk/I	Column	Storage type	Description	Nullable
	net	*varchar2(8)*	Network identification string	N
	sta	*varchar2(6)*	Station code	N
	lddate	*date*	Load date	Y

3.1.3. AFFILIATION_LP (VIEW)

Owner(s)	STATIC
Category	Reference
Description	The AFFILIATION_LP view groups stations into networks. It is used for station-specific configuration for surface wave processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
	net	*varchar2(8)*	Network identification string	Y
	sta	*varchar2(6)*	Station code	Y
	lddate	*date*	Load date	Y

SQL Body	`SELECT NET, STA, LDDATE FROM STATIC.AFFILIATION WHERE NET NOT IN (SELECT NET FROM STATIC.AFFILIATION_SUB WHERE OFFDATE=-1) UNION SELECT NET, STA, LDDATE FROM STATIC.AFFILIATION_SUB WHERE OFFDATE=-1`

3.1.4. AFFILIATION_SUB

Owner(s)	STATIC
Category	Reference
Description	The AFFILIATION_SUB table groups stations into networks. It is used for station-specific configuration for surface wave processing.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	net	*varchar2(8)*	Network identification string	N
PI	sta	*varchar2(6)*	Station code	N
	lddate	*date*	Load date	Y
	ondate	*number(8,0)*	First julian date this row is valid	N
PI	offdate	*number(8,0)*	Last Julian date this row is valid	N

Primary key	PK_AFFILIATION_SUB(offdate, net, sta)

3.1.5. ALLOCATE_HOUR

Owner(s)	REB
Category	Interactive Processing
Description	The ALLOCATE_HOUR table is used by the analyst_log application to manage analyst schedules and to initiate the process of REB bulletin production.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
I	jdate	*number(8,0)*	Julian date for this data	N
I	hour	*number(2,0)*	Starting hour of analysis	N
I	action	*varchar2(16)*	Analyst task	N
	auth	*varchar2(16)*	Name of the Analyst processing the time block	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Indexes	ALLOCATE_HOURX(jdate, hour, action)

3.1.6. ALLOCATE_HOUR

Owner(s)	LEB
Category	Interactive Processing

Description	The ALLOCATE_HOUR table is used by the analyst_log application to manage analyst schedules and to initiate the process of REB bulletin production.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	jdate	*number(8,0)*	Julian date for this data	N
PI	hour	*number(2,0)*	Starting hour of analysis	N
PI	action	*varchar2(16)*	Analyst task	N
	auth	*varchar2(16)*	Name of the Analyst processing the time block	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_ALLOCATE_HOUR(jdate, hour, action)
Indexes	PK_ALLOCATE_HOUR(jdate, hour, action)

3.1.7. ALLOW_RESID

Owner(s)	LEB
Category	Automatic Processing
Description	The ALLOW_RESID table contains the allowable maximums for the absolute values of residuals for time, azimuth, and slowness. The allowed residuals are a function of phase type and station type (single station or array), and for each of these, the allowed residuals may vary for different distance ranges. A negative value for the residual indicates an arrival field that is not permitted to be defining for that phase, statype, and distance range.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	phase	*varchar2(8)*	Phase type	N
PI	mindelta	*float(24)*	Minimum station to event distance	N
	maxdelta	*float(24)*	Maximum station to event distance	Y
PI	statype	*varchar2(4)*	Station type: single station, array	N
	timeres	*float(24)*	Time residual	Y
	azres	*float(24)*	Azimuth residual	Y
	slores	*float(24)*	Slowness residual (s/deg)	Y
	lddate	*date*	Load date	Y

Primary key	PK_ALLOW_RESID(phase, mindelta, statype)
Indexes	ALLOWPHASEX(phase, statype), PK_ALLOW_RESID(phase, mindelta, statype)

3.1.8. AMP3C

Owner(s)	REB, LEB
Category	Fundamental

Description	The AMP3C table contains amplitude measurements made on three-component data for a specific detection.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	arid	*number(10,0)*	Arrival identifier	N
	cfreq	*float(24)*	Central frequency (Hz)	Y
	vamp	*float(24)*	Vertical amplitude	Y
	vsnr	*float(24)*	Vertical signal-to-noise ratio	Y
	hamp	*float(24)*	Horizontal amplitude	Y
	hsnr	*float(24)*	Horizontal signal-to-noise ratio	Y
	htov	*float(24)*	Horizontal to vertical amplitude ratio	Y
PI	rid	*varchar2(8)*	Recipe identifier	N
	lddate	*date*	Load date	Y

Primary key	PK_AMP3C(arid, rid)
Indexes	AMP3CARIDX(arid), PK_AMP3C(arid, rid)

3.1.9. AMP3C

Owner(s)	IDCX
Category	Fundamental
Description	The AMP3C table contains amplitude measurements made on three-component data for a specific detection.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	arid	*number(10,0)*	Arrival identifier	N
	cfreq	*float(24)*	Central frequency (Hz)	Y
	vamp	*float(24)*	Vertical amplitude	Y
	vsnr	*float(24)*	Vertical signal-to-noise ratio	Y
	hamp	*float(24)*	Horizontal amplitude	Y
	hsnr	*float(24)*	Horizontal signal-to-noise ratio	Y
	htov	*float(24)*	Horizontal to vertical amplitude ratio	Y
PI	rid	*varchar2(8)*	Recipe identifier	N
I	lddate	*date*	Load date	Y

Primary key	PK_AMP3C(arid, rid)
Indexes	AMP3CARIDX(arid), PK_AMP3C(arid, rid), AMP3C_LDDATEX(lddate)

3.1.10. AMPDESCRIPT

Owner(s)	IDCX

Category	Fundamental
Description	The AMPDESCRIPT table contains a description of how each type of amplitude measurement in amplitude table was made.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	amptype	*varchar2(8)*	Amplitude measure descriptor	N
	toff	*float(24)*	Offset from theoretical or observed arrival time	Y
	tlen	*float(24)*	Duration of measurement window	Y
	gvlo	*float(24)*	Low group velocity for measurement window (km/sec)	Y
	gvhi	*float(24)*	High group velocity for measurement window (km/sec)	Y
	mtype	*varchar2(8)*	Measurement type	Y
	descr	*varchar2(255)*	Description	Y
	lddate	*date*	Load date	Y

Primary key	PK_AMPDESCRIPT(amptype)
Indexes	PK_AMPDESCRIPT(amptype)

3.1.11. AMPLITUDE

Owner(s)	LEB
Category	Fundamental
Description	The AMPLITUDE table contains arrival-based and origin-based amplitude measurements. The amplitude measurement is described in AMPDESCRIPT.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	ampid	*number(10,0)*	Amplitude identifier	N
I	arid	*number(10,0)*	Arrival identifier	Y
I	parid	*number(10,0)*	Predicted arrival identifier	Y
	chan	*varchar2(8)*	Channel name	Y
P	amp	*float(24)*	Amplitude	Y
	per	*float(24)*	Period (s)	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	amptime	*float(53)*	Time of amplitude measure	Y
	start_time	*float(53)*	Start time of measurement window	Y
	duration	*float(24)*	Duration of measurement window	Y
	bandw	*float(24)*	Difference between the high and low frequency bands used (Hz).	Y
	amptype	*varchar2(8)*	Amplitude measure descriptor	Y

	units	*varchar2(15)*	Units	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
	inarrival	*varchar2(1)*	"y" or "n" flag indicating if amp is the same in the ARRIVAL table	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
I	lddate	*date*	Load date	Y

Primary key	PK_AMPLITUDE(ampid)
Indexes	AMPPARIDX(parid), AMPLITUDE_LDDATEX(lddate), PK_AMPLITUDE(ampid), AMPARIDX(arid), AMP_PARID_ARID_X(parid, arid)

3.1.12. AMPLITUDE

Owner(s)	REB
Category	Fundamental
Description	The AMPLITUDE table contains arrival-based and origin-based amplitude measurements. The amplitude measurement is described in AMPDESCRIPT.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	ampid	*number(10,0)*	Amplitude identifier	N
I	arid	*number(10,0)*	Arrival identifier	Y
I	parid	*number(10,0)*	Predicted arrival identifier	Y
	chan	*varchar2(8)*	Channel name	Y
P	amp	*float(24)*	Amplitude	Y
	per	*float(24)*	Period (s)	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	amptime	*float(53)*	Time of amplitude measure	Y
	start_time	*float(53)*	Start time of measurement window	Y
	duration	*float(24)*	Duration of measurement window	Y
	bandw	*float(24)*	Difference between the high and low frequency bands used (Hz).	Y
	amptype	*varchar2(8)*	Amplitude measure descriptor	Y
	units	*varchar2(15)*	Units	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
	inarrival	*varchar2(1)*	"y" or "n" flag indicating if amp is the same in the ARRIVAL table	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
I	lddate	*date*	Load date	Y

Primary key	PK_AMPLITUDE(ampid)
Indexes	AMPPARIDX(parid), AMPLITUDE_LDDATEX(lddate), PK_AMPLITUDE(ampid), AMPARIDX(arid)

3.1.13. AMPLITUDE

Owner(s)	IDCX
Category	Fundamental
Description	The AMPLITUDE table contains arrival-based and origin-based amplitude measurements. The amplitude measurement is described in AMPDESCRIPT.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	ampid	*number(10,0)*	Amplitude identifier	N
I	arid	*number(10,0)*	Arrival identifier	Y
I	parid	*number(10,0)*	Predicted arrival identifier	Y
	chan	*varchar2(8)*	Channel name	Y
P	amp	*float(24)*	Amplitude	Y
	per	*float(24)*	Period (s)	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	amptime	*float(53)*	Time of amplitude measure	Y
	start_time	*float(53)*	Start time of measurement window	Y
	duration	*float(24)*	Duration of measurement window	Y
	bandw	*float(24)*	Difference between the high and low frequency bands used (Hz).	Y
	amptype	*varchar2(8)*	Amplitude measure descriptor	Y
	units	*varchar2(15)*	Units	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
	inarrival	*varchar2(1)*	"y" or "n" flag indicating if amp is the same in the ARRIVAL table	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
I	lddate	*date*	Load date	Y

Primary key	PK_AMPLITUDE(ampid)
Indexes	AMPPARIDX(parid), PK_AMPLITUDE(ampid), AMPLDDATEX(lddate), AMPARIDX(arid)

3.1.14. APMA

Owner(s)	REB
Category	Fundamental

Description	The APMA table contains results of particle motion analysis for a specific detection.

Pk/Fk/I	Column	Storage type	Description	Nullable
	phase	*varchar2(8)*	Phase type	Y
PI	arid	*number(10,0)*	Arrival identifier	N
	freq	*float(24)*	Frequency	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	ampp	*float(24)*	P-phase amplitude	Y
	amps	*float(24)*	S-phase amplitude	Y
	amplr	*float(24)*	Rayleigh-phase amplitude	Y
	rect	*float(24)*	Rectilinearity	Y
	plans	*float(24)*	S-phase planarity	Y
	planlr	*float(24)*	Rayleigh-phase planarity	Y
	hvratp	*float(24)*	P-phase horizontal-to-vertical ratio	Y
	hvrat	*float(24)*	S-phase horizontal-to-vertical ratio	Y
	hmxmn	*float(24)*	Maximum-to-minimum horizontal ratio	Y
	inang3	*float(24)*	Short-axis incidence angle	Y
	seazp	*float(24)*	P-phase observed azimuth	Y
	seazs	*float(24)*	S-phase observed azimuth	Y
	seazlr	*float(24)*	Rayleigh-phase observed azimuth	Y
	inang1	*float(24)*	Long-axis incidence angle	Y
	ptime	*float(53)*	P-phase extraction time	Y
	stime	*float(53)*	S-phase extraction time	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	apmarid	*number(10,0)*	Apma recipe identifier	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_APMA(arid)
Indexes	PK_APMA(arid)

3.1.15. APMA

Owner(s)	IDCX, LEB
Category	Fundamental
Description	The APMA table contains results of particle motion analysis for a specific detection.

Pk/Fk/I	Column	Storage type	Description	Nullable
	phase	*varchar2(8)*	Phase type	Y

PI	arid	*number(10,0)*	Arrival identifier	N
	freq	*float(24)*	Frequency	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	ampp	*float(24)*	P-phase amplitude	Y
	amps	*float(24)*	S-phase amplitude	Y
	amplr	*float(24)*	Rayleigh-phase amplitude	Y
	rect	*float(24)*	Rectilinearity	Y
	plans	*float(24)*	S-phase planarity	Y
	planlr	*float(24)*	Rayleigh-phase planarity	Y
	hvratp	*float(24)*	P-phase horizontal-to-vertical ratio	Y
	hvrat	*float(24)*	S-phase horizontal-to-vertical ratio	Y
	hmxmn	*float(24)*	Maximum-to-minimum horizontal ratio	Y
	inang3	*float(24)*	Short-axis incidence angle	Y
	seazp	*float(24)*	P-phase observed azimuth	Y
	seazs	*float(24)*	S-phase observed azimuth	Y
	seazlr	*float(24)*	Rayleigh-phase observed azimuth	Y
	inang1	*float(24)*	Long-axis incidence angle	Y
	ptime	*float(53)*	P-phase extraction time	Y
	stime	*float(53)*	S-phase extraction time	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	apmarid	*number(10,0)*	Amplitude measure descriptor	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_APMA(arid)
Indexes	APMA_LDDATEX(lddate), PK_APMA(arid)

3.1.16. ARCH_DATA_TYPE

Owner(s)	IDCX
Category	Core, Fundamental
Description	The ARCH_DATA_TYPE table contains information used by the archiving software.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	archid	*number(10,0)*	Archive row identifier	N
	datatype	*varchar2(24)*	Data type (or class of data) being archived (e.g. PRIARC, MSG, SQUAL)	Y
	table_name	*varchar2(32)*	Name of the table encapsulating the data type to be archived	Y

	primarykey	*varchar2(24)*	Not used	Y
	secondkey	*varchar2(24)*	Not used	Y
	merge_adjacent	*varchar2(2)*	Y indicates Archive should merge adjacent rows	Y
	note_missing_data	*varchar2(2)*	Indicates that this data type is for late arriving data	Y
	ondate	*number(8,0)*	First julian date this row is valid	Y
	offdate	*number(8,0)*	Last Julian date this row is valid	Y
	lddate	*date*	Load date	Y

Primary key	PK_ARCH_DATA_TYPE(archid)
Indexes	PK_ARCH_DATA_TYPE(archid)

3.1.17. ARRIVAL

Owner(s)	REB
Category	Core, Fundamental
Description	The ARRIVAL table contains summary information about arrivals.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
I	time	*float(53)*	Epoch time of the arrival	N
PI	arid	*number(10,0)*	Arrival identifier	N
	jdate	*number(8,0)*	Julian date for this data	Y
I	stassid	*number(10,0)*	Station association (stassoc) identifier.	Y
	chanid	*number(8,0)*	Channel recording identifier	Y
	chan	*varchar2(8)*	Channel name	Y
	iphase	*varchar2(8)*	Reported phase	Y
	stype	*varchar2(1)*	Signal type	Y
	deltim	*float(24)*	Time uncertainty	Y
	azimuth	*float(24)*	Observed azimuth	Y
	delaz	*float(24)*	Azimuth uncertainty	Y
	slow	*float(24)*	Observed slowness for a detected arrival (s/deg)	Y
	delslo	*float(24)*	Slowness uncertainty (s/deg)	Y
	ema	*float(24)*	Emergence angle	Y
	rect	*float(24)*	Rectilinearity	Y
	amp	*float(24)*	Amplitude	Y
	per	*float(24)*	Period	Y
	logat	*float(24)*	Log(amp/per)	Y

	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
	fm	*varchar2(2)*	First motion	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	qual	*varchar2(1)*	Signal onset quality	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
I	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_ARRIVAL(arid)
Indexes	ARRIVAL_ARID_COMMID_X(arid, commid), ARTIMEX(time), ARSTASSX(stassid), PK_ARRIVAL(arid), ARRIVAL_LDDATEX(lddate)

3.1.18. ARRIVAL

Owner(s)	LEB
Category	Core, Fundamental
Description	The ARRIVAL table contains summary information about arrivals.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
I	time	*float(53)*	Epoch time of the arrival	N
PI	arid	*number(10,0)*	Arrival identifier	N
	jdate	*number(8,0)*	Julian date for this data	Y
I	stassid	*number(10,0)*	Station association (stassoc) identifier.	Y
	chanid	*number(8,0)*	Channel recording identifier	Y
	chan	*varchar2(8)*	Channel name	Y
	iphase	*varchar2(8)*	Reported phase	Y
	stype	*varchar2(1)*	Signal type	Y
	deltim	*float(24)*	Time uncertainty	Y
	azimuth	*float(24)*	Observed azimuth	Y
	delaz	*float(24)*	Azimuth uncertainty	Y
	slow	*float(24)*	Observed slowness for a detected arrival (s/deg)	Y
	delslo	*float(24)*	Slowness uncertainty (s/deg)	Y
	ema	*float(24)*	Emergence angle	Y
	rect	*float(24)*	Rectilinearity	Y
	amp	*float(24)*	Amplitude	Y
	per	*float(24)*	Period	Y

	logat	*float(24)*	Log(amp/per)	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
	fm	*varchar2(2)*	First motion	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	qual	*varchar2(1)*	Signal onset quality	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_ARRIVAL(arid)
Indexes	ARTIMEX(time), ARSTASSX(stassid), ARRIVAL_LDDATEX(lddate), PK_ARRIVAL(arid)

3.1.19. ARRIVAL

Owner(s)	IDCX
Category	Core, Fundamental
Description	The ARRIVAL table contains summary information about arrivals.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	sta	*varchar2(6)*	Station code	N
I	time	*float(53)*	Epoch time of the arrival	N
PI	arid	*number(10,0)*	Arrival identifier	N
	jdate	*number(8,0)*	Julian date for this data	Y
I	stassid	*number(10,0)*	Station association (stassoc) identifier.	Y
	chanid	*number(8,0)*	Channel recording identifier	Y
	chan	*varchar2(8)*	Channel name	Y
	iphase	*varchar2(8)*	Reported phase	Y
	stype	*varchar2(1)*	Signal type	Y
	deltim	*float(24)*	Time uncertainty	Y
	azimuth	*float(24)*	Observed azimuth	Y
	delaz	*float(24)*	Azimuth uncertainty	Y
	slow	*float(24)*	Observed slowness for a detected arrival (s/deg)	Y
	delslo	*float(24)*	Slowness uncertainty (s/deg)	Y
	ema	*float(24)*	Emergence angle	Y
	rect	*float(24)*	Rectilinearity	Y
	amp	*float(24)*	Amplitude	Y

	per	*float(24)*	Period	Y
	logat	*float(24)*	Log(amp/per)	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
	fm	*varchar2(2)*	First motion	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	qual	*varchar2(1)*	Signal onset quality	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_ARRIVAL(arid)
Indexes	ARSTTIMEX(sta, time), ARTIMEX(time), ARSTASSX(stassid), PK_ARRIVAL(arid), ARRIVAL_LDDATEX(lddate)

3.1.20. ASSOC

Owner(s)	REB, LEB
Category	Core, Fundamental
Description	The ASSOC table contains information that connects arrivals (entries in the arrival table) to a particular origin.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
PI	orid	*number(10,0)*	Origin identifier	N
	sta	*varchar2(6)*	Station code	Y
	phase	*varchar2(8)*	Phase type	Y
	belief	*float(24)*	Phase confidence	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	seaz	*float(24)*	Station-to-event azimuth	Y
	esaz	*float(24)*	Event-to-station azimuth	Y
	timeres	*float(24)*	Time residual	Y
	timedef	*varchar2(1)*	Time = defining (d), non defining (n)	Y
	azres	*float(24)*	Azimuth residual	Y
	azdef	*varchar2(1)*	Azimuth = defining (d), non defining (n)	Y
	slores	*float(24)*	Slowness residual (s/km)	Y
	slodef	*varchar2(1)*	Flag indicating whether slowness is defining (d) or non defining (n).	Y
	emares	*float(24)*	Incidence angle residual	Y
	wgt	*float(24)*	Location weight	Y

	vmodel	*varchar2(15)*	Velocity model	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_ASSOC(arid, orid)
Indexes	ASSARIDX(arid), ASSOCX(orid, arid)

3.1.21. ASSOC

Owner(s)	SEL2, SEL3, SEL1
Category	Core, Fundamental
Description	The ASSOC table contains information that connects arrivals (entries in the arrival table) to a particular origin.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
PI	orid	*number(10,0)*	Origin identifier	N
	sta	*varchar2(6)*	Station code	Y
	phase	*varchar2(8)*	Phase type	Y
	belief	*float(24)*	Phase confidence	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	seaz	*float(24)*	Station-to-event azimuth	Y
	esaz	*float(24)*	Event-to-station azimuth	Y
	timeres	*float(24)*	Time residual	Y
	timedef	*varchar2(1)*	Time = defining (d), non defining (n)	Y
	azres	*float(24)*	Azimuth residual	Y
	azdef	*varchar2(1)*	Azimuth = defining (d), non defining (n)	Y
	slores	*float(24)*	Slowness residual (s/km)	Y
	slodef	*varchar2(1)*	Flag indicating whether slowness is defining (d) or non defining (n).	Y
	emares	*float(24)*	Incidence angle residual	Y
	wgt	*float(24)*	Location weight	Y
	vmodel	*varchar2(15)*	Velocity model	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_ASSOC(arid, orid)
Indexes	ASSARIDX(arid), ASSOC_LDDATEX(lddate), ASSOCX(orid, arid)

3.1.22. ASSOC

Owner(s)	IDCX
Category	Core, Fundamental
Description	The ASSOC table contains information that connects arrivals (entries in the arrival table) to a particular origin.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
PI	orid	*number(10,0)*	Origin identifier	N
	sta	*varchar2(6)*	Station code	Y
	phase	*varchar2(8)*	Phase type	Y
	belief	*float(24)*	Phase confidence	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	seaz	*float(24)*	Station-to-event azimuth	Y
	esaz	*float(24)*	Event-to-station azimuth	Y
	timeres	*float(24)*	Time residual	Y
	timedef	*varchar2(1)*	Time = defining (d), non defining (n)	Y
	azres	*float(24)*	Azimuth residual	Y
	azdef	*varchar2(1)*	Azimuth = defining (d), non defining (n)	Y
	slores	*float(24)*	Slowness residual (s/km)	Y
	slodef	*varchar2(1)*	Flag indicating whether slowness is defining (d) or non defining (n).	Y
	emares	*float(24)*	Incidence angle residual	Y
	wgt	*float(24)*	Location weight	Y
	vmodel	*varchar2(15)*	Velocity model	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_ASSOC(arid, orid)
Indexes	ASSARIDX(arid), ASSOCX(orid, arid)

3.1.23. ASSOC_TEMP_GA

Owner(s)	SEL2, SEL3
Category	Core, Fundamental
Description	The ASSOC_TEMP_GA table is used by the Global Association (GA) application to store temporary associations.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	arid	*number(10,0)*	Arrival identifier	N

I	orid	*number(10,0)*	Origin identifier	N
	sta	*varchar2(6)*	Station code	Y
	phase	*varchar2(8)*	Phase type	Y
	belief	*float(24)*	Phase confidence	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	seaz	*float(24)*	Station-to-event azimuth	Y
	esaz	*float(24)*	Event-to-station azimuth	Y
	timeres	*float(24)*	Time residual	Y
	timedef	*varchar2(1)*	Time = defining (d), non defining (n)	Y
	azres	*float(24)*	Azimuth residual	Y
	azdef	*varchar2(1)*	Azimuth = defining (d), non defining (n)	Y
	slores	*float(24)*	Slowness residual (s/km)	Y
	slodef	*varchar2(1)*	Flag indicating whether slowness is defining (d) or non defining (n).	Y
	emares	*float(24)*	Incidence angle residual	Y
	wgt	*float(24)*	Location weight	Y
	vmodel	*varchar2(15)*	Velocity model	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Indexes	ATGORIDX(orid), ASSOC_TEMP_GAX(arid, orid)

3.1.24. ATTENCOEF

Owner(s)	STATIC
Category	Fundamental
Description	The ATTENCOEF table contains station-specific attenuation corrections for regional Event Characterization analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	attenid	*varchar2(20)*	Attenuation coefficient set identifier	N
PI	sta	*varchar2(6)*	Station code	N
PI	ratiotype	*varchar2(6)*	Amplitude ratio identifier	N
PI	chan	*varchar2(8)*	Channel name	N
	xcoef	*float(24)*	Constant coefficient	Y
	ycoef	*float(24)*	Geometrical spreading coefficient	Y
	zcoef	*float(24)*	Attenuation coefficient	Y
	dmin	*float(24)*	Minimum distance of applicability	Y
	dmax	*float(24)*	Maximum distance of applicability	Y
	lddate	*date*	Load date	Y

Primary key	SYS_C0028354(attenid, sta, ratiotype, chan)
Indexes	PK_ATTENCOEF(attenid, sta, ratiotype, chan)

3.1.25. BEAMAUX (VIEW)

Owner(s)	STATIC
Category	Interactive Analysis
Description	The BEAMAUX view joins the sensor, instrument and forbeamaux tables. It is used by the SHI processing application, GSEBull.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
	chan	*varchar2(8)*	Channel name	N
	ncalib	*float(24)*	Nominal calibration (nanometers/digital count)	Y
	ncalper	*float(24)*	Nominal calibration period (seconds)	Y
	time	*float(53)*	Epoch time of start of beam	N
	endtime	*float(53)*	Epoch time of end of beam	N

SQL Body

```
SELECT f.sta,
       f.chan,
       i.ncalib,
       i.ncalper,
       s.time,
       s.endtime
FROM instrument i, sensor s, forbeamaux f
WHERE i.inid=s.inid and s.sta=f.sta and s.chan=f.chan
```

3.1.26. BREGION_DEF

Owner(s)	STATIC
Category	Core, Reference
Description	The BREGION_DEF table holds the latitude and longitude boundaries for each region as well as a region name and region identification number. This table is used by Scanner Tool 2.0.

Pk/Fk/I	Column	Storage type	Description	Nullable
	bregname	*varchar2(32)*	Region name	N
	bregid	*number(8,0)*	Region identification number	N
	lat_lo	*number(8,3)*	Minimum latitude of the region	Y
	lat_hi	*number(8,3)*	Maximum latitude of the region	Y
	lon_lo	*number(8,3)*	Minimum longitude of the region	Y
	lon_hi	*number(8,3)*	Maximum longitude of the region	Y

	lddate	*date*	Load date	Y

3.1.27. BREGION_TAB

Owner(s)	STATIC
Category	Core, Reference
Description	The BREGION_TAB table contains the stations and arrival parameter window attributes and is linked to BREGION_DEF by the bregname and bregid columns. This table is used by Scanner Tool 2.0.

Pk/Fk/I	Column	Storage type	Description	Nullable
	bregname	*varchar2(32)*	Region name	N
	bregid	*number(8,0)*	Region identification number	N
	sta	*varchar2(6)*	Station code	N
	azi_lo	*number(8,3)*	Minimum azimuth	Y
	azi_hi	*number(8,3)*	Maximum azimuth	Y
	slo_lo	*number(8,3)*	Minimal slowness	Y
	slo_hi	*number(8,3)*	Maximal slowness	Y
	snr_lo	*number(8,3)*	Minimal snr	Y
	snr_hi	*number(8,3)*	Maximal snr	Y
	per_lo	*number(8,3)*	Minimum period	Y
	per_hi	*number(8,3)*	Maximum period	Y
	amp_lo	*number(8,3)*	Minimum amplitude	Y
	amp_hi	*number(8,3)*	Maximum amplitude	Y
	lddate	*date*	Load date	Y

3.1.28. CEPPKS

Owner(s)	REB
Category	Fundamental
Description	The CEPPKS table contains results of cepstral analysis and includes the amplitude and frequency of cepstral peaks that are consistent among multiple phases associated with the same event.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
PI	sta	*varchar2(6)*	Station code	N
	ptyp	*varchar2(6)*	Consistent peak type code	Y
	pkamp	*float(24)*	Consistent peak amplitude	Y
PI	pkqf	*float(24)*	Consistent peak quefrency	N

I	lddate	*date*	Load date	Y

Primary key	PK_CEPPKS(orid, sta, pkqf)
Indexes	CEPPKS_LDDATEX(lddate), PK_CEPPKS(orid, sta, pkqf)

3.1.29. CHANNAME

Owner(s)	STATIC
Category	Data Services
Description	The CHANNAME table provides mapping between channel and station names.

Pk/Fk/I	Column	Storage type	Description	Nullable
	station	*varchar2(6)*	Station name	N
PI	stream	*varchar2(6)*	Stream name	N
PI	extern_sta	*varchar2(6)*	External station name	N
PI	extern_chan	*varchar2(8)*	External channel name	N
	intern_sta	*varchar2(6)*	Internal station name	N
	intern_chan	*varchar2(8)*	Internal channel name	N
	capability	*number(4,0)*	Capability	Y
	position	*number(4,0)*	Position	Y
PI	revision	*number(4,0)*	Revision number	N
	lddate	*date*	Load date	Y

Primary key	PK_CHANNAME(stream, extern_sta, extern_chan, revision)
Indexes	PK_CHANNAME(stream, extern_sta, extern_chan, revision)

3.1.30. CHAN_GROUPS

Owner(s)	STATIC
Category	Core, Reference
Description	The CHAN_GROUPS table is used for configuration of station data archiving.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	class	*varchar2(16)*	Type of channel group	N
PI	name	*varchar2(20)*	Station name	N
PI	sta	*varchar2(6)*	Station code	N
PI	chan	*varchar2(8)*	Channel name	N
	duration	*number(8,0)*	Not used	Y
	inwfactivity	*number(1,0)*	Not used	Y
PI	ondate	*number(8,0)*	First julian date this row is valid	N
PI	offdate	*number(8,0)*	Last Julian date this row is valid	N

	lddate	*date*	Load date	Y

Primary key	PK_CHAN_GROUPS(class, name, sta, chan, ondate, offdate)
Indexes	PK_CHAN_GROUPS(class, name, sta, chan, ondate, offdate)

3.1.31. CHAN_GROUPS_AUXWF

Owner(s)	STATIC
Category	Core, Reference
Description	Obsolete. Extract from chan_groups table made in 2012. Not used since.

Pk/Fk/I	Column	Storage type	Description	Nullable
	class	*varchar2(16)*	Type of channel group	Y
	name	*varchar2(20)*	Station name	Y
	sta	*varchar2(6)*	Station code	Y
	chan	*varchar2(8)*	Channel name	Y
	duration	*number(8,0)*	Not used	Y
	inwfactivity	*number(1,0)*	Not used	Y
	ondate	*number(8,0)*	First julian date this row is valid	Y
	offdate	*number(8,0)*	Last Julian date this row is valid	Y
	lddate	*date*	Load date	Y

3.1.32. COLORDISC

Owner(s)	MAP
Category	Interactive Processing
Description	The COLORDISK table links a unique colormapid to a colormap name and disk file.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	colormapid	*number(8,0)*	Colormap identifier	N
	dfile	*varchar2(32)*	Data file name	N
	dir	*varchar2(64)*	Absolute path to the data file	N
	colormapname	*varchar2(64)*	Colormap name	Y
	lddate	*date*	Load date	Y

Primary key	COLORDISC_PK(colormapid)
Indexes	COLORDISC_PK(colormapid)

3.1.33. COMPLEXITY

Owner(s)	REB

Category	Fundamental
Description	The COMPLEXITY table contains the complexity event characterization parameter estimated by the Detection and Feature Extraction (DFX) application.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
PI	sta	*varchar2(6)*	Station code	N
PI	phase	*varchar2(6)*	Phase type	N
	rectype	*varchar2(8)*	Recipe type	Y
	complexity	*float(24)*	Complexity measure	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
I	lddate	*date*	Load date	Y

Primary key	PK_COMPLEXITY(orid, sta, phase)
Indexes	COMPLEXITY_LDDATEX(lddate), COMPLORIDX(orid), PK_COMPLEXITY(orid, sta, phase)

3.1.34. DATAREADY

Owner(s)	IDCX
Category	Data Services
Description	The DATAREADY table indicates which data are ready to be processed by the Subscription Subsystem. Entries to the table are made by applications running in the operational systems that process the data. For applications under the control of the Distributed Application Control System (DACS), such as Station Processing by DFX (Detection and Feature Extraction) and StaPro, this is achieved via a stop-hook call from the tuxshell process controlling the transaction.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	dataid	*number(10,0)*	Data ready identifier	N
	tablename	*varchar2(24)*	Name of table containing data to be distributed by the Subscription system.	Y
	account	*varchar2(24)*	Account name	N
	tagname	*varchar2(12)*	Name of reference field (for example, orid)	Y
I	tagid	*number(10,0)*	Numeric value of the identifier specified by the tagname column	Y
	tagid2	*number(10,0)*	Secondary tag value	Y
	data_info	*varchar2(24)*	Miscellaneous data information	Y
	status	*varchar2(8)*	Status of this row.	Y
I	lddate	*date*	Load date	Y

Primary key	DATAREADY_PK(dataid)
Indexes	PK_DATAREADY(dataid), DRTAGIDX(tagid), DATAREADY_LDDATEX(lddate)
Triggers	DATAREADY_TO_PRODTRACK

3.1.35. DATAUSER

Owner(s)	IDCX
Category	Data Services
Description	The DATAUSER table tracks authorized users of the Message and Subscription Subsystems. Each user is identified by a (unique) username and domain, which must match all email headers. The priority column specifies the class of user, and servicetime is the last time a request from the user was processed. Priority and servicetime are considered when selecting the order in which requests will be processed. The status can either be active or inactive.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	userid	*number(8,0)*	User identifier	N
	pocid	*number(8,0)*	Point of contact identifier	Y
	username	*varchar2(24)*	user name component of email address	N
	domain	*varchar2(48)*	Domain name from the incoming subscription message	N
	msgtype	*varchar2(16)*	Message type	N
	status	*varchar2(24)*	Status of this user	N
	priority	*number(2,0)*	User priority	N
	commid	*number(10,0)*	Comment identifier	Y
	emaillimit	*number(8,0)*	Maximum size of message (in bytes) that will be delivered via email	Y
	servicetime	*float(53)*	Last time a request from that user was serviced	Y
	lddate	*date*	Load date	Y

Primary key	SYS_C0028353(userid)
Indexes	PK_DATAUSER(userid)

3.1.36. DEFINE_PHASE_RESIDUAL

Owner(s)	STATIC
Category	Interactive Analysis
Description	Definitions of allowable phase residuals for use in Interactive SHI analysis

Pk/Fk/I	Column	Storage type	Description	Nullable
	phase	*varchar2(8)*	Phase type	N

	statype	*varchar2(4)*	Station type (array (ar) or three-component (3c))	N
	mindelta	*float(24)*	Minimum allowable distance (degrees)	N
	maxdelta	*float(24)*	Maximum allowable distance (degrees)	N
	mindepth	*float(24)*	Minimum allowable depth (km)	N
	maxdepth	*float(24)*	Maximum allowable depth (km)	N
	timres	*float(24)*	Time residual (seconds)	N
	slores	*float(24)*	Slowness residual (s/km)	N
	azres	*float(24)*	Allowable azimuth residual	N
	lddate	*date*	Load date	N

3.1.37. DERVDISC

Owner(s)	IDCX
Category	Automatic Processing
Description	The DERVDISC table contains pointers to detection pixel information derived from PMCC infrasound processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	sta	*varchar2(6)*	Station code	N
	chan	*varchar2(8)*	Channel name	N
I	time	*float(53)*	Start time of the data interval (epochal)	N
	tlen	*float(24)*	Duration of the data interval in seconds.	Y
	net	*varchar2(8)*	Network identification string	Y
PI	dervid	*number(10,0)*	Derived data unique identifier	N
	recid	*number(10,0)*	Unique record identifier	Y
	method	*varchar2(17)*	Method or Application associated to the dervdisc entry.	Y
	datatype	*varchar2(2)*	Data type (bi: binary as:ASCII)	Y
	dervtype	*varchar2(4)*	Nature of the derived data (pix:PMCC detection pixels, fam: PMCC detection family)	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	dir	*varchar2(64)*	Absolute path to the data file	Y
	dfile	*varchar2(32)*	Name of the data file.	Y
	foff	*number(10,0)*	Byte offset of data segment within file	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	DERVDISC_PK(dervid)
Indexes	DERVDISC_PK(dervid), DERV_STAX(sta, time)

3.1.38. DETECTION

Owner(s)	IDCX, REB, LEB
Category	Fundamental
Description	The DETECTION table contains summary information about SHI detections.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
	jdate	*number(8,0)*	Julian date for this data	Y
I	time	*float(53)*	Epoch time	Y
	sta	*varchar2(6)*	Station code	Y
	chan	*varchar2(8)*	Channel name	Y
	bmtyp	*varchar2(4)*	Beam type	Y
	sproid	*number(10,0)*	Signal processor identifier	Y
	cfreq	*float(24)*	Central frequency (Hz)	Y
	seaz	*float(24)*	Observed azimuth	Y
	delaz	*float(24)*	Azimuth uncertainty	Y
	slow	*float(24)*	Slowness (s/km)	Y
	delslo	*float(24)*	Slowness uncertainty (s/km)	Y
	snr	*float(24)*	Signal-to-noise ratio	Y
	stav	*float(24)*	Short-term average	Y
	fstat	*float(24)*	F-statistic	Y
	deltim	*float(24)*	Time uncertainty	Y
	bandw	*float(24)*	Difference between the high and low frequency bands used (Hz).	Y
	fkqual	*number(4,0)*	F-k quality	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_DETECTION(arid)
Indexes	PK_DETECTION(arid), DETTIMEX(time), DETLDDATEX(lddate)

3.1.39. DISCARD

Owner(s)	LEB
Category	Interactive Processing
Description	The DISCARD table contains the reason why an analyst discarded an event that was hypothesized by the automated system.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	evid	*number(10,0)*	Event identifier	N

PI	reason	*varchar2(30)*	Reason event was discarded	N
	auth	*varchar2(15)*	Name of the Analyst discarding the event	Y
PI	lddate	*date*	Load date	N

Primary key	PK_DISCARD(evid, reason, lddate)
Indexes	DISCARD_LDDATEX(lddate), DISCARDEVIDX(evid), PK_DISCARD(evid, reason, lddate)

3.1.40. DISCARD

Owner(s)	REB
Category	Interactive Processing
Description	The DISCARD table contains the reason why an analyst discarded an event that was hypothesized by the automated system.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	evid	*number(10,0)*	Event identifier	N
PI	reason	*varchar2(30)*	Reason event was discarded	N
	auth	*varchar2(15)*	Name of the Analyst discarding the event	Y
PI	lddate	*date*	Load date	N

Primary key	PK_DISCARD(evid, reason, lddate)
Indexes	DISCARDEVIDX(evid), PK_DISCARD(evid, reason, lddate)

3.1.41. DLMAN

Owner(s)	IDCX
Category	Data Services
Description	Not used at the IDC.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	dlid	*number(10,0)*	diskloop manager identifier	N
	machine	*varchar2(32)*	machine name	N
	running	*varchar2(1)*	DLMan running = y/n	Y
	connmanport	*number(6,0)*	ConnMan port	Y
	controlport	*number(6,0)*	DataControl port	Y
	archiveport	*number(6,0)*	Archiver port	Y
	forwardport	*number(6,0)*	Forwarder port	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Indexes	DLMANX(dlid)

3.1.42. DSEISGRID

Owner(s)	STATIC
Category	Automatic Processing
Description	The DSEISGRID table contains a natural seismicity grid, which includes the average number of events per year with magnitude greater than the threshold in this table for each latitude-longitude grid point (the grid points are defined in the seisindex table). The DSEISGRID table is used by the AEQ application to help identify anomalous events.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	grdname	*varchar2(6)*	Grid name	N
PI	icell	*number(8,0)*	Grid cell index	N
	magth	*float(24)*	Magnitude threshold	Y
	magtype	*varchar2(6)*	Magnitude type, for example mb, Ms, ML	Y
	nevyr	*float(24)*	Average number of events/year	Y
	lddate	*date*	Load date	Y

Primary key	PK_DSEISGRID(grdname, icell)
Indexes	PK_DSEISGRID(grdname, icell)

3.1.43. DSEISINDEX

Owner(s)	STATIC
Category	Automatic Processing
Description	The DESEISINDEX table contains the geographic grids of natural seismicity data in the seisgrid table. The DESEISINDEX table contains the geographic grids of seismicity data in the DSEISGRID table. The DESEISINDEX table is used by the AEQ application to help identify anomalous events.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	grdname	*varchar2(6)*	Grid name	N
	lat1	*float(24)*	Initial latitude	Y
	lon1	*float(24)*	Initial longitude	Y
	dlat	*float(24)*	Latitude increment	Y
	dlon	*float(24)*	Longitude increment	Y
	nlat	*number(8,0)*	Number of latitudes	Y
	nlon	*number(8,0)*	Number of longitudes	Y
	orderby	*varchar2(6)*	Order by either latitude or longitude	Y
	lddate	*date*	Load date	Y

Primary key	PK_DSEISINDEX(grdname)
Indexes	PK_DSEISINDEX(grdname)

3.1.44. EVCHAR_PROD

Owner(s)	IDCX
Description	Obsolete. Never used.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prodid	*number(10,0)*	Obsolete.	N
PI	orid	*number(10,0)*	Obsolete.	N
	ec_icat	*varchar2(8)*	Obsolete.	Y
	ec_score	*float(24)*	Obsolete.	Y
	ec_dscore	*float(24)*	Obsolete.	Y
	ec_mscore	*float(24)*	Obsolete.	Y
	ec_pscore	*float(24)*	Obsolete.	Y
	ec_hscore	*float(24)*	Obsolete.	Y
	ec_iscore	*float(24)*	Obsolete.	Y
	ec_lscore	*float(24)*	Obsolete.	Y
	ec_deptherr	*float(24)*	Obsolete.	Y
	ec_msmb	*float(24)*	Obsolete.	Y
	ec_msmberr	*float(24)*	Obsolete.	Y
	ec_smajax	*float(24)*	Obsolete.	Y
	ec_sminax	*float(24)*	Obsolete.	Y
	ec_strike	*float(24)*	Obsolete.	Y
	ec_consider	*number(8,0)*	Obsolete.	Y
	ec_ievcat	*varchar2(8)*	Obsolete.	Y
	ec_nsta_ms	*number(8,0)*	Obsolete.	Y
	ec_nsta_mb	*number(8,0)*	Obsolete.	Y
	ec_pctoffsh	*float(24)*	Obsolete.	Y
	ec_irej	*varchar2(8)*	Obsolete.	Y
	ec_stamin	*varchar2(6)*	Obsolete.	Y
	ec_pvaluemin	*float(24)*	Obsolete.	Y
I	lddate	*date*	Load date	Y

Primary key	PK_EVCHAR_PROD(prodid, orid)
Indexes	PK_EVCHAR_PROD(prodid, orid), EVCHAR_PROD_LDDATEX(lddate)

3.1.45. EVENT

Owner(s)	SEL2, SEL3, SEL1
Category	Core, Fundamental
Description	The EVENT table contains a list of events. Multiple origins may be defined for any

	one event.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	evid	*number(10,0)*	Event identifier	N
	evname	*varchar2(15)*	Event name	Y
	prefor	*number(10,0)*	Preferred origin	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_EVENT(evid)
Indexes	EVENT_LDDATEX(lddate), PK_EVENT(evid)

3.1.46. EVENT

Owner(s)	REB, LEB
Category	Core, Fundamental
Description	The EVENT table contains a list of events. Multiple origins may be defined for any one event.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	evid	*number(10,0)*	Event identifier	N
	evname	*varchar2(15)*	Event name	Y
	prefor	*number(10,0)*	Preferred origin	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_EVENT(evid)
Indexes	PK_EVENT(evid)

3.1.47. EVENT_CONTROL

Owner(s)	REB
Category	Automatic Processing
Description	The EVENT_CONTROL and IN_EVENT_CONTROL tables contain event location and magnitude control parameters. This information acts as an archive of the specific user-defined controls that were used to determine the location and magnitude of a given orid. The table also includes two measurement columns (cov_sm_axes and cov_depth_time) that allow the coverage ellipse to be determined from the confidence ellipse axes.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
	evid	*number(10,0)*	Event identifier	Y
	prefer_loc	*varchar2(1)*	Preferred location identifier (S, F, R)	Y
	constrain_ot	*number(1,0)*	Flag to constrain origin time	Y
	constrain_latlon	*number(1,0)*	Flag to constrain latitude/longitude	Y
	constrain_depth	*number(1,0)*	Flag to constrain depth	Y
	src_dpnt_corr	*number(2,0)*	Source-dependent correction code	Y
	loc_src_dpnt_reg	*varchar2(15)*	Region name of source-dependent location correction	Y
	loc_sdv_screen	*number(1,0)*	Flag to ignore large data residuals in location	Y
	loc_sdv_mult	*float(24)*	Location large residual multiplier factor	Y
	loc_alpha_only	*number(1,0)*	Flag to use only primary stations in location	Y
	loc_all_stas	*number(1,0)*	Flag to use only stations with src_dpnt_corr	Y
	loc_dist_varwgt	*number(1,0)*	Flag to use distance variance weighting	Y
	mag_src_dpnt_reg	*varchar2(15)*	Region name of source-dependent magnitude correction	Y
	mag_sdv_screen	*number(1,0)*	Flag to ignore large magnitude data residuals	Y
	mag_sdv_mult	*float(24)*	Magnitude large residual multiplier factor	Y
	mag_alpha_only	*number(1,0)*	Flag to limit station net used in magnitude	Y
	mag_all_stas	*number(1,0)*	Flag to use only primary stations in magnitude	Y
	mb_min_dist	*float(24)*	Minimum distance (degrees) for mb	Y
	mb_max_dist	*float(24)*	Maximum distance (degrees) for mb	Y
	mmodel	*varchar2(15)*	Network magnitude model	Y
	cov_sm_axes	*float(24)*	Coverage ellipse semi-axes conversion factor	Y
	cov_depth_time	*float(24)*	Coverage ellipse depth/time conversion factor	Y
	lddate	*date*	Load date	Y

Primary key	PK_EVENT_CONTROL(orid)
Indexes	PK_EVENT_CONTROL(orid)

3.1.48. EVENT_CONTROL

Owner(s)	LEB
Category	Automatic Processing
Description	The EVENT_CONTROL and IN_EVENT_CONTROL tables contain event location and magnitude control parameters. This information acts as an archive of the specific

	user-defined controls that were used to determine the location and magnitude of a given orid. The table also includes two measurement columns (cov_sm_axes and cov_depth_time) that allow the coverage ellipse to be determined from the confidence ellipse axes.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
I	evid	*number(10,0)*	Event identifier	Y
	prefer_loc	*varchar2(1)*	Preferred location identifier (S, F, R)	Y
	constrain_ot	*number(1,0)*	Flag to constrain origin time	Y
	constrain_latlon	*number(1,0)*	Flag to constrain latitude/longitude	Y
	constrain_depth	*number(1,0)*	Flag to constrain depth	Y
	src_dpnt_corr	*number(2,0)*	Source-dependent correction code	Y
	loc_src_dpnt_reg	*varchar2(15)*	Region name of source-dependent location correction	Y
	loc_sdv_screen	*number(1,0)*	Flag to ignore large data residuals in location	Y
	loc_sdv_mult	*float(24)*	Location large residual multiplier factor	Y
	loc_alpha_only	*number(1,0)*	Flag to use only primary stations in location	Y
	loc_all_stas	*number(1,0)*	Flag to use only stations with src_dpnt_corr	Y
	loc_dist_varwgt	*number(1,0)*	Flag to use distance variance weighting	Y
	mag_src_dpnt_reg	*varchar2(15)*	Region name of source-dependent magnitude correction	Y
	mag_sdv_screen	*number(1,0)*	Flag to ignore large magnitude data residuals	Y
	mag_sdv_mult	*float(24)*	Magnitude large residual multiplier factor	Y
	mag_alpha_only	*number(1,0)*	Flag to limit station net used in magnitude	Y
	mag_all_stas	*number(1,0)*	Flag to use only primary stations in magnitude	Y
	mb_min_dist	*float(24)*	Minimum distance (degrees) for mb	Y
	mb_max_dist	*float(24)*	Maximum distance (degrees) for mb	Y
	mmodel	*varchar2(15)*	Network magnitude model	Y
	cov_sm_axes	*float(24)*	Coverage ellipse semi-axes conversion factor	Y
	cov_depth_time	*float(24)*	Coverage ellipse depth/time conversion factor	Y
	lddate	*date*	Load date	Y

Primary key	PK_EVENT_CONTROL(orid)
Indexes	PK_EVENT_CONTROL(orid), EVNTCTRLEVIDX(evid)

3.1.49. EVSC_HYDRO

Owner(s)	IDCX
Category	Automatic Processing
Description	The EVSC_HYDRO table contains station-specific hydroacoustic event-screening results for the standard and custom subscription criteria. The table contains one record per origin identifier (orid) per station (sta).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
PI	sta	*varchar2(6)*	Station code	N
	cp_broad_band	*float(24)*	Hydroacoustic cepstral peak amplitude in 2-80 Hz band	Y
	snr_high_band	*float(24)*	Hydroacoustic signal-to-noise ratio in 32-64 Hz band	Y
	noise_high_band	*float(24)*	Hydroacoustic noise level in 32-64 Hz band	Y
	lddate	*date*	Load date	Y

Primary key	PK_EVSC_HYDRO(orid, sta)
Indexes	PK_EVSC_HYDRO(orid, sta)

3.1.50. EVSC_PROD

Owner(s)	IDCX
Category	Fundamental
Description	The EVSC_PROD table contains event screening results for the standard and custom subscription criteria. The table contains one record per origin identifier (orid) for each subscription (prodid).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prodid	*number(10,0)*	Product identifier	N
PI	orid	*number(10,0)*	Origin identifier	N
	score	*float(24)*	Composite screening score	Y
	dscore	*float(24)*	Depth screening score	Y
	mscore	*float(24)*	Mb minus Ms screening score	Y
	rscore	*float(24)*	Regional P/S screening score	Y
	hscore	*float(24)*	Hydroacoustic screening score	Y
	depth	*float(24)*	Depth estimate (kilometers)	Y
	deptherr	*float(24)*	Depth confidence interval	Y
	moveout_pp	*float(24)*	Moveout of pP-P travel times	Y
	moveout_sp	*float(24)*	Moveout of sP-P travel times	Y
	min_dt_pp	*float(24)*	PP-P travel time difference at nearest station	Y

			beyond 25 deg.	
	min_dt_sp	*float(24)*	SP-P travel time difference at nearest station beyond 25 deg.	Y
	ndp_snr_pp	*number(8,0)*	Number of pP phases with sufficient snr for moveout processing	Y
	ndp_snr_sp	*number(8,0)*	Number of sP phases with sufficient snr for moveout processing	Y
	mbms	*float(24)*	Mb minus Ms (including slope term)	Y
	mbmserr	*float(24)*	Mb minus Ms confidence interval	Y
	mb	*float(24)*	Mb magnitude used for event screening	Y
	ms	*float(24)*	Ms magnitude used for event screening	Y
	nsta_mb	*number(8,0)*	Number of stations used in network mb estimate	Y
	nsta_ms	*number(8,0)*	Number of stations used in network Ms estimate	Y
	smaj_sc	*float(24)*	Scaled semi-major axis of location error ellipse	Y
	smin_sc	*float(24)*	Scaled semi-minor axis of location error ellipse	Y
	strike	*float(24)*	Strike angle of location error ellipse semi-major axis	Y
	pctoffsh	*float(24)*	Percent of location error ellipse that is offshore	Y
	min_wdepth	*float(24)*	Minimum water depth within location error ellipse	Y
	clrpth	*number(1,0)*	Indicator of clear path to at least one hydroacoustic station	Y
	consider	*number(1,0)*	Considered for event screening flag	Y
	lddate	*date*	Load date	Y

Primary key	PK_EVSC_PROD(prodid, orid)
Indexes	PK_EVSC_PROD(prodid, orid), EVSC_PROD_ORIDX(orid)

3.1.51. EVSC_REGIONAL

Owner(s)	IDCX
Category	Fundamental
Description	The EVSC_REGIONAL table contains station-specific regional seismic phase amplitude measurements and quality flags for the standard and subscription criteria. The table contains one record per origin identifier (orid) per station (sta).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N

PI	sta	*varchar2(6)*	Station code	N
PI	chan	*varchar2(8)*	Channel name	N
	delta	*float(24)*	Source-receiver distance in degrees	Y
	pnsmax	*float(24)*	Maximum of Pn/Sn and Pn/Lg	Y
	pnsmax_corr	*float(24)*	Amplitude ratio correction term	Y
	pnsmax_err	*float(24)*	Amplitude ratio error term	Y
	pnsn	*float(24)*	Pn/Sn amplitude ratio	Y
	pnlg	*float(24)*	Pn/Lg amplitude ratio	Y
	pn_snr	*float(24)*	Pn amplitude snr ratio	Y
	sn_snr	*float(24)*	Sn amplitude snr ratio	Y
	lg_snr	*float(24)*	Lg amplitude snr ratio	Y
	pnsn_qual	*varchar2(10)*	Pn/Sn amplitude ratio quality flags	Y
	pnlg_qual	*varchar2(10)*	Pn/Lg amplitude ratio quality flags	Y
	lddate	*date*	Load date	Y

Primary key	PK_EVSC_REGIONAL(orid, sta, chan)
Indexes	PK_EVSC_REGIONAL(orid, sta, chan)

3.1.52. FDSNCHAN

Owner(s)	STATIC
Category	Automatic Processing
Description	The FDSNCHAN table maps FDSN channel names to "old" channel names.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	mapid	*number(10,0)*	Mapping id	N
	oldname	*varchar2(20)*	Old channel name	N
	newname	*varchar2(20)*	FDSN channel name	N

Primary key	FDSNCHAN_PK(mapid)
Indexes	FDSNCHAN_PK(mapid)

3.1.53. FILEPRODUCT

Owner(s)	IDCX
Category	Data Services
Description	The FILEPRODUCT table contains descriptions of data files containing products stored on the UNIX filesystem. The typeid column links to the corresponding text description of that product in the FPDESCRIPTION table.

Pk/Fk/I	Column	Storage type	Description	Nullable

PI	fpid	*number(10,0)*	File product identifier	N
I	typeid	*number(10,0)*	Identifier for the product type	Y
	dir	*varchar2(64)*	Absolute path to the data file	Y
I	dfile	*varchar2(32)*	Name of file containing the fileproduct	Y
	foff	*number(10,0)*	Byte offset of data segment within file	Y
	dsize	*number(10,0)*	Size of the data file (in bytes)	Y
I	time	*float(53)*	Beginning time of the data	Y
I	endtime	*float(53)*	End time of the data	Y
I	sta	*varchar2(6)*	Station code	Y
I	chan	*varchar2(8)*	Channel name	Y
	author	*varchar2(16)*	Author name	Y
	version	*float(53)*	Version of the author	Y
	revision	*number(4,0)*	Revision number	Y
	obsolete	*number(1,0)*	Flag to indicate whether (1) or not (0) the data are obsolete.	Y
I	lddate	*date*	Load date	Y

Primary key	PK_FILEPRODUCT(fpid)
Indexes	FILEPROD_CHANX(chan), FILEPROD_TIMEX(time), FILEPROD_LDDATEX(lddate), FILEPROD_ENDTIMEX(endtime), FILEPROD_TID(typeid, sta), FILEPROD_DFILEX(dfile), PK_FILEPRODUCT(fpid)
Triggers	FILEPRODUCT_TO_DATAREADY, FILEPRODUCT_NULL_LDDATE

3.1.54. FILEPRODUCT_BU_20200610

Owner(s)	IDCX
Description	Not used at the IDC.

Pk/Fk/I	Column	Storage type	Description	Nullable
	fpid	*number(10,0)*		Y
	typeid	*number(10,0)*		Y
	dir	*varchar2(64)*		Y
	dfile	*varchar2(32)*		Y
	foff	*number(10,0)*		Y
	dsize	*number(10,0)*		Y
	time	*float(53)*		Y
	endtime	*float(53)*		Y
	sta	*varchar2(6)*		Y
	chan	*varchar2(8)*		Y

	author	*varchar2(16)*		Y
	version	*float(53)*		Y
	revision	*number(4,0)*		Y
	obsolete	*number(1,0)*		Y
	lddate	*date*		Y

3.1.55. FILEPRODUCT_TEMP

Owner(s)	IDCX
Description	Not used at the IDC.

Pk/Fk/I	Column	Storage type	Description	Nullable
	fpid	*number(10,0)*		Y
	typeid	*number(10,0)*		Y
	dir	*varchar2(64)*		Y
	dfile	*varchar2(32)*		Y
	foff	*number(10,0)*		Y
	dsize	*number(10,0)*		Y
	time	*float(53)*		Y
	endtime	*float(53)*		Y
	sta	*varchar2(6)*		Y
	chan	*varchar2(8)*		Y
	author	*varchar2(16)*		Y
	version	*float(53)*		Y
	revision	*number(4,0)*		Y
	obsolete	*number(1,0)*		Y
	lddate	*date*		Y

3.1.56. FKDISC

Owner(s)	LEB
Category	Interactive Processing
Description	Not used at the IDC.

Pk/Fk/I	Column	Storage type	Description	Nullable
	jdate	*number(8,0)*	Julian date for this data	Y
	time	*float(53)*	Epoch time	N
	tlen	*float(24)*	time window	Y
	sta	*varchar2(6)*	station code	Y

	fktyp	*varchar2(4)*	f-k type	Y
	arid	*number(10,0)*	Arrival identifier	N
	maxkx	*float(24)*	maximum x-wavenumber	Y
	maxsx	*float(24)*	maximum x-slowness	Y
	nx	*number(4,0)*	number of x-samples	Y
	maxky	*float(24)*	maximum y-wavenumber	Y
	maxsy	*float(24)*	maximum y-slowness	Y
	ny	*number(4,0)*	number of y-samples	Y
	cfreq	*float(24)*	center frequency	Y
	bandw	*float(24)*	bandwidth	Y
	commid	*number(10,0)*	Comment identifier	Y
	fkrid	*number(10,0)*	f-k recipe identifier	Y
	fkid	*number(10,0)*	f-k identifier	Y
	datsw	*number(10,0)*	data switch	Y
	foff	*number(10,0)*	Byte offset of data segment within file	Y
	dir	*varchar2(64)*	f-k directory	Y
	dfile	*varchar2(32)*	f-k data file	Y
	lddate	*date*	Load date	Y

3.1.57. FORBEAMAUX

Owner(s)	STATIC
Category	Data Services
Description	The FORBEAMAUX table contains the channels for which amplitude measurements in the event lists and bulletins have been made.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	chan	*varchar2(8)*	Channel name	N

Primary key	PK_FORBEAMAUX(sta, chan)
Indexes	PK_FORBEAMAUX(sta, chan)

3.1.58. FPDESCRIPTION

Owner(s)	IDCX
Category	Data Services
Description	The FPDESCRIPTION table contains descriptions of product types used with file products.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	typeid	*number(10,0)*	Identifier for the product type description	N
	prodtype	*varchar2(12)*	Product type	Y
	name	*varchar2(64)*	Descriptive listing of the product name	Y
	msgdtype	*varchar2(16)*	Type of data (ASCII, GIF89, and so on)	Y
	msgdformat	*varchar2(8)*	Format of the data, for example, binary, ps, IMS1.0, TXT, CD-1.1, CD-1.0, GSE2.0, IMS2.0).	Y
	header_fpid	*number(10,0)*	Fpid pointing to the header row for this product type	Y
	lddate	*date*	Load date	Y

Primary key	PK_FPDESCRIPTION(typeid)
Indexes	PK_FPDESCRIPTION(typeid)

3.1.59. FS_STAGEPRODUCT

Owner(s)	IDCX
Category	Data Services
Description	Obsolete.

Pk/Fk/I	Column	Storage type	Description	Nullable
	fpid	*number(10,0)*	obsolete	Y
	typeid	*number(10,0)*	obsolete	Y
	dir	*varchar2(64)*	obsolete	Y
	dfile	*varchar2(32)*	obsolete	Y
	foff	*number(8,0)*	obsolete	Y
	dsize	*number(8,0)*	obsolete	Y
	time	*float(53)*	obsolete	Y
	endtime	*float(53)*	obsolete	Y
	sta	*varchar2(6)*	obsolete	Y
	chan	*varchar2(8)*	obsolete	Y
	author	*varchar2(16)*	obsolete	Y
	version	*float(53)*	obsolete	Y
	revision	*number(4,0)*	obsolete	Y
	obsolete	*number(1,0)*	obsolete	Y
	lddate	*date*	Load date	Y

3.1.60. FTPFAILED

Owner(s)	IDCX
Category	Data Services
Description	The FTPFAILED table contains information on FTP data retrieval by VDMS, including the msgid (linking to the MSGDISC table), the destination address, number of failed attempts and the epochal time at which the transfer was last attempted.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	msgid	*number(10,0)*	Message identifier	N
	numfailedattempt	*number(4,0)*	Number of failed attempts	Y
	lastfailedtime	*float(53)*	Time of most recent attempt	Y
	ftp_address	*varchar2(64)*	FTP address	Y
	status	*varchar2(8)*	Status of this row.	Y
	lddate	*date*	Load date	Y

Primary key	PK_FTPFAILED(msgid)
Indexes	PK_FTPFAILED(msgid)

3.1.61. FTPLOGIN

Owner(s)	IDCX
Category	Data Services
Description	The FTPLOGIN table contains log on information for FTP data retrieval. The rows are used by the auxiliary data retrieval system to obtain data via FTP from auxiliary stations.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	ftp_address	*varchar2(64)*	FTP address for auxiliary data	N
	username	*varchar2(16)*	UNIX login	Y
	password	*varchar2(16)*	User password for FTP access	Y
	lddate	*date*	Load date	Y

Primary key	FTPLOGIN_PK(ftp_address)
Indexes	FTPLOGIN_PK(ftp_address)

3.1.62. GA_TAG

Owner(s)	SEL1
Category	Automatic Processing
Description	The GA_TAG table contains information on the use of arrivals and origins in the GA application.

Pk/Fk/I	Column	Storage type	Description	Nullable
	objtype	*varchar2(1)*	Type of identifier (a for arrival, o for origin)	Y
I	id	*number(10,0)*	Identification number (arid or orid)	Y
I	state	*varchar2(20)*	Use of arid or orid by the Global Association (GA) subsystem	Y

Indexes	GAIDX(id, state)

3.1.63. GREGION

Owner(s)	STATIC
Category	Core, Reference
Description	The GREGION table contains Flinn-Engdahl (F-E) geographic region numbers and their corresponding descriptions (see [Fli74]).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	grn	*number(8,0)*	Geographic region number	N
I	grname	*varchar2(40)*	Geographic region name	Y
	lddate	*date*	Load date	Y

Primary key	PK_GREGION(grn)
Indexes	PK_GREGION(grn), GREGGRNAMEX(grname)

3.1.64. HYDRO_ARR_GROUP

Owner(s)	IDCX, REB, LEB
Category	Fundamental
Description	The HYDRO_ARR_GROUP table contains hydroacoustic arrival based estimates of slowness and azimuth.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	hydro_id	*number(10,0)*	Hydro-arrival-group identifier	N
	az1	*float(24)*	Azimuth estimated from lag times	Y
	az2	*float(24)*	Second possible azimuth (2 arrivals)	Y
	slow	*float(24)*	Slowness (s/km)	Y
	delaz	*float(24)*	Azimuth uncertainty	Y
	nhydarr	*number(4,0)*	Number of arrivals in hydro-arrival-group	Y
	net	*varchar2(8)*	Network identification string	Y
	hyd_grp_phase	*varchar2(8)*	Hydro-arrival-group phase	Y
	lddate	*date*	Load date	Y

Primary key	PK_HYDRO_ARR_GROUP(hydro_id)
Indexes	PK_HYDRO_ARR_GROUP(hydro_id)

3.1.65. HYDRO_ASSOC

Owner(s)	IDCX, REB, LEB
Category	Fundamental
Description	The HYDRO_ASSOC table contains hydroacoustic arrival based estimates of slowness and azimuth.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
I	hydro_id	*number(10,0)*	Hydro-arrival-group identifier	N
	azcontrib	*varchar2(1)*	Azimuth contribution flag (y or n)	Y
	lddate	*date*	Load date	Y

Primary key	PK_HYDRO_ASSOC(arid)
Indexes	HYDRO_ASSOC_ARIDX(arid, hydro_id), HYDRO_ASSOC_HIDX(hydro_id, arid)

3.1.66. HYDRO_FEATURES

Owner(s)	REB, LEB
Category	Fundamental
Description	The HYDRO_FEATURES table contains feature measurements for hydroacoustic data by the Detection and Feature Extraction (DFX) application.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
	peak_time	*float(53)*	Time of largest absolute signal value	Y
	peak_level	*float(24)*	Pressure of largest absolute signal value	Y
	total_energy	*float(24)*	Total energy in signal	Y
	mean_arrival_time	*float(53)*	Mean arrival time of signal energy	Y
	time_spread	*float(24)*	Rms time spread of signal	Y
	onset_time	*float(53)*	Estimated onset time of signal	Y
	termination_time	*float(53)*	Estimated termination time of signal	Y
	total_time	*float(53)*	Total time above threshold	Y
	num_cross	*number(8,0)*	Number of signal threshold crossings	Y
	ave_noise	*float(24)*	Average pressure of the noise segment	Y
	skewness	*float(24)*	Skew of the estimated signal energy	Y
	kurtosis	*float(24)*	Kurtosis of the estimated signal energy	Y

	cep_var_signal	*float(24)*	Variance of the cepstrum using spectrum	Y
	cep_delay_time_signal	*float(24)*	Bubble pulse delay time using spectrum	Y
	cep_peak_std_signal	*float(24)*	Number of standard deviations from the mean for amplitude using spectrum	Y
	cep_var_trend	*float(24)*	Variance of the cepstrum detrended using Noise Spectrum Equalization (NSE)	Y
	cep_delay_time_trend	*float(24)*	Bubble pulse delay time estimate using trend NSE	Y
	cep_peak_std_trend	*float(24)*	Number of standard deviations from the mean for amplitude using the trend	Y
PI	low_cut	*float(24)*	Low-cut frequency of filter	N
PI	high_cut	*float(24)*	High-cut frequency of filter	N
	ford	*number(8,0)*	Filter order	Y
	ftype	*varchar2(2)*	Filter type	Y
	fzp	*number(8,0)*	Filter causality	Y
	prob_weight_time	*float(53)*	Probability-weighted time	Y
	sigma_time	*float(53)*	Variance of the probability-weighted time	Y
	lddate	*date*	Load date	Y

Primary key	PK_HYDRO_FEATURES(arid, high_cut, low_cut)
Indexes	HYDARIDX(arid), PK_HYDRO_FEATURES(arid, high_cut, low_cut)

3.1.67. HYDRO_FEATURES

Owner(s)	IDCX
Category	Fundamental
Description	The HYDRO_FEATURES table contains feature measurements for hydroacoustic data by the Detection and Feature Extraction (DFX) application.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
	peak_time	*float(53)*	Time of largest absolute signal value	Y
	peak_level	*float(24)*	Pressure of largest absolute signal value	Y
	total_energy	*float(24)*	Total energy in signal	Y
	mean_arrival_time	*float(53)*	Mean arrival time of signal energy	Y
	time_spread	*float(24)*	Rms time spread of signal	Y
	onset_time	*float(53)*	Estimated onset time of signal	Y
	termination_time	*float(53)*	Estimated termination time of signal	Y
	total_time	*float(53)*	Total time above threshold	Y

	num_cross	*number(8,0)*	Number of signal threshold crossings	Y
	ave_noise	*float(24)*	Average pressure of the noise segment	Y
	skewness	*float(24)*	Skew of the estimated signal energy	Y
	kurtosis	*float(24)*	Kurtosis of the estimated signal energy	Y
	cep_var_signal	*float(24)*	Variance of the cepstrum using spectrum	Y
	cep_delay_time_signal	*float(24)*	Bubble pulse delay time using spectrum	Y
	cep_peak_std_signal	*float(24)*	Number of standard deviations from the mean for amplitude using spectrum	Y
	cep_var_trend	*float(24)*	Variance of the cepstrum detrended using Noise Spectrum Equalization (NSE)	Y
	cep_delay_time_trend	*float(24)*	Bubble pulse delay time estimate using trend NSE	Y
	cep_peak_std_trend	*float(24)*	Number of standard deviations from the mean for amplitude using the trend	Y
PI	low_cut	*float(24)*	Low-cut frequency of filter	N
PI	high_cut	*float(24)*	High-cut frequency of filter	N
	ford	*number(8,0)*	Filter order	Y
	ftype	*varchar2(2)*	Filter type	Y
	fzp	*number(8,0)*	Filter causality	Y
	prob_weight_time	*float(53)*	Probability-weighted time	Y
	sigma_time	*float(53)*	Variance of the probability-weighted time	Y
I	lddate	*date*	Load date	Y

Primary key	PK_HYDRO_FEATURES(arid, high_cut, low_cut)
Indexes	HYDARIDX(arid), PK_HYDRO_FEATURES(arid, high_cut, low_cut), HYDRO_FEATURES_LDDATEX(lddate)

3.1.68. INFRA_FEATURES

Owner(s)	REB, LEB
Category	Fundamental
Description	Never used.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	arid	*number(10,0)*	Arrival identifier	N
	eng_time	*float(53)*	energy start time	Y
	eng_dur	*float(24)*	energy duration	Y
	eng_deldur	*float(24)*	standard deviation for energy duration	Y
	coh_time	*float(53)*	coherence start time	Y

	coh_dur	*float(24)*	coherence duration	Y
	coh_deldur	*float(24)*	standard deviation for coherence duration	Y
	coinc_time	*float(53)*	coincident start time	Y
	coinc_dur	*float(24)*	coincident duration	Y
	coinc_deldur	*float(24)*	standard deviation for coincident duration	Y
	ford	*number(4,0)*	Filter order	Y
	zrcr_freq	*float(24)*	zero crossing frequency	Y
	zrcr_delfreq	*float(24)*	standard deviation for zero crossing frequency	Y
	crnr_freq	*float(24)*	corner frequency	Y
	crnr_delfreq	*float(24)*	standard deviation for corner frequency	Y
	coh_per	*float(24)*	spatial coherence period	Y
	coh_snr	*float(24)*	spatial coherence signal-to-noise ratio	Y
	total_energy	*float(24)*	Total energy in signal	Y
	auth	*varchar2(15)*	Source/originator	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Indexes	INFRAARIDX(arid)

3.1.69. INFRA_FEATURES

Owner(s)	IDCX
Category	Fundamental
Description	The INFRA_FEATURES table contains feature measurements for infrasonic data from the DFX application.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	arid	*number(10,0)*	Arrival identifier	N
	eng_time	*float(53)*	energy start time	Y
	eng_dur	*float(24)*	energy duration	Y
	eng_deldur	*float(24)*	standard deviation for energy duration	Y
	coh_time	*float(53)*	coherence start time	Y
	coh_dur	*float(24)*	coherence duration	Y
	coh_deldur	*float(24)*	standard deviation for coherence duration	Y
	coinc_time	*float(53)*	coincident start time	Y
	coinc_dur	*float(24)*	coincident duration	Y
	coinc_deldur	*float(24)*	standard deviation for coincident duration	Y
	ford	*number(4,0)*	Filter order	Y
	zrcr_freq	*float(24)*	zero crossing frequency	Y

	zrcr_delfreq	*float(24)*	standard deviation for zero crossing frequency	Y
	crnr_freq	*float(24)*	corner frequency	Y
	crnr_delfreq	*float(24)*	standard deviation for corner frequency	Y
	coh_per	*float(24)*	spatial coherence period	Y
	coh_snr	*float(24)*	spatial coherence signal-to-noise ratio	Y
	total_energy	*float(24)*	Total energy in signal	Y
	auth	*varchar2(15)*	Source/originator	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_INFRA_FEATURES(arid)
Indexes	PK_INFRA_FEATURES(arid), INFRA_FEATURES_LDDATEX(lddate)

3.1.70. INSTRUMENT

Owner(s)	STATIC
Category	Core, Reference
Description	The INSTRUMENT table contains ancillary calibration information. This table holds nominal one-frequency calibration factors for each instrument and pointers to the nominal frequency-dependent calibration for an instrument. It also holds pointers to the exact calibrations obtained by direct measurement on a particular instrument (see sensor).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	inid	*number(8,0)*	Instrument identifier	N
	insname	*varchar2(50)*	Instrument name	Y
	instype	*varchar2(6)*	Instrument type (e.g. CMG3ES, Akashi, HMP45A)	Y
	band	*varchar2(1)*	Frequency band	Y
	digital	*varchar2(1)*	Data type, digital (d), or analog (a)	Y
	samprate	*float(24)*	Sampling rate (samples per second)	Y
	ncalib	*float(24)*	Nominal calibration (nanometers/digital count)	Y
	ncalper	*float(24)*	Nominal calibration period (seconds)	Y
	dir	*varchar2(64)*	Absolute path to the data file	Y
	dfile	*varchar2(32)*	Instrument response file name	Y
	rsptype	*varchar2(6)*	Response type	Y
	lddate	*date*	Load date	Y

Primary key	PK_INSTRUMENT(inid)
Indexes	PK_INSTRUMENT(inid)

3.1.71. INTERVAL

Owner(s)	IDCX
Category	Distributed Processing
Description	The INTERVAL table defines units of processing. The time, endtime, and name fields indicate processing times for a named object. The class field allows a single interval table to be used for different classes of objects.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	intvlid	*number(10,0)*	Interval identifier	N
I	class	*varchar2(16)*	Type of interval	N
I	name	*varchar2(20)*	String identifying the content of the interval	N
	time	*float(53)*	Starting time of data	Y
I	endtime	*float(53)*	Ending time of data	Y
I	state	*varchar2(16)*	Current processing state	Y
I	moddate	*date*	Time of last processing state change	Y
	lddate	*date*	Load date	Y

Primary key	PK_INTERVAL(intvlid)
Indexes	INMODX(moddate), INENDX(endtime), INCLANASTAMO(class, name, state, moddate), INCLASSX(class, name, endtime), PK_INTERVAL(intvlid)

3.1.72. INTERVAL

Owner(s)	SEGMENT
Category	Distributed Processing
Description	The INTERVAL table defines units of processing. The time, endtime, and name fields indicate processing times for a named object. The class field allows a single interval table to be used for different classes of objects.

Pk/Fk/I	Column	Storage type	Description	Nullable
	intvlid	*number(8,0)*	Interval identifier	N
I	class	*varchar2(16)*	Type of interval	N
I	name	*varchar2(20)*	String identifying the content of the interval	N
	time	*float(53)*	Starting time of data	Y
I	endtime	*float(53)*	Ending time of data	Y
	state	*varchar2(16)*	Current processing state	Y
I	moddate	*date*	Time of last processing state change	Y
	lddate	*date*	Load date	Y

Indexes	INMODX(moddate), INENDX(endtime), INCLASSX(class, name, endtime)

3.1.73. IN_EVENT_CONTROL

Owner(s)	LEB
Category	Automatic Processing
Description	The EVENT_CONTROL and IN_EVENT_CONTROL tables contain event location and magnitude control parameters. This information acts as an archive of the specific user-defined controls that were used to determine the location and magnitude of a given orid. The table also includes two measurement columns (cov_sm_axes and cov_depth_time) that allow the coverage ellipse to be determined from the confidence ellipse axes.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	orid	*number(10,0)*	origin identifier	N
	evid	*number(10,0)*	event identification	Y
	prefer_loc	*varchar2(1)*	preferred location identifier (S, F, R)	Y
	constrain_ot	*number(1,0)*	flag to constrain origin time	Y
	constrain_latlon	*number(1,0)*	flag to constrain latitude/longitude	Y
	constrain_depth	*number(1,0)*	flag to constrain depth	Y
	src_dpnt_corr	*number(2,0)*	source-dependent correction code	Y
	loc_src_dpnt_reg	*varchar2(15)*	region name of source-dependent location correction	Y
	loc_sdv_screen	*number(1,0)*	flag to ignore large data residuals in location	Y
	loc_sdv_mult	*float(24)*	location large residual multiplier factor	Y
	loc_alpha_only	*number(1,0)*	flag to use only primary stations in location	Y
	loc_all_stas	*number(1,0)*	flag to use only stations with src_dpnt_corr	Y
	loc_dist_varwgt	*number(1,0)*	flag to use distance variance weighting	Y
	mag_src_dpnt_reg	*varchar2(15)*	region name of source-dependent magnitude correction	Y
	mag_sdv_screen	*number(1,0)*	flag to ignore large magnitude data residuals	Y
	mag_sdv_mult	*float(24)*	magnitude large residual multiplier factor	Y
	mag_alpha_only	*number(1,0)*	flag to limit station net used in magnitude	Y
	mag_all_stas	*number(1,0)*	flag to use only primary stations in mag-nitude	Y
	mb_min_dist	*float(24)*	minimum distance (degrees) for mb	Y
	mb_max_dist	*float(24)*	maximum distance (degrees) for mb	Y
	mmodel	*varchar2(15)*	network magnitude model	Y
	cov_sm_axes	*float(24)*	coverage ellipse semi-axes conversion factor	Y
	cov_depth_time	*float(24)*	coverage ellipse depth/time conversion factor	Y
	lddate	*date*	Load date	Y

3.1.74. LASTID

Owner(s)	IDCX
Category	Database and Utility
Description	The LASTID table contains counter values (last value used for keys). This table is a reference table from which programs may retrieve the last sequential value of one of the numeric keys. Unique keys are required before inserting a record in numerous tables. The table has exactly one row for each keyname. PROBLASTID and RMS_LASTID are views of the LASTID table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	keyname	*varchar2(15)*	Identifier name (arid, orid, and so on)	N
	keyvalue	*number(10,0)*	Last value used for the identifier specified in the keyname column	N
	lddate	*date*	Load date	Y

Primary key	LASTID_PK(keyname)
Indexes	LASTIDX(keyname)
Triggers	LASTID_CHECKS

3.1.75. LASTID

Owner(s)	SEGMENT
Category	Database and Utility
Description	The LASTID table contains counter values (last value used for keys). This table is a reference table from which programs may retrieve the last sequential value of one of the numeric keys. Unique keys are required before inserting a record in numerous tables. The table has exactly one row for each keyname. PROBLASTID and RMS_LASTID are views of the LASTID table.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	keyname	*varchar2(15)*	Identifier name (arid, orid, and so on)	N
	keyvalue	*number(10,0)*	Last value used for the identifier specified in the keyname column	N
	lddate	*date*	Load date	Y

Indexes	LASTIDX(keyname)

3.1.76. MAPCOLOR

Owner(s)	MAP
Category	Interactive Processing
Description	The MAPCOLOR table contains information to associate a mapid from the MAP-DISC table with a colormapid from the COLORDISC table. This table is used to plot the same map (mapid) in different colors (for example, brown, green, or outline).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	mapid	*number(8,0)*	Map identifier	N
PI	colormapid	*number(8,0)*	Colormap identifier	N
	lddate	*date*	Load date	Y

Primary key	MAPCOLOR_PK(mapid, colormapid)
Indexes	MAPCOLOR_PK(mapid, colormapid)

3.1.77. MAPDISC

Owner(s)	MAP
Category	Interactive Processing
Description	The MAPDISC table contains information about map files that are on disk.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	mapid	*number(8,0)*	Map identifier	N
	mapname	*varchar2(64)*	Map name	N
	dfile	*varchar2(32)*	Map data filename	N
	dir	*varchar2(64)*	Absolute path to the data file	N
	maptype	*number(8,0)*	Map type	N
	mapfiletype	*varchar2(4)*	Map file type	N
	projection	*number(8,0)*	Map projection	N
	dimx	*number(8,0)*	Map x dimension	N
	dimy	*number(8,0)*	Map y dimension	N
	reflon	*float(24)*	Reference longitude	Y
	reflat	*float(24)*	Reference latitude	Y
	refoffsetlon	*float(24)*	Longitude reference offset	Y
	refoffsetlat	*float(24)*	Latitude reference offset	Y
	lonorigradians	*float(24)*	Longitude origin radians	Y
	latorigradians	*float(24)*	Latitude origin radians	Y
	scale	*float(24)*	Map scale	Y
	rotation	*float(24)*	Map rotation	Y
	latminor	*float(24)*	Latitude interval for minor grid lines	Y
	latmajor	*float(24)*	Latitude interval for major grid lines	Y
	lonminor	*float(24)*	Longitude interval for minor grid lines	Y
	lonmajor	*float(24)*	Longitude interval for major grid lines	Y
	bordercolor	*varchar2(32)*	Border color name	Y
	label	*varchar2(65)*	Map category	Y
	lddate	*date*	Load date	Y

	gctp1	*float(24)*	General cartographic transformation package variables	Y
	gctp2	*float(24)*	General cartographic transformation package variables	Y
	gctp3	*float(24)*	General cartographic transformation package variables	Y
	gctp4	*float(24)*	General cartographic transformation package variables	Y
	gctp5	*float(24)*	General cartographic transformation package variables	Y
	gctp6	*float(24)*	General cartographic transformation package variables	Y
	gctp7	*float(24)*	General cartographic transformation package variables	Y
	gctp8	*float(24)*	General cartographic transformation package variables	Y
	gctp9	*float(24)*	General cartographic transformation package variables	Y
	gctp10	*float(24)*	General cartographic transformation package variables	Y
	gctp11	*float(24)*	General cartographic transformation package variables	Y
	gctp12	*float(24)*	General cartographic transformation package variables	Y
	gctp13	*float(24)*	General cartographic transformation package variables	Y
	gctp14	*float(24)*	General cartographic transformation package variables	Y
	gctp15	*float(24)*	General cartographic transformation package variables	Y

Primary key	MAPDISC_PK(mapid)
Indexes	MAPDISC_PK(mapid)

3.1.78. MAPOVER

Owner(s)	MAP
Category	Interactive Processing
Description	The MAPOVER table contains links between the MAPDISC and OVERLAYDISC tables.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	mapid	*number(8,0)*	Map identifier	N
PI	overlayid	*number(8,0)*	Overlay identifier	N

	lddate	*date*	Load date	Y

Primary key	MAPOVER_PK(mapid, overlayid)
Indexes	MAPOVER_PK(mapid, overlayid)

3.1.79. MAPPOINT

Owner(s)	MAP
Category	Interactive Processing
Description	The MAPPOINT table contains labeled point data to be displayed by the Map application.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	lat	*float(24)*	Geographic latitude. Locations north of the equator have positive latitudes.	N
PI	lon	*float(24)*	Geographic longitude. Longitudes are measured positive east of the Greenwich meridian.	N
	mplabel	*varchar2(65)*	Map point label	Y
PI	mptype	*varchar2(20)*	Map point type	N
	mpdescrip	*varchar2(50)*	Map point description	Y
	lddate	*date*	Load date	Y

Primary key	MAPPOINT_PK(lat, lon, mptype)
Indexes	MAPPOINT_PK(lat, lon, mptype)

3.1.80. MSGAUX

Owner(s)	IDCX
Category	Data Services
Description	The MSGAUX table contains records of unsuccessfully processed VDMS messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	msgid	*number(10,0)*	Message identifier	N
PI	msgrow	*number(4,0)*	Line number in message	N
PI	state_count	*number(4,0)*	Number of failures	N
	command	*varchar2(24)*	VDMS command that could not be processed	Y
	sub_status	*varchar2(24)*	Cause of failure	Y
I	lddate	*date*	Load date	Y

Primary key	PK_MSGAUX(msgid, msgrow, state_count)
Indexes	MGSXMSGIDX(msgid), PK_MSGAUX(msgid, msgrow, state_count), MSGAUX_LDDATEX(lddate)

3.1.81. MSGDATATYPE

Owner(s)	IDCX
Category	Data Services
Description	The MSGDATATYPE table supports data tracking by recording each data section in a message for both incoming and outgoing VDMS data messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	msgid	*number(10,0)*	Message identifier	N
	msgdtype	*varchar2(16)*	Data type of the data section within the message	Y
	msgdformat	*varchar2(16)*	Format of the data, currently one of GSE2.0, IMS1.0 or IMS2.0.	Y
	status	*varchar2(32)*	Status of this row.	Y
PI	foff	*number(8,0)*	Byte offset of data segment within file	N
	msize	*number(10,0)*	Size of data section	Y
I	lddate	*date*	Load date	Y

Primary key	PK_MSGDATATYPE(msgid, foff)
Indexes	MSGTYPMSGIDX(msgid), MSGDATATYPE_LDDATEX(lddate), PK_MSGDATATYPE(msgid, foff)

3.1.82. MSGDEST

Owner(s)	IDCX
Category	Data Services
Description	The MSGDEST table contains information about VDMS messages sent from the IDC.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	msgdid	*number(10,0)*	Message destination identifier	N
I	msgid	*number(10,0)*	Message identifier of the VDMS response	Y
	transmeth	*varchar2(16)*	Method by which the response is to be delivered to the requester	Y
	emailto	*varchar2(64)*	Email address to send message	Y
I	status	*varchar2(32)*	Status of this row.	Y
	itime	*float(53)*	Initial time when message was received	Y
	stime	*float(53)*	Epochal time at which message was sent	Y
I	lddate	*date*	Load date	Y

Primary key	PK_MSGDEST(msgdid)
Indexes	MSGDSTSTATX(status), MSGDSTMSGIDZ(msgid), MSGDEST_LDDATEX(ld-date), PK_MSGDEST(msgdid)

3.1.83. MSGDISC

Owner(s)	IDCX
Category	Data Services
Description	The MSGDISC table contains information pertinent to VDMS messages including the date and time that the message was sent or received, identification information, and where the message is stored.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	msgid	*number(10,0)*	Message identifier	N
	userid	*number(8,0)*	User identifier	Y
	msgver	*varchar2(8)*	Message system version number	Y
	msgtype	*varchar2(16)*	Message type	Y
	subtype	*varchar2(2)*	Message subtype	Y
	extmsgid	*varchar2(20)*	Message identification string provided by the sender	Y
I	intid	*number(10,0)*	Internal identifier for message tracking	Y
	intidtype	*varchar2(16)*	Identifier type for the intid	Y
	msgsrc	*varchar2(16)*	Message source code	Y
I	itime	*float(53)*	Initial time when message was received	Y
I	idate	*number(8,0)*	Initial date message was received	Y
	imethod	*varchar2(8)*	Input method (email or ftp)	Y
	isrc	*varchar2(64)*	Initial source of message	Y
	msize	*number(10,0)*	Message size in bytes	Y
I	status	*varchar2(32)*	Status of this row.	Y
	subject	*varchar2(64)*	Subject header from email message	Y
	dir	*varchar2(64)*	Absolute path to the data file	Y
	dfile	*varchar2(32)*	Name of message file	Y
	foff	*number(10,0)*	Byte offset of data segment within file	Y
	mfoff	*number(10,0)*	Offset in bytes to beginning of message	Y
	fileoff	*number(10,0)*	Number of bytes to the first character of the email file (first character of the email header)	Y
	filesize	*number(10,0)*	Size of file	Y
	sigtype	*varchar2(64)*	Digital signature type	Y
	verifstatus	*varchar2(4)*	Authentication status	Y

	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_MSGDISC(msgid)
Indexes	MSGDISC_LDDATEX(lddate), MSGIDATEX(idate), MSGSTATX(status), MSGINTIDX(intid), PK_MSGDISC(msgid), MSGDISC_ITIMEX(itime)

3.1.84. NA_VALUE

Owner(s)	IDCX
Category	Database and Utility
Description	The NA_VALUE table contains the default values to be inserted for a column when not available.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	attribute	*varchar2(30)*	Column name	N
	na_value	*varchar2(30)*	Value inserted when not available	Y

Primary key	PK_NA_VALUE(attribute)
Indexes	PK_NA_VALUE(attribute)

3.1.85. NETMAG

Owner(s)	SEL3, REB, SEL1, SEL2, LEB
Category	Core, Fundamental
Description	The NETMAG table contains estimates of network magnitudes of different types for an event. Each network magnitude has a unique magid. Station magnitudes used to compute the network magnitude are in the STAMAG table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	magid	*number(10,0)*	Network magnitude identifier	N
	net	*varchar2(8)*	Network identification string	Y
I	orid	*number(10,0)*	Origin identifier	N
	evid	*number(10,0)*	Event identifier	Y
	magtype	*varchar2(6)*	Magnitude type, for example mb, Ms, ML	Y
	nsta	*number(8,0)*	Number of stations used	Y
	magnitude	*float(24)*	Magnitude	Y
	uncertainty	*float(24)*	Magnitude uncertainty	Y
	auth	*varchar2(15)*	Source/originator	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_NETMAG(magid)
Indexes	PK_NETMAG(magid), NETMAG_LDDATEX(lddate), NMORIDX(orid)

3.1.86. NETWORK

Owner(s)	STATIC
Category	Core, Reference
Description	The NETWORK table contains general information about seismic networks (see AFFILIATION).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	net	*varchar2(8)*	Network identification string	N
	netname	*varchar2(80)*	Network name	Y
	nettype	*varchar2(4)*	Network type: array, local, world-wide, and so on	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_NETWORK(net)
Indexes	PK_NETWORK(net)

3.1.87. NOOPS_CHANNAME

Owner(s)	STATIC
Description	Obsolete.

Pk/Fk/I	Column	Storage type	Description	Nullable
	station	*varchar2(6)*	Station name	N
	stream	*varchar2(6)*	Stream name	N
	extern_sta	*varchar2(6)*	External station name	N
	extern_chan	*varchar2(8)*	External channel name	N
	intern_sta	*varchar2(6)*	Internal station name	N
	intern_chan	*varchar2(8)*	Internal channel name	N
	capability	*number(4,0)*	Capability	Y
	position	*number(4,0)*	Position	Y
	revision	*number(4,0)*	Revision number	Y
	lddate	*date*	Load date	Y

3.1.88. OLD_WFPROTO (VIEW)

Owner(s)	IDCX
Description	Obsolete.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
	chan	*varchar2(8)*	Channel name	N
	time	*float(53)*	Start epochal time of the waveform segment	N
	wfid	*number*	Unique waveform identifier (integer)	Y
	chanid	*number(8,0)*	Channel recording identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	endtime	*float(53)*	End epochal time of the waveform segment	N
	nsamp	*number*	Number of samples in the waveform segment	Y
	samprate	*float(24)*	Sample rate for this record (samples per second)	Y
	calib	*float(24)*	Nominal calibration	Y
	calper	*float(24)*	Calibration period (seconds)	Y
	instype	*varchar2(6)*	Instrument type (e.g. CMG3ES, Akashi, HMP45A)	Y
	segtype	*char(1)*	Waveform segment type indicating if the waveform is original, segmented, virtual or duplicate	Y
	datatype	*varchar2(2)*	Data type (e.g. s3, s4, t4)	Y
	clip	*char(1)*	Flag indicating whether or not the data are clipped	Y
	dir	*varchar2(64)*	Absolute path to the data file	Y
	dfile	*varchar2(32)*	Waveform data file name	Y
	foff	*number*	Byte offset of data segment within file	Y
	commid	*number*	Comment identifier	Y
	lddate	*date*	Load date	Y

SQL Body	

```
SELECT      s.sta,
      s.chan,
      s.time,
      -1          wfid,
      s.chanid,
      s.jdate,
      s.endtime,
      0           nsamp,
      i.samprate,
      i.ncalib    calib,
      i.ncalper   calper,
      i.instype,
      '-'         segtype,
      c.outtype   datatype,
      '-'         clip,
      rpad('-', 64, ' ')      dir,
      rpad('-', 32, ' ')      dfile,
      0           foff,
      -1          commid,
      sysdate           lddate
FROM sensor s, instrument i, wfconv c
WHERE c.chanid=s.chanid and i.inid=s.inid
AND c.sta = s.sta and s.endtime > 9999999999
```

3.1.89. ORIGERR

Owner(s)	IDCX, REB
Category	Core, Fundamental
Description	The ORIGERR table contains summaries of confidence bounds in origin estimations. The measurement fields are the elements of the location covariance matrix. The descriptive fields give the uncertainties in location, depth, and origin time. These quantities are calculated from the covariance matrix, assuming gaussian errors and a confidence level.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
	sxx	*float(24)*	Covariance matrix element	Y
	syy	*float(24)*	Covariance matrix element	Y
	szz	*float(24)*	Covariance matrix element	Y
	stt	*float(24)*	Covariance matrix element	Y
	sxy	*float(24)*	Covariance matrix element	Y
	sxz	*float(24)*	Covariance matrix element	Y
	syz	*float(24)*	Covariance matrix element	Y
	stx	*float(24)*	Covariance matrix element	Y
	sty	*float(24)*	Covariance matrix element	Y
	stz	*float(24)*	Covariance matrix element	Y

	sdobs	*float(24)*	Standard error of observations	Y
	smajax	*float(24)*	Semi-major axis of error	Y
	sminax	*float(24)*	Semi-minor axis of error	Y
	strike	*float(24)*	Strike angle of location error ellipse semi-major axis	Y
	sdepth	*float(24)*	Depth error	Y
	stime	*float(24)*	Origin time error (seconds)	Y
	conf	*float(24)*	Confidence	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_ORIGERR(orid)
Indexes	PK_ORIGERR(orid)

3.1.90. ORIGERR

Owner(s)	SEL2, SEL3, SEL1, LEB
Category	Core, Fundamental
Description	The ORIGERR table contains summaries of confidence bounds in origin estimations. The measurement fields are the elements of the location covariance matrix. The descriptive fields give the uncertainties in location, depth, and origin time. These quantities are calculated from the covariance matrix, assuming gaussian errors and a confidence level.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
	sxx	*float(24)*	Covariance matrix element	Y
	syy	*float(24)*	Covariance matrix element	Y
	szz	*float(24)*	Covariance matrix element	Y
	stt	*float(24)*	Covariance matrix element	Y
	sxy	*float(24)*	Covariance matrix element	Y
	sxz	*float(24)*	Covariance matrix element	Y
	syz	*float(24)*	Covariance matrix element	Y
	stx	*float(24)*	Covariance matrix element	Y
	sty	*float(24)*	Covariance matrix element	Y
	stz	*float(24)*	Covariance matrix element	Y
	sdobs	*float(24)*	Standard error of observations	Y
	smajax	*float(24)*	Semi-major axis of error	Y
	sminax	*float(24)*	Semi-minor axis of error	Y
	strike	*float(24)*	Strike angle of location error ellipse semi-major axis	Y

	sdepth	*float(24)*	Depth error	Y
	stime	*float(24)*	Origin time error (seconds)	Y
	conf	*float(24)*	Confidence	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_ORIGERR(orid)
Indexes	PK_ORIGERR(orid), ORIGERR_LDDATEX(lddate)

3.1.91. ORIGERR_TEMP_GA

Owner(s)	SEL2, SEL3
Category	Core, Fundamental
Description	The ORIGERR_TEMP_GA table is used by the GA application to store temporary origin error information. Its structure is identical to the ORIGERR table. The measurement fields are the elements of the location covariance matrix. The descriptive fields give the uncertainties in location, depth, and origin time. These quantities are calculated from the covariance matrix, assuming gaussian errors and a confidence level.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	orid	*number(10,0)*	Origin identifier	N
	sxx	*float(24)*	Covariance matrix element	Y
	syy	*float(24)*	Covariance matrix element	Y
	szz	*float(24)*	Covariance matrix element	Y
	stt	*float(24)*	Covariance matrix element	Y
	sxy	*float(24)*	Covariance matrix element	Y
	sxz	*float(24)*	Covariance matrix element	Y
	syz	*float(24)*	Covariance matrix element	Y
	stx	*float(24)*	Covariance matrix element	Y
	sty	*float(24)*	Covariance matrix element	Y
	stz	*float(24)*	Covariance matrix element	Y
	sdobs	*float(24)*	Standard error of observations	Y
	smajax	*float(24)*	Semi-major axis of error	Y
	sminax	*float(24)*	Semi-minor axis of error	Y
	strike	*float(24)*	Strike angle of location error ellipse semi-major axis	Y
	sdepth	*float(24)*	Depth error	Y
	stime	*float(24)*	Origin time error (seconds)	Y
	conf	*float(24)*	Confidence	Y
	commid	*number(10,0)*	Comment identifier	Y

	lddate	*date*	Load date	Y

3.1.92. ORIGIN

Owner(s)	SEL2, SEL3, IDCX, SEL1
Category	Core, Fundamental
Description	The ORIGIN table contains information describing a derived or reported origin for a particular event.

Pk/Fk/I	Column	Storage type	Description	Nullable
	lat	*float(24)*	Estimated latitude	N
	lon	*float(24)*	Estimated longitude	N
	depth	*float(24)*	Estimated depth (kilometers)	N
I	time	*float(53)*	Epoch time	N
PI	orid	*number(10,0)*	Unique origin identifier	N
	evid	*number(10,0)*	Event identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	nass	*number(4,0)*	Number of associated phases	Y
	ndef	*number(4,0)*	Number of time-defining phases	Y
	ndp	*number(4,0)*	Number of depth phases	Y
	grn	*number(8,0)*	Geographic region number	Y
	srn	*number(8,0)*	Seismic region number, as given in [Fli74]	Y
	etype	*varchar2(7)*	Event type	Y
	depdp	*float(24)*	Estimated depth from depth phases	Y
	dtype	*varchar2(1)*	Depth method used	Y
	mb	*float(24)*	Body wave magnitude	Y
	mbid	*number(10,0)*	Mb magnitude identifier	Y
	ms	*float(24)*	Surface wave magnitude (Ms)	Y
	msid	*number(10,0)*	Ms magnitude identifier	Y
	ml	*float(24)*	Local magnitude	Y
	mlid	*number(10,0)*	ML magnitude identifier	Y
	algorithm	*varchar2(15)*	Location algorithm used	Y
	auth	*varchar2(15)*	Source/originator	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_ORIGIN(orid)
Indexes	PK_ORIGIN(orid), ORTIMEX(time), ORIGIN_LDDATEX(lddate)

3.1.93. ORIGIN

Owner(s)	LEB
Category	Core, Fundamental
Description	The ORIGIN table contains information describing a derived or reported origin for a particular event.

Pk/Fk/I	Column	Storage type	Description	Nullable
	lat	*float(24)*	Estimated latitude	N
	lon	*float(24)*	Estimated longitude	N
	depth	*float(24)*	Estimated depth (kilometers)	N
I	time	*float(53)*	Epoch time	N
PI	orid	*number(10,0)*	Unique origin identifier	N
I	evid	*number(10,0)*	Event identifier	Y
I	jdate	*number(8,0)*	Julian date for this data	Y
	nass	*number(4,0)*	Number of associated phases	Y
	ndef	*number(4,0)*	Number of time-defining phases	Y
	ndp	*number(4,0)*	Number of depth phases	Y
	grn	*number(8,0)*	Geographic region number	Y
	srn	*number(8,0)*	Seismic region number, as given in [Fli74]	Y
	etype	*varchar2(7)*	Event type	Y
	depdp	*float(24)*	Estimated depth from depth phases	Y
	dtype	*varchar2(1)*	Depth method used	Y
	mb	*float(24)*	Body wave magnitude	Y
	mbid	*number(10,0)*	Mb magnitude identifier	Y
	ms	*float(24)*	Surface wave magnitude (Ms)	Y
	msid	*number(10,0)*	Ms magnitude identifier	Y
	ml	*float(24)*	Local magnitude	Y
	mlid	*number(10,0)*	ML magnitude identifier	Y
	algorithm	*varchar2(15)*	Location algorithm used	Y
	auth	*varchar2(15)*	Source/originator	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_ORIGIN(orid)
Indexes	EVIDX(evid), ANLOG_JDATEX(jdate), ORTIMEX(time), PK_ORIGIN(orid), ORIGIN_LDDATEX(lddate)

3.1.94. ORIGIN

Owner(s)	REB
Category	Core, Fundamental
Description	The ORIGIN table contains information describing a derived or reported origin for a particular event.

Pk/Fk/I	Column	Storage type	Description	Nullable
	lat	*float(24)*	Estimated latitude	N
	lon	*float(24)*	Estimated longitude	N
	depth	*float(24)*	Estimated depth (kilometers)	N
I	time	*float(53)*	Epoch time	N
PI	orid	*number(10,0)*	Unique origin identifier	N
	evid	*number(10,0)*	Event identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	nass	*number(4,0)*	Number of associated phases	Y
	ndef	*number(4,0)*	Number of time-defining phases	Y
	ndp	*number(4,0)*	Number of depth phases	Y
	grn	*number(8,0)*	Geographic region number	Y
	srn	*number(8,0)*	Seismic region number, as given in [Fli74]	Y
	etype	*varchar2(7)*	Event type	Y
	depdp	*float(24)*	Estimated depth from depth phases	Y
	dtype	*varchar2(1)*	Depth method used	Y
	mb	*float(24)*	Body wave magnitude	Y
	mbid	*number(10,0)*	Mb magnitude identifier	Y
	ms	*float(24)*	Surface wave magnitude (Ms)	Y
	msid	*number(10,0)*	Ms magnitude identifier	Y
	ml	*float(24)*	Local magnitude	Y
	mlid	*number(10,0)*	ML magnitude identifier	Y
	algorithm	*varchar2(15)*	Location algorithm used	Y
	auth	*varchar2(15)*	Source/originator	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_ORIGIN(orid)
Indexes	PK_ORIGIN(orid), ORTIMEX(time)

3.1.95. ORIGIN_TEMP_GA

Owner(s)	SEL2, SEL3

Category	Core, Fundamental
Description	The ORIGIN_TEMP_GA table contains information describing a derived or reported origin for a particular event. Its structure is identical to the ORIGIN table. The ORIGIN_TEMP_GA table is used by the GA application to store temporary origins.

Pk/Fk/I	Column	Storage type	Description	Nullable
	lat	*float(24)*	Geographic latitude. Locations north of the equator have positive latitudes.	N
	lon	*float(24)*	Geographic longitude. Longitudes are measured positive east of the Greenwich meridian.	N
	depth	*float(24)*	Estimated depth (kilometers)	N
I	time	*float(53)*	Epoch time	N
I	orid	*number(10,0)*	Origin identifier	N
	evid	*number(10,0)*	Event identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	nass	*number(4,0)*	Number of associated phases	Y
	ndef	*number(4,0)*	Number of time-defining phases	Y
	ndp	*number(4,0)*	Number of depth phases	Y
	grn	*number(8,0)*	Geographic region number	Y
	srn	*number(8,0)*	Seismic region number, as given in [Fli74]	Y
	etype	*varchar2(7)*	Event type	Y
	depdp	*float(24)*	Estimated depth from depth phases	Y
	dtype	*varchar2(1)*	Depth method used	Y
	mb	*float(24)*	Body wave magnitude	Y
	mbid	*number(10,0)*	Mb magnitude identifier	Y
	ms	*float(24)*	Surface wave magnitude (Ms)	Y
	msid	*number(10,0)*	Ms magnitude identifier	Y
	ml	*float(24)*	Local magnitude	Y
	mlid	*number(10,0)*	ML magnitude identifier	Y
	algorithm	*varchar2(15)*	Location algorithm used	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Indexes	OTGTIMEX(time), ORIGIN_TEMP_GAX(orid)

3.1.96. OUTAGE

Owner(s)	IDCX

Category	Data Services
Description	The OUTAGE table contains the availability of time-series data. If the available column is set to "f," it specifies that no data are available for the interval. Conversely, the outage table can be used to specify that data are available for an interval by setting the available column to "t."

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	otgid	*number(10,0)*	Outage identifier	N
I	sta	*varchar2(6)*	Station code	N
I	chan	*varchar2(8)*	Channel name	Y
	auxid	*varchar2(4)*	Auxiliary identification code	Y
I	time	*float(53)*	Start time of outage interval	Y
	endtime	*float(53)*	End time of outage interval	Y
	status	*varchar2(32)*	Status of this row.	Y
	auth	*varchar2(15)*	Source/originator of outage report	Y
	available	*varchar2(1)*	Flag to specify if data are unavailable (f) or available (t)	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_OUTAGE(otgid)
Indexes	PK_OUTAGE(otgid), OTGTIMEX(time), OTGSTAX(sta, chan, time)

3.1.97. OVERLAYDISC

Owner(s)	MAP
Category	Interactive Processing
Description	The OVERLAYDISC table contains the location of the overlays for the Map application.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	overlayid	*number(8,0)*	Overlay identifier	N
	overlayname	*varchar2(64)*	Overlay name	N
	dfile	*varchar2(32)*	Name of overlay file	N
	dir	*varchar2(64)*	Absolute path to the data file	N
	colorname	*varchar2(32)*	Overlay color name	N
	lddate	*date*	Load date	Y

Primary key	OVERLAYDISC_PK(overlayid)
Indexes	OVERLAYDISC_PK(overlayid)

3.1.98. PARRIVAL

Owner(s)	LEB
Category	Fundamental
Description	The PARRIVAL table contains the predicted arrivals and associations for origin-based amplitude measurements.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	parid	*number(10,0)*	Predicted arrival identifier	N
I	orid	*number(10,0)*	Origin identifier	Y
I	evid	*number(10,0)*	Event identifier	Y
	sta	*varchar2(6)*	Station code	Y
	time	*float(53)*	Epoch time	Y
	azimuth	*float(24)*	Azimuth	Y
	slow	*float(24)*	Predicted slowness for an arrival (s/deg)	Y
	phase	*varchar2(8)*	Phase type	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	vmodel	*varchar2(15)*	Velocity model	Y
I	lddate	*date*	Load date	Y

Primary key	PK_PARRIVAL(parid)
Indexes	PARRORIDX(orid), PARREVIDX(evid), PARRIVAL_LDDATEX(lddate), PK_PARRIVAL(parid)

3.1.99. PARRIVAL

Owner(s)	SEL2, SEL3, SEL1
Category	Fundamental
Description	The PARRIVAL table contains the predicted arrivals and associations for origin-based amplitude measurements.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	parid	*number(10,0)*	Predicted arrival identifier	N
I	orid	*number(10,0)*	Origin identifier	Y
	evid	*number(10,0)*	Event identifier	Y
	sta	*varchar2(6)*	Station code	Y
	time	*float(53)*	Epoch time	Y
	azimuth	*float(24)*	Azimuth	Y
	slow	*float(24)*	Predicted slowness for an arrival (s/deg)	Y
	phase	*varchar2(8)*	Phase type	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y

	vmodel	*varchar2(15)*	Velocity model	Y
	lddate	*date*	Load date	Y

Indexes	PARRORIDX(orid), PARRIVALX(parid)

3.1.100. PARRIVAL

Owner(s)	REB
Category	Fundamental
Description	The PARRIVAL table contains the predicted arrivals and associations for origin-based amplitude measurements.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	parid	*number(10,0)*	Predicted arrival identifier	N
I	orid	*number(10,0)*	Origin identifier	Y
I	evid	*number(10,0)*	Event identifier	Y
	sta	*varchar2(6)*	Station code	Y
	time	*float(53)*	Epoch time	Y
	azimuth	*float(24)*	Azimuth	Y
	slow	*float(24)*	Predicted slowness for an arrival (s/deg)	Y
	phase	*varchar2(8)*	Phase type	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	vmodel	*varchar2(15)*	Velocity model	Y
I	lddate	*date*	Load date	Y

Primary key	PK_PARRIVAL(parid)
Indexes	PAREVIDX(evid), PARRORIDX(orid), PARRIVAL_LDDATEX(lddate), PK_PARRIVAL(parid)

3.1.101. PARTICIPATION

Owner(s)	STATIC
Category	Reference
Description	The PARTICIPATION table contains station participation information for performance reporting.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	net	*varchar2(8)*	Network identification string	N
PI	sta	*varchar2(6)*	Station code	N
PI	begin_date	*number(8,0)*	Date participation began	N
	end_date	*number(8,0)*	Date participation ended	Y
	lddate	*date*	Load date	Y

Primary key	PK_PARTICIPATION(net, sta, begin_date)

3.1.102. PHASE_DESCRIPTION

Owner(s)	STATIC
Category	Interactive Analysis
Description	Distance and depth information for phases used in Interactive SHI analysis

Pk/Fk/I	Column	Storage type	Description	Nullable
	phase	*varchar2(8)*	Phase type	N
	mindelta	*float(24)*	Minimum distance (degrees)	N
	maxdelta	*float(24)*	Maximum allowable distance (degrees)	N
	mindepth	*float(24)*	Minimum allowable depth (km)	N
	maxdepth	*float(24)*	Maximum allowable depth (km)	N
	phase_type	*varchar2(24)*	Phase type	N
	prop_mode	*varchar2(1)*	Propagation mode (one of P,S,H,I,V or N)	N
	is_selected	*varchar2(1)*	Flag indicating if the phase can be selected by the analyst	N
	is_unique	*varchar2(1)*	Flag indicating whether or not the phase name is unique	N
	is_timeavail	*varchar2(1)*	Flag indicating whether or not travel time information is available for this phase	N
	is_depth_phase	*varchar2(1)*	Flag indicating whether or not the phase is a depth phase	N
	is_depth_sensitive	*varchar2(1)*	Flag indicating whether or not the phase is depth sensitive	N
	is_coda	*varchar2(1)*	Flag indicating whether or not the phase is a coda phase	N
	lddate	*date*	Load date	N

3.1.103. PHASE_WEIGHT_EVENT_DEFINE

Owner(s)	STATIC
Category	Interactive Analysis
Description	Time, azimuth and slowness weights used in the Event Definition Criteria

Pk/Fk/I	Column	Storage type	Description	Nullable
	phase_type	*varchar2(24)*	Phase type	N
	obs_type	*varchar2(24)*	Observation type	N
	time_weight	*float(24)*	Weight given to defining time for this phase	N
	az_weight	*float(24)*	Weight given to defining azimuth for this	N

			phase type.	
	slow_weight	*float(24)*	Weight given to defining slowness for this phase	N
	num_defining	*number(8,0)*	Number of defining phases	N
	lddate	*date*	Load date	N

3.1.104. PHASOR_DIST_DEPTH_RANGES

Owner(s)	STATIC
Category	Interactive Analysis
Description	Table containing distance and depth ranges for different phases used by the Phasor application.

Pk/Fk/I	Column	Storage type	Description	Nullable
	phase	*varchar2(8)*	Phase type	Y
	min_dist	*float(24)*	Minimum allowable distance (degrees)	Y
	max_dist	*float(24)*	Maximum allowable distance (degrees)	Y
	min_depth	*float(24)*	Minimum allowable depth (km)	Y
	max_depth	*float(24)*	Maximum allowable depth (km)	Y
	mag_cutoff	*float(24)*	Lower bound of body wave magnitude for a phase to be considered for processing by Phasor	Y
	travel_time_error	*float(24)*	travel time error	Y
	lddate	*date*	Load date	N

3.1.105. PMCC_FEATURES

Owner(s)	IDCX
Category	Automatic Processing
Description	Features of infrasonic arrivals, populated by DFX-PMCC

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	arid	*number(10,0)*	Arrival identifier	N
	initial_time	*float(53)*	Initial PMCC detection time (epochal)	Y
	duration	*float(24)*	Duration in seconds	Y
	consistency	*float(24)*	Consistency value	Y
	correlation	*float(24)*	Correlation coefficient	Y
	famsize	*number(8,0)*	PMCC family size	Y
	minfreq	*float(24)*	Minimum frequency (Hz)	Y

	maxfreq	*float(24)*	Maximum frequency (Hz)	Y
	cfreq	*float(24)*	Central frequency (Hz)	Y
	sigmafreq	*float(24)*	Standard deviation of frequency values	Y
	rmsamp	*float(24)*	Root mean square amplitude	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_PARTICIPATION(sta, arid)
Indexes	PK_PARTICIPATION(sta, arid), PMCC_FEAT_ARIX(arid)

3.1.106. PMCC_RECIPE

Owner(s)	IDCX
Category	Automatic Processing
Description	Processing recipes used for infrasound processing by DFX-PMCC

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	pmccrecid	*number(10,0)*	unique pmcc recipe identifier (integer)	N
	deflt	*number(8,0)*	flag indicating whether parameter takes default value	Y
	fgroup	*number(8,0)*	frequency group	Y
	winlen	*float(24)*	window length (seconds)	Y
	wingap	*float(24)*	window gap (seconds)	Y
	threshcons	*float(24)*	threshold consistency	Y
	threshnsens	*number(8,0)*	minimum number of sensors	Y
	qfactor	*float(24)*	quality factor	Y
	pmcc3d	*number(8,0)*	flag indicating whether three-dimensional mode is used	Y
	sound_speed	*float(24)*	sound speed (km/s)	Y
	elevation_angle	*float(24)*	elevation angle	Y
	threshfamlen	*float(24)*	threshold family length	Y
	threshfammin	*number(8,0)*	minimum family size threshold	Y
	threshfammax	*number(8,0)*	maximum family size threshold	Y
	speed_transition	*float(24)*	speed transition	Y
	time_tol	*float(24)*	time tolerance (seconds)	Y
	freq_tol	*float(24)*	frequency tolerance	Y
	sp_tol1	*float(24)*	minimum speed tolerance	Y
	sp_tol2	*float(24)*	maximum speed tolerance	Y

	az_tol1	*float(24)*	minimum azimuth tolerance	Y
	az_tol2	*float(24)*	maximum azimuth tolerance	Y
	auth	*varchar2(17)*	name of the application or user making the update	Y
	lddate	*date*	Load date	Y

Primary key	PMCC_RECIPE_PK(pmccrecid)
Indexes	PMCC_RECIPE_PK(pmccrecid)

3.1.107. PROBLASTID (VIEW)

Owner(s)	IDCX
Category	System Monitoring
Description	View of the LASTID table, used by System Monitoring software (Xlogger)

Pk/Fk/I	Column	Storage type	Description	Nullable
	keyname	*varchar2(15)*	Identifier name (arid, orid, and so on)	N
	keyvalue	*number(10,0)*	Last value used for the identifier specified in the keyname column	N
	lddate	*date*	Load date	Y

SQL Body	`SELECT keyname, keyvalue, lddate`
	`FROM lastid`
	`where keyname in ( 'prid', 'plid', 'tmp$object' )`

3.1.108. PROBLEM

Owner(s)	IDCX
Category	System Monitoring
Description	The PROBLEM table contains information related to problems with stations, communications, software, and hardware.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prid	*number(10,0)*	Problem identifier	N
	procclass	*varchar2(17)*	Xlogger problem category	Y
	procname	*varchar2(17)*	Xlogger problem name	Y
	time	*number(17,5)*	Epoch time of start of problem	Y
	endtime	*number(17,5)*	Epoch time of end of problem	Y
	status	*varchar2(33)*	Status of the problem	Y
	class	*varchar2(33)*	Class of problem	Y
	descrip	*varchar2(129)*	One line description of problem	Y
	login	*varchar2(33)*	Login name of originator	Y

	lddate	*date*	Load date	Y

Primary key	PK_PROBLEM(prid)
Indexes	PK_PROBLEM(prid)

3.1.109. PROBLEMLOG

Owner(s)	IDCX
Category	System Monitoring
Description	The PROBLEMLOG table contains individual problem log entries relating to problems in the database table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	plid	*number(10,0)*	Problem log identifier	N
	prid	*number(10,0)*	Problem identifier	Y
	time	*number(17,5)*	Time of log entry	Y
	dir	*varchar2(129)*	Absolute path to the data file	Y
	dfile	*varchar2(65)*	Filename of log file	Y
	login	*varchar2(33)*	Login name of originator	Y
	lddate	*date*	Load date	Y

Primary key	PK_PROBLEMLOG(plid)
Indexes	PK_PROBLEMLOG(plid)

3.1.110. PROBLEMMAIL

Owner(s)	IDCX
Category	System Monitoring
Description	The PROBLEMMAIL table contains instructions for distributing problem messages to subscribers.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	procclass	*varchar2(17)*	Xlogger problem category	N
PI	login	*varchar2(33)*	Login name of originator	N
	lddate	*date*	Load date	Y

Primary key	PK_PROBLEMMAIL(procclass, login)
Indexes	PK_PROBLEMMAIL(procclass, login)

3.1.111. PRODTRACK

Owner(s)	IDCX
Category	Data Services

Description	The PRODTRACK table queues and links data and subscription products. The dataid column links the product delivery to the data encapsulated in the DATAREADY table, and the msgid column links the product to the outgoing message, as referenced in the MSGDISC table. Thus, the PRODTRACK table allows for subscription tracking from the data to the product.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prodid	*number(10,0)*	Product identifier	N
	delivid	*number(10,0)*	Delivery identifier	Y
PI	dataid	*number(10,0)*	Data ready identifier	N
PI	msgid	*number(10,0)*	Message identifier	N
I	status	*varchar2(12)*	Status of the product	Y
I	lddate	*date*	Load date	Y

Primary key	PRODTRACK_PK(prodid, dataid, msgid)
Indexes	INMSGIDX(msgid), INSTATUSX(status), PRODTRACK_PK(prodid, dataid, msgid), PRODTRACK_LDDATEX(lddate), INPRODIDX(prodid)

3.1.112. PRODUCTCRITERIA

Owner(s)	IDCX
Category	Data Services
Description	The PRODUCTCRITERIA table contains subscription information about the type of product and when and how often to distribute the data. Each unique product has one record.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	prodid	*number(8,0)*	Product identifier	Y
	prodname	*varchar2(24)*	Product name (if a standard product)	Y
	delivid	*number(8,0)*	Last delivery for this product	Y
	cycle_size	*number(8,0)*	Product size to be reached before release (0 for no limit)	Y
	cycle_time	*number(4,0)*	Time interval between releases (0 for immediate)	Y
	hour_to_send	*number(2,0)*	Hour to send product	Y
	dow_to_send	*number(2,0)*	Day of week to send subscription	Y
	dom_to_send	*number(2,0)*	Day of month to send subscription	Y
	prodtype	*varchar2(32)*	Product type	Y
	prodsubtype	*varchar2(12)*	Product sub-type	Y
	prodfmt	*varchar2(6)*	Product format	Y
	supressempty	*char(2)*	Flag indicating whether or not to suppress sending of empty products	Y
	timelastsend	*float(54)*	Time at which product was last sent	Y

			(epochal)	
	timenextsend	*float(54)*	Time at which product will next be sent (epochal)	Y
	lddate	*date*	Load date	Y

3.1.113. PRODUCTTYPEEVPARS

Owner(s)	IDCX
Description	Obsolete.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prodid	*number(8,0)*	Obsolete.	N
	ec_grpname	*varchar2(24)*	Obsolete.	Y
	ec_regname	*varchar2(24)*	Obsolete.	Y
	ec_depthco	*float(24)*	Obsolete.	Y
	ec_depthp	*float(24)*	Obsolete.	Y
	ec_mbmsco	*float(24)*	Obsolete.	Y
	ec_mbmsns	*number(8,0)*	Obsolete.	Y
	ec_mbmsp	*float(24)*	Obsolete.	Y
	ec_mbmscal	*float(24)*	Obsolete.	Y
	ec_mbmsfac	*float(24)*	Obsolete.	Y
	ec_onshorep	*float(24)*	Obsolete.	Y
	ec_ldpflg	*varchar2(1)*	Obsolete.	Y
	ec_ndef	*number(8,0)*	Obsolete.	Y
	ec_udat	*varchar2(1)*	Obsolete.	Y
	ec_popcrit	*varchar2(60)*	Obsolete.	Y
	ec_popmagc	*float(24)*	Obsolete.	Y
	ec_popmint	*number(8,0)*	Obsolete.	Y
	ec_popsnco	*float(24)*	Obsolete.	Y
	ec_popsimp	*float(24)*	Obsolete.	Y

Primary key	PK_PRODUCTTYPEEVPARS(prodid)

3.1.114. PRODUCTTYPEEVSC

Owner(s)	IDCX
Category	Data Services
Description	The PRODUCTTYPEEVSC table contains the input parameters that define the event-screening criteria for the standard case and national subscriptions. It is used in conjunction with the PRODUCTTYPEORIGIN table to specify the set of user input criteria. There is one record for each subscription (prodid).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prodid	*number(10,0)*	Product identifier	N
	grpname	*varchar2(24)*	Subscription group name	Y
	regname	*varchar2(24)*	Region name	Y
	min_ndef	*number(8,0)*	Minimum number of defining phases	Y
	min_mb	*float(24)*	Minimum mb for screening	Y
	depth_conf	*float(24)*	Depth screening confidence	Y
	depth_thresh	*float(24)*	Depth screening threshold	Y
	depth_kvalue	*float(24)*	Depth uncertainty k-value	Y
	min_ndp_pp	*number(8,0)*	Minimum required number of pP depth phases	Y
	min_ndp_sp	*number(8,0)*	Minimum required number of sP depth phases	Y
	min_moveout_pp	*float(24)*	Minimum moveout of pP-P travel time differences	Y
	min_moveout_sp	*float(24)*	Minimum moveout of sP-P travel time differences	Y
	min_dp_snr_pp	*float(24)*	Minimum pP depth phase snr	Y
	min_dp_snr_sp	*float(24)*	Minimum sP depth phase snr	Y
	mbms_conf	*float(24)*	Mb minus Ms screening confidence	Y
	mbms_thresh	*float(24)*	Mb minus Ms screening threshold	Y
	mbms_slope	*float(24)*	Slope term for mb minus Ms relation	Y
	magpref_mb	*varchar2(6)*	Mb magnitude type used for screening.	Y
	magpref_ms	*varchar2(6)*	Ms magnitude type used for screening.	Y
	mb_err	*float(24)*	Single-station mb uncertainty	Y
	ms_err	*float(24)*	Single-station Ms uncertainty	Y
	min_nsta_ms	*number(8,0)*	Minimum number of stations for ms	Y
	loc_conf	*float(24)*	Location error ellipse confidence	Y
	hydro_te_thresh	*float(24)*	Hydroacoustic total energy threshold	Y
	hydro_cp_thresh	*float(24)*	Hydroacoustic cepstral peak threshold	Y
	min_wdepth_thresh	*float(24)*	Minimum water depth threshold	Y
	reg_conf	*float(24)*	Regional screening confidence	Y
	lddate	*date*	Load date	Y

Primary key	PK_PRODUCTTYPEEVSC(prodid)

3.1.115. PRODUCTTYPEORIGIN

Owner(s)	IDCX

Category	Data Services
Description	The PRODUCTTYPEORIGIN table contains the details of origin subscriptions. It extends the PRODUCTCRITERIA table that is generic to all subscriptions. Each origin-based product has one record.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prodid	*number(8,0)*	Product identifier	N
	minlat	*float(24)*	Minimum latitude of event	Y
	maxlat	*float(24)*	Maximum latitude of event	Y
	minlon	*float(24)*	Minimum longitude of event	Y
	maxlon	*float(24)*	Maximum longitude of event	Y
	mindepth	*float(24)*	Minimum depth value for a product constraint (km).	Y
	maxdepth	*float(24)*	Maximum allowable depth (km)	Y
	minmag	*float(24)*	Minimum magnitude	Y
	maxmag	*float(24)*	Maximum magnitude	Y
	magtype	*varchar2(4)*	Magnitude type, for example mb, Ms, ML	Y
	minmb_ms	*float(24)*	Minimum value of mb minus Ms	Y
	maxmb_ms	*float(24)*	Maximum value of mb minus Ms	Y
	mindep_err	*float(24)*	Minimum value of depth minus error	Y
	maxdep_err	*float(24)*	Maximum value of depth minus error	Y
	minesd	*float(24)*	Minimum event-station distance	Y
	maxesd	*float(24)*	Maximum event-station distance	Y

Primary key	PK_PRODUCTTYPEORIGIN(prodid)

3.1.116. PRODUCTTYPESTA

Owner(s)	IDCX
Category	Data Services
Description	The PRODUCTTYPESTA table contains the criteria used for defining subscription products with constraints that are based on the station. If a user requests a subscription for multiple stations, the table will have one PRODUCTCRITERIA row, but multiple PRODUCTTYPESTA rows (one for each station); all of the PRODUCTTYPESTA rows will have a common value of prodid.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	prodid	*number(8,0)*	Product identifier	N
	minlat	*float(24)*	Minimum station latitude	Y
	maxlat	*float(24)*	Maximum station latitude	Y
	minlon	*float(24)*	Minimum station longitude	Y
	maxlon	*float(24)*	Maximum station longitude	Y

PI	sta	*varchar2(6)*	Station code	N
PI	chan	*varchar2(8)*	Channel name	N

Primary key	PK_PRODUCTTYPESTA(prodid, sta, chan)
Indexes	PK_PRODUCTTYPESTA(prodid, sta, chan)

3.1.117. QCSTATS

Owner(s)	IDCX
Category	System Monitoring
Description	The QCSTATS table contains waveform data quality statistics.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	qcstatsid	*number(10,0)*	Data quality statistics identifier	N
PI	sta	*varchar2(6)*	Station code	N
I	chan	*varchar2(8)*	Channel name	N
I	time	*float(53)*	Interval start time	N
	jdate	*number(8,0)*	Julian date for this data	Y
I	endtime	*float(53)*	Interval end time	Y
	dettime	*float(53)*	Detection interval start time	Y
	detendtime	*float(53)*	Detection interval end time	Y
	missing	*float(53)*	Amount of missing data (seconds)	Y
	dropped	*number(8,0)*	Flag indicating if interval was dropped	Y
	nseg	*number(8,0)*	Number of masked segments	Y
	masked	*float(53)*	Amount of data in masked segments	Y
	pointspike	*float(53)*	Amount of data in masked due to point-spikes	Y
	spike	*float(53)*	Amount of data in masked due to spikes	Y
	nconstseg	*number(8,0)*	Number of constant valued segments	Y
	const	*float(53)*	Amount of data masked due to constant valued segments	Y
	avgconstval	*float(53)*	Average value in constant segments	Y
	stdconstval	*float(53)*	Standard deviation of constant values	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
I	lddate	*date*	Load date	Y

Primary key	QCSTATS_PK(qcstatsid)
Indexes	QCSTATSIDX(qcstatsid), QCSTATS_ENDSTCHNX(endtime, sta, chan), QCSTATS_LDDATEX(lddate), QCSTATS_TIMEX(time)

3.1.118. REBDONE_DATADAY_FLAG

Owner(s)	LEB
Category	Interactive Processing
Description	Not used at the IDC.

Pk/Fk/I	Column	Storage type	Description	Nullable
	dataday	*number(12,0)*	Julian day	Y
	state	*varchar2(12)*	state of processing	Y
	donetime	*float(53)*	time that REB was produced	Y
	lddate	*date*	load date	Y

3.1.119. REGCOEF

Owner(s)	STATIC
Category	Fundamental
Description	The REGCOEF table contains linear coefficients for regional processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	rcoefid	*varchar2(20)*	Linear coefficient set identifier	N
PI	rcoeftype	*varchar2(10)*	Linear coefficient value identifier	N
	rcoefvalue	*float(24)*	Linear coefficient value	Y
	ondate	*number(8,0)*	First julian date this row is valid	Y
	offdate	*number(8,0)*	Last Julian date this row is valid	Y
	lddate	*date*	Load date	Y

Primary key	SYS_C0028355(rcoefid, rcoeftype)
Indexes	PK_REGCOEF(rcoefid, rcoeftype)

3.1.120. REMARK

Owner(s)	IDCX, STATIC, SEL3, SEL1, SEL2
Category	Database and Utility
Description	Obsolete. It was used in the past to store tables and column comments.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	commid	*number(10,0)*	Comment identifier	N
PI	commid	*number(10,0)*	Comment identifier	N
I	lineno	*number(8,0)*	Comment line number	N
PI	lineno	*number(8,0)*	Comment line number	N
	remark	*varchar2(80)*	Free-format comment	Y

	lddate	*date*	Load date	Y

Primary key	PK_REMARK(commid, lineno)
Indexes	REMARKX(commid, lineno)

3.1.121. REMARK

Owner(s)	LEB
Category	Database and Utility
Description	Obsolete. It was used in the past to store tables and column comments.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	commid	*number(10,0)*	Comment identifier	N
PI	lineno	*number(8,0)*	Comment line number	N
	remark	*varchar2(80)*	Free-format comment	Y
I	lddate	*date*	Load date	Y

Primary key	PK_REMARK(commid, lineno)
Indexes	REMARK_LDDATEX(lddate), PK_REMARK(commid, lineno)

3.1.122. REMARK

Owner(s)	REB
Category	Database and Utility
Description	Obsolete. It was used in the past to store tables and column comments.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	commid	*number(10,0)*	Comment identifier	N
PI	lineno	*number(8,0)*	Comment line number	N
	remark	*varchar2(80)*	Free-format comment	Y
	lddate	*date*	Load date	Y

Primary key	PK_REMARK(commid, lineno)
Indexes	PK_REMARK(commid, lineno)

3.1.123. REQUEST

Owner(s)	IDCX
Category	Data Services
Description	The REQUEST table defines segments of auxiliary waveform data to be acquired. The start_time, end_time, sta and chan fields define a single unit of data. Data import programs must succeed in acquiring all the data for a time interval before changing the state to indicate successful retrieval, otherwise the state will be updated to indicate

	whether data were partially retrieved or no data was retrieved.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	reqid	*number(10,0)*	Request identifier	N
	sta	*varchar2(6)*	Station code	N
	chan	*varchar2(8)*	Channel name	N
	array	*varchar2(8)*	Array code	Y
	orid	*number(10,0)*	Origin identifier	Y
	evid	*number(10,0)*	Event identifier	Y
I	start_time	*float(53)*	Starting time of requested waveform data	N
I	end_time	*float(53)*	Ending time of requested waveform data	N
	class	*varchar2(16)*	Type of request	Y
I	state	*varchar2(16)*	Current processing state	Y
	statecount	*number(8,0)*	Number of failed attempts (when state = failed)	Y
	complete	*number(8,0)*	Percentage of data acquired	Y
	requestor	*varchar2(15)*	Original author of record	Y
I	modtime	*float(53)*	Time of last state change (epoch time)	Y
	modauthor	*varchar2(15)*	Author of last state change	Y
I	lddate	*date*	Load date	Y

Primary key	PK_REQUEST(reqid)
Indexes	REQMODX(modtime), REQSTATEX(state), REQSTENDX(start_time, end_time), REQUEST_LDDATEX(lddate), PK_REQUEST(reqid), REQENDX(end_time)

3.1.124. REVAUDIT

Owner(s)	LEB
Category	Interactive Analysis
Description	Provides an audit trail of actions taken by SHI Lead Analysts running rebrevise during the course of LEB review.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	revid	*number(10,0)*	Revision identifier	N
	revfunction	*varchar2(32)*	Name of the revision function used	N
	auth	*varchar2(15)*	Name of the Analyst making the revision	N
I	revtagid1	*number(10,0)*	Revtagname1 value; the value of the foreign key identified in revtagname1	N
I	revtagname1	*varchar2(8)*	Revtagname1 type; the name of the foreign key whose value is revtagid1	N
	revtagid2	*number(10,0)*	Revtagname2 value; the value of the foreign	Y

			key identified in revtagname2	
	revtagname2	*varchar2(8)*	Revtagname2 type; the name of the foreign key whose value is revtagid2	Y
	revstate	*varchar2(16)*	State of the revision	Y
	lddate	*date*	Load date	Y

Primary key	PK_REVAUDIT(revid)
Indexes	REVTAGNDX(revtagid1, revtagname1), PK_REVAUDIT(revid)

3.1.125. RMS_LASTID (VIEW)

Owner(s)	IDCX
Category	Automatic Processing
Description	View of the LASTID table, used by radionuclide processing software

Pk/Fk/I	Column	Storage type	Description	Nullable
	keyname	*varchar2(15)*	Identifier name (arid, orid, and so on)	N
	keyvalue	*number(10,0)*	Last value used for the identifier specified in the keyname column	N
	lddate	*date*	Load date	Y

SQL Body	`SELECT keyname, keyvalue, lddate`
	`FROM lastid`
	`where keyname = 'fpid'`

3.1.126. RULESET

Owner(s)	STATIC
Category	Automatic Processing
Description	The RULESET table defines rule sets to fine tune cdquals Minimum Channel Mission Capability calculation. It is used by cdqual, see cdtools user guide for details.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	ondate	*number(8,0)*	First julian date this row is valid	N
I	offdate	*number(8,0)*	Last Julian date this row is valid	Y
I	name	*varchar2(32)*	Rule name	Y
PI	regexp	*varchar2(64)*	Regular expression	N
	minchan	*varchar2(8)*	Minimum number of channels	Y
	lddate	*date*	Load date	Y

Primary key	RULESET_PK(sta, ondate, regexp)
Indexes	RULESET_PK(sta, ondate, regexp), RULESETSN(sta, offdate, name)

3.1.127. SEISGRID

Owner(s)	STATIC
Category	Automatic Processing
Description	The SEISGRID table contains a natural seismicity grid, which includes the average number of events per year with magnitude greater than the threshold in this table for each latitude-longitude grid point (the grid points are defined in the SEISINDEX table). The SEISGRID table is used by the Anomalous Event Qualifier (AEQ) application to help identify anomalous events.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	grdname	*varchar2(6)*	Grid name	N
PI	icell	*number(8,0)*	Grid cell index	N
	magth	*float(24)*	Magnitude threshold	Y
	magtype	*varchar2(6)*	Magnitude type, for example mb, Ms, ML	Y
	nevyr	*float(24)*	Average number of events/year	Y
	lddate	*date*	Load date	Y

Primary key	PK_SEISGRID(grdname, icell)
Indexes	SEISGRIDX(icell), PK_SEISGRID(grdname, icell)

3.1.128. SEISINDEX

Owner(s)	STATIC
Category	Automatic Processing
Description	The SEISINDEX table contains the geographic grids of natural seismicity data in the SEISGRID table. The DSEISINDEX table contains the geographic grids of seismicity data in the DSEISGRID table. The SEISINDEX table is used by the Anomalous Event Qualifier (AEQ) application to help identify anomalous events.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	grdname	*varchar2(6)*	Grid name	N
	lat1	*float(24)*	Initial latitude	Y
	lon1	*float(24)*	Initial longitude	Y
	dlat	*float(24)*	Latitude increment	Y
	dlon	*float(24)*	Longitude increment	Y
	nlat	*number(8,0)*	Number of latitudes	Y
	nlon	*number(8,0)*	Number of longitudes	Y
	orderby	*varchar2(6)*	Order by (either latitude or longitude)	Y
	lddate	*date*	Load date	Y

Primary key	PK_SEISINDEX(grdname)
Indexes	PK_SEISINDEX(grdname)

3.1.129. SEMAPHORE

Owner(s)	IDCX
Category	Data Services
Description	Table used by the Message System to enforce locking of logfiles to prevent race conditions while rotating logfiles.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
	pid	*number(10,0)*	Process id	Y
	lddate	*date*	Load date	Y

3.1.130. SENSOR

Owner(s)	STATIC
Category	Core, Reference
Description	The SENSOR table contains calibration information for specific sensor channels. This table provides a record of updates in the calibration factor or clock error of each instrument and links a sta/chan/time to a complete instrument response in the table instrument. Waveform data are converted into physical units through multiplication by the calib field located in WFDISC. The correct value of calib may not be accurately known when the wfdisc record is entered into the database. The sensor table provides the mechanism (calratio and calper) to "update" calib, without requiring possibly hundreds of wfdisc records to be updated. Through the foreign key inid, this table is linked to instrument, which has fields pointing to flat files holding detailed calibration information in a variety of formats (see instrument).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	chan	*varchar2(8)*	Channel name	N
PI	time	*float(53)*	Epoch time of start of recording period	N
PI	endtime	*float(53)*	Epoch time of end of recording period	N
	inid	*number(8,0)*	Instrument identifier	N
	chanid	*number(8,0)*	Channel recording identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	calratio	*float(24)*	Calibration	Y
	calper	*float(24)*	Calibration period (seconds)	Y
	tshift	*float(24)*	Correction of data processing time	Y
	instant	*varchar2(1)*	(y, n) discrete/continuing snapshot	N
	lddate	*date*	Load date	Y

Primary key	PK_SENSOR(sta, chan, time, endtime)
Indexes	SENSSTAX(sta, chan), PK_SENSOR(sta, chan, time, endtime)

3.1.131. SITE

Owner(s)	STATIC
Category	Core, Reference
Description	The SITE table contains station location information. Site names and describes a point on the earth where measurements are made (for example, the location of an instrument or array of instruments). It contains information that normally changes infrequently, such as location. In addition, site contains fields that describe the offset of a station relative to an array reference location. Global data integrity implies that the sta/ondate in site be consistent with the sta/chan/ondate in SITECHAN.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	ondate	*number(8,0)*	First julian date this row is valid	N
	offdate	*number(8,0)*	Last Julian date this row is valid	Y
	lat	*number*	Geographic latitude. Locations north of the equator have positive latitudes.	Y
	lon	*number*	Geographic longitude. Longitudes are measured positive east of the Greenwich meridian.	Y
	elev	*number*	Elevation	Y
	staname	*varchar2(50)*	Station description	Y
	statype	*varchar2(4)*	Station type: single station, array	Y
	refsta	*varchar2(6)*	Reference station for array members	Y
	dnorth	*number*	Offset from array reference (km)	Y
	deast	*number*	Offset from array reference (km)	Y
	lddate	*date*	Load date	Y

Primary key	PK_SITE(sta, ondate)
Indexes	SITESTAX(sta), PK_SITE(sta, ondate)

3.1.132. SITEAUX

Owner(s)	STATIC
Category	Reference
Description	The SITEAUX table contains additional site-dependent parameters that are not included in the SITE table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	chan	*varchar2(8)*	Channel name	N
PI	time	*float(53)*	Epoch time	N
	nois	*float(24)*	Noise amplitude	Y

	noissd	*float(24)*	Standard deviation of log noise	Y
	amcor	*float(24)*	Amplitude correction	Y
	amcorsd	*float(24)*	Correction standard deviation	Y
	snthrsh	*float(24)*	Signal/noise detection threshold	Y
	rely	*float(24)*	Station reliability	Y
	ptmcor	*float(24)*	P arrival time correction	Y
	stmcor	*float(24)*	S arrival time correction	Y
	staper	*float(24)*	Period for measurements	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_SITEAUX(sta, chan, time)
Indexes	PK_SITEAUX(sta, chan, time)

3.1.133. SITECHAN

Owner(s)	STATIC
Category	Core, Reference
Description	The SITECHAN table contains station-channel information. This table describes the orientation of a recording channel at the site referenced by sta. It provides information about the various channels that are available at a station and maintains a record of the physical channel configuration at a site.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	chan	*varchar2(8)*	Channel identifier	N
PI	ondate	*number(8,0)*	First julian date this row is valid	N
I	chanid	*number(8,0)*	Channel recording identifier	Y
	offdate	*number(8,0)*	Last Julian date this row is valid	Y
	ctype	*varchar2(4)*	Channel type	Y
	edepth	*number*	Emplacement depth	Y
	hang	*number*	Horizontal angle	Y
	vang	*number*	Vertical angle	Y
	descrip	*varchar2(50)*	Channel description	Y
	lddate	*date*	Load date	Y

Primary key	PK_SITECHAN(sta, chan, ondate)
Indexes	SITECHANX(chanid), STCSTAX(sta, chan), PK_SITECHAN(sta, chan, ondate)

3.1.134. SITEPOLL

Owner(s)	STATIC
Category	Data Services
Description	The SITEPOLL table contains the station and channel names of auxiliary seismic stations that are polled periodically for system monitoring purposes.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
PI	net	*varchar2(6)*	Network identification string	N
PI	chan	*varchar2(6)*	Channel name	N
	lddate	*date*	Load date	Y

Primary key	PK_SITEPOLL(sta, net, chan)
Indexes	PK_SITEPOLL(sta, net, chan)

3.1.135. SITEPOLLCA

Owner(s)	STATIC
Description	Obsolete.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
	net	*varchar2(6)*	Network identification string	Y
	chan	*varchar2(6)*	Channel name	Y
	lddate	*date*	Load date	Y

3.1.136. SITE_ADDRESS

Owner(s)	STATIC
Category	Reference
Description	The SITE_ADDRESS table contains address information not included in the SITE table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sta	*varchar2(6)*	Station code	N
	name	*varchar2(20)*	Expanded station name	Y
	locality	*varchar2(40)*	Locality of station	Y
	stat_prov	*varchar2(40)*	State or province	Y
	country	*varchar2(40)*	Country name	Y
	lddate	*date*	Load date	Y

Primary key	PK_SITE_ADDRESS(sta)
Indexes	PK_SITE_ADDRESS(sta)

3.1.137. SPLP

Owner(s)	REB
Category	Fundamental
Description	The SPLP table contains event characterization parameters for short-period/long-period energy ratios. The ratios are computed using the DFX application.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
PI	sta	*varchar2(6)*	Station code	N
	rectype	*varchar2(8)*	Recipe type	N
	ratio	*float(24)*	Ratio of short-period to long-period energy	Y
I	lddate	*date*	Load date	Y

Primary key	PK_SPLP(orid, sta)
Indexes	SPLP_LDDATEX(lddate), PK_SPLP(orid, sta), SPLPORIDX(orid)

3.1.138. SPVAR

Owner(s)	REB
Category	Fundamental
Description	The SPVAR table contains the variance of the detrended log spectrum between fmin and fmax for an arrival identified by arid. The frequency bandwidth is based on a signal-to-noise ratio criterion. Each phase associated with an event has a SPVAR record.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
	fsid	*number(10,0)*	Fourier spectrum identifier	N
	acoef	*float(24)*	"a" coefficient for nonlinear trend	Y
	bcoef	*float(24)*	"b" coefficient for nonlinear trend	Y
	ccoef	*float(24)*	"c" coefficient for nonlinear trend	Y
PI	fmin	*float(24)*	Minimum frequency	N
PI	fmax	*float(24)*	Maximum frequency	N
	svar	*float(24)*	Variance of detrended log spectrum	Y
I	lddate	*date*	Load date	Y

Primary key	PK_SPVAR(arid, fmin, fmax)
Indexes	PK_SPVAR(arid, fmin, fmax), SPVAR_LDDATEX(lddate)

3.1.139. SREGION

Owner(s)	STATIC
Category	Core, Reference
Description	The SREGION table contains seismic region numbers and their equivalent descriptions (see [Fli74]).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	srn	*number(8,0)*	Seismic region number, as given in [Fli74]	N
	srname	*varchar2(40)*	Seismic region name	Y
	lddate	*date*	Load date	Y

Primary key	PK_SREGION(srn)
Indexes	PK_SREGION(srn)

3.1.140. STAMAG

Owner(s)	LEB
Category	Core, Fundamental
Description	The STAMAG table contains station magnitude estimates based upon measurements made on specific seismic phases. Values in STAMAG are used to calculate network magnitudes stored in NETMAG.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	magid	*number(10,0)*	Station magnitude identifier	N
PI	ampid	*number(10,0)*	Amplitude identifier	N
PI	sta	*varchar2(6)*	Station code	N
I	arid	*number(10,0)*	Arrival identifier	Y
I	orid	*number(10,0)*	Origin identifier	Y
	evid	*number(10,0)*	Event identifier	Y
	phase	*varchar2(8)*	Phase type	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	magtype	*varchar2(6)*	Magnitude type, for example mb, Ms, ML	Y
	magnitude	*float(24)*	Magnitude	Y
	uncertainty	*float(24)*	Magnitude uncertainty	Y
	magres	*float(24)*	Magnitude residual	Y
	magdef	*varchar2(1)*	"d" or "n" flag indicating if magnitude is defining or nondefining	Y
	mmodel	*varchar2(15)*	Magnitude model	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y

I	lddate	*date*	Load date	Y

Primary key	PK_STAMAG(magid, ampid, sta)
Indexes	SMAMPIDX(ampid), STAMAG_LDDATEX(lddate), SMARIDX(arid), PK_STAMAG(magid, ampid, sta), SMMAGIDX(magid, sta), SMORIDX(orid)

3.1.141. STAMAG

Owner(s)	REB
Category	Core, Fundamental
Description	The STAMAG table contains station magnitude estimates based upon measurements made on specific seismic phases. Values in STAMAG are used to calculate network magnitudes stored in NETMAG.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	magid	*number(10,0)*	Station magnitude identifier	N
PI	ampid	*number(10,0)*	Amplitude identifier	N
PI	sta	*varchar2(6)*	Station code	N
I	arid	*number(10,0)*	Arrival identifier	Y
I	orid	*number(10,0)*	Origin identifier	Y
	evid	*number(10,0)*	Event identifier	Y
	phase	*varchar2(8)*	Phase type	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	magtype	*varchar2(6)*	Magnitude type, for example mb, Ms, ML	Y
	magnitude	*float(24)*	Magnitude	Y
	uncertainty	*float(24)*	Magnitude uncertainty	Y
	magres	*float(24)*	Magnitude residual	Y
	magdef	*varchar2(1)*	"d" or "n" flag indicating if magnitude is defining or nondefining	Y
	mmodel	*varchar2(15)*	Magnitude model	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
I	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_STAMAG(magid, ampid, sta)
Indexes	SMAMPIDX(ampid), STAMAG_LDDATEX(lddate), SMARIDX(arid), PK_STAMAG(magid, ampid, sta), SM_ARID_COMMID_X(arid, commid), SMMAGIDX(magid, sta), SMORIDX(orid)

3.1.142. STAMAG

Owner(s)	SEL2, SEL3, SEL1

Category	Core, Fundamental
Description	The STAMAG table contains station magnitude estimates based upon measurements made on specific seismic phases. Values in STAMAG are used to calculate network magnitudes stored in NETMAG.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	magid	*number(10,0)*	Station magnitude identifier	N
PI	ampid	*number(10,0)*	Amplitude identifier	N
PI	sta	*varchar2(6)*	Station code	N
I	arid	*number(10,0)*	Arrival identifier	Y
I	orid	*number(10,0)*	Origin identifier	Y
	evid	*number(10,0)*	Event identifier	Y
	phase	*varchar2(8)*	Phase type	Y
	delta	*float(24)*	Source-receiver distance in degrees	Y
	magtype	*varchar2(6)*	Magnitude type, for example mb, Ms, ML	Y
	magnitude	*float(24)*	Magnitude	Y
	uncertainty	*float(24)*	Magnitude uncertainty	Y
	magres	*float(24)*	Magnitude residual	Y
	magdef	*varchar2(1)*	"d" or "n" flag indicating if magnitude is defining or nondefining	Y
	mmodel	*varchar2(15)*	Magnitude model	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_STAMAG(magid, ampid, sta)
Indexes	STAMAG_LDDATEX(lddate), SMARIDX(arid), PK_STAMAG(magid, ampid, sta), SMORIDX(orid), SMMAGIDX(magid, sta)

3.1.143. STANET

Owner(s)	STATIC
Category	Core, Reference
Description	The STANET table groups array sites into an array "network." Its structure is identical to the AFFILIATION table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	net	*varchar2(8)*	Network identification string	N
PI	sta	*varchar2(6)*	Station code	N
	lddate	*date*	Load date	Y

Primary key	PK_STANET(net, sta)
Indexes	STANETSTAX(sta), PK_STANET(net, sta)

3.1.144. STASSOC

Owner(s)	IDCX
Category	Core, Fundamental
Description	The STASSOC table contains summary information about groups of related arrivals. This table defines the group of phases seen at a single station from the same event.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	stassid	*number(10,0)*	Station association (stassoc) identifier.	N
P	sta	*varchar2(6)*	Station code	Y
	etype	*varchar2(7)*	Event type	Y
	location	*varchar2(32)*	Apparent location description	Y
	dist	*float(24)*	Estimated distance	Y
	azimuth	*float(24)*	Estimated azimuth	Y
	lat	*float(24)*	Geographic latitude. Locations north of the equator have positive latitudes.	Y
	lon	*float(24)*	Geographic longitude. Longitudes are measured positive east of the Greenwich meridian.	Y
	depth	*float(24)*	Estimated depth (kilometers)	Y
I	time	*float(53)*	Estimated origin time	Y
	imb	*float(24)*	Estimated mb	Y
	ims	*float(24)*	Initial estimated Ms	Y
	iml	*float(24)*	Initial estimated ML	Y
	auth	*varchar2(15)*	Name of the application or user making the update	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_STASSOC(stassid)
Indexes	PK_STASSOC(stassid), STASSOC_LDDATEX(lddate), STASSTIMEX(time)

3.1.145. STATION_PHASE_RELY

Owner(s)	STATIC
Category	Interactive Analysis
Description	Table containing reliability estimates for azimuth and slowness at different primary seismic stations.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
	statype	*varchar2(24)*	Station type	Y
	az_rel	*number(8,3)*	Azimuth reliability	Y
	slo_rel	*number(8,3)*	Slowness reliability	Y
	min_azerr	*number(8,3)*	Minimum azimuth error	Y
	max_azerr	*number(8,3)*	Maximum azimuth error	Y
	req_azslo_dist	*number(8,0)*	Maximum distance from hypocenter to disregard azimuth and slowness residuals	Y
	req_azslo_snr	*number(8,0)*	Minimum signal-to-noise ratio required to disregard azimuth and slowness residuals	Y
	lddate	*date*	Load date	N

3.1.146. STA_RELY

Owner(s)	STATIC
Category	Interactive Analysis
Description	Table containing estimates for time, azimuth and slowness reliability at different primary seismic stations.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N
	az_rely	*varchar2(1)*	Azimuth reliability	Y
	slow_rely	*varchar2(1)*	Slowness reliability	Y
	time_rely	*varchar2(1)*	Time reliability	Y
	lddate	*date*	Load date	N

3.1.147. SUBS

Owner(s)	IDCX
Category	Data Services
Description	The SUBS table is used by the Subscription Subsystem. It records what email address receives which products. In addition, it tracks the POC for the products.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	userid	*number(8,0)*	User identifier	N
	intid	*number(10,0)*	Internal identifier for message tracking	Y
	intidtype	*varchar2(16)*	Identifier type for the intid	Y
	subsid	*number(10,0)*	Subscription identifier	Y
	subsname	*varchar2(24)*	Subscription name	Y

PI	prodid	*number(10,0)*	Product identifier	N
	address	*varchar2(64)*	FTP and email address of destination	Y
	deliv_meth	*varchar2(6)*	Delivery method (email or ftp)	Y
	status	*varchar2(6)*	Status of this row	Y
	ondate	*date*	The date subscription is "on"	Y
	offdate	*date*	The date that the subscription is "off"	Y
	initialdate	*date*	Initial date of subscription	Y
	lddate	*date*	Load date	Y

Primary key	PK_SUBS(userid, prodid)
Indexes	PK_SUBS(userid, prodid)

3.1.148. SUBSUSER

Owner(s)	IDCX
Category	Data Services
Description	The SUBSUSER table is used by the Subscription Subsystem to track authorized users of the system. A user is identified by the username and domain from the email header.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	userid	*number(8,0)*	User identifier	N
	username	*varchar2(24)*	UNIX login or user name component of email address	Y
	domain	*varchar2(48)*	Domain name from the incoming subscription message	Y
	status	*varchar2(24)*	Status of this user	Y
	pocid	*number(8,0)*	Point of contact identifier	Y
	priority	*number(2,0)*	User priority	Y
	lddate	*date*	Load date	Y

Primary key	PK_SUBSUSER(userid)
Indexes	PK_SUBSUSER(userid)

3.1.149. THIRDMOM

Owner(s)	REB
Category	Fundamental
Description	The THIRDMOM table contains the third moment of frequency and the percentage of signal frequency amplitudes greater than the corresponding noise frequency amplitudes for an arrival identified by arid. The measurements are used for event screening.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	arid	*number(10,0)*	Arrival identifier	N
	sta	*varchar2(6)*	Station code	N
	rectype	*varchar2(8)*	Recipe type	N
	tmf	*float(24)*	Third moment of frequency	Y
	tmfpct	*float(24)*	Percentage of signal spectrum amplitudes greater than noise spectrum amplitudes	Y
	lddate	*date*	Load date	Y

Primary key	PK_THIRDMOM(arid)
Indexes	PK_THIRDMOM(arid)

3.1.150. TIMEFREQ

Owner(s)	REB
Category	Fundamental
Description	The TIMEFREQ table contains the time-frequency measurements for event characterization.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	orid	*number(10,0)*	Origin identifier	N
PI	sta	*varchar2(6)*	Station code	N
	rectype	*varchar2(8)*	Recipe type	N
	zavpct	*float(24)*	Ratio of bad points to total (vertical)	Y
	navpct	*float(24)*	Ratio of bad points to total (north)	Y
	eavpct	*float(24)*	Ratio of bad points to total (east)	Y
	xavpct	*float(24)*	Ratio of bad points to total (cross-correlation)	Y
	zavcep	*float(24)*	Average 2-D cepstrum max (vertical)	Y
	navcep	*float(24)*	Average 2-D cepstrum max (north)	Y
	eavcep	*float(24)*	Average 2-D cepstrum max (east)	Y
	zavcor	*float(24)*	Average autocorrelation (vertical)	Y
	navcor	*float(24)*	Average autocorrelation (north)	Y
	eavcor	*float(24)*	Average autocorrelation (east)	Y
	xcor	*float(24)*	Zero cross-correlation	Y
	lddate	*date*	Load date	Y

Primary key	PK_TIMEFREQ(orid, sta)
Indexes	PK_TIMEFREQ(orid, sta)

3.1.151. TIMESTAMP

Owner(s)	IDCX
Category	Distributed Processing
Description	The TIMESTAMP table is used for scheduling automatic processing of time-series data.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	procclass	*varchar2(16)*	Process class	N
PI	procname	*varchar2(16)*	Process name	N
	time	*float(53)*	Last epoch time	Y
	lddate	*date*	Load date	Y

Primary key	TIMESTAMP_PK(procclass, procname)
Indexes	TIMESTAMP_PK(procclass, procname)

3.1.152. WEIGHTS

Owner(s)	STATIC
Category	Data Services
Description	The WEIGHTS table is used to store weighting information for calculating the weights of REB events. Only those events with sufficient weight are migrated from the LEB to the REB.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	phase	*varchar2(8)*	Phase type	N
PI	statype	*varchar2(4)*	Station type (ar or ss)	N
PI	net	*varchar2(8)*	Network identification string	N
	tweight	*float(24)*	Time weight	Y
	aweight	*float(24)*	Azimuth weight	Y
	sweight	*float(24)*	Slowness weight	Y
	ondate	*number(8,0)*	First julian date this row is valid	Y
	offdate	*number(8,0)*	Last Julian date this row is valid	Y
	lddate	*date*	Load date	Y

Primary key	SYS_C0028356(phase, statype, net)
Indexes	PK_WEIGHTS(phase, statype, net)

3.1.153. WFAUX

Owner(s)	IDCX
Category	Fundamental

Description	Not used at the IDC.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	wfid	*number(10,0)*	waveform identifier	N
	length	*number(10,0)*	waveform length, bytes	N

Primary key	WFAUX_PK(wfid)
Indexes	WFAUXX(wfid)

3.1.154. WFCONV

Owner(s)	IDCX
Category	Data Services
Description	Not used at the IDC.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	sta	*varchar2(6)*	station code	N
PI	chan	*varchar2(8)*	channel code	N
	chanid	*number(8,0)*	channel identifier	Y
	inauth	*varchar2(1)*	input authenticated (y or n)	Y
	incomp	*varchar2(2)*	input compression type	Y
	intype	*varchar2(2)*	input fixed-width data type	Y
	insamp	*number(8,0)*	input samples per packet	Y
	outauth	*varchar2(1)*	output authenticated (y or n)	Y
	outcomp	*varchar2(2)*	output compression type	Y
	outtype	*varchar2(2)*	output fixed-width data type	Y
	outsamp	*number(8,0)*	output samples per packet	Y
	strip	*varchar2(1)*	data stripped of headers	Y
	commid	*number(10,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

Primary key	PK_WFCONV(sta, chan)
Indexes	PK_WFCONV(sta, chan)

3.1.155. WFDISC

Owner(s)	SEGMENT
Category	Core
Description	The WFDISC table contains a waveform header file and descriptive information. This table provides a pointer (or index) to waveforms stored on disk. The waveforms themselves are stored in ordinary disk files called waveform or ".w" (dotw) files as a sequence of sample values (usually in a binary representation, as defined in the data-

	type column).

Pk/Fk/I	Column	Storage type	Description	Nullable
I	sta	*varchar2(6)*	Station code	N
	chan	*varchar2(8)*	Channel name	N
I	time	*float(53)*	Start epochal time of the waveform segment	N
PI	wfid	*number(10,0)*	Unique waveform identifier (integer)	N
	chanid	*number(8,0)*	Channel recording identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	endtime	*float(53)*	End epochal time of the waveform segment	Y
	nsamp	*number(8,0)*	Number of samples in the waveform segment	Y
	samprate	*float(24)*	Sample rate for this record (samples per second)	Y
	calib	*float(24)*	Nominal calibration	Y
	calper	*float(24)*	Calibration period (seconds)	Y
	instype	*varchar2(6)*	Instrument type (e.g. CMG3ES, Akashi, HMP45A)	Y
	segtype	*varchar2(1)*	Waveform segment type indicating if the waveform is original, segmented, virtual or duplicate	Y
	datatype	*varchar2(2)*	Data type (e.g. s3, s4, t4)	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
I	dir	*varchar2(64)*	Absolute path to the data file	Y
I	dfile	*varchar2(32)*	Waveform data file name	Y
	foff	*number(10,0)*	Byte offset of data segment within file	Y
	commid	*number(8,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_WFDISC(wfid)
Indexes	PK_WFDISC(wfid), WFDISC_LDDATEX(lddate), WFSTAX(sta, time), WFDIRX(dir, dfile), WFTIMEX(time)

3.1.156. WFDISC

Owner(s)	IDCX
Category	Core
Description	The WFDISC table contains a waveform header file and descriptive information. This table provides a pointer (or index) to waveforms stored on disk. The waveforms themselves are stored in ordinary disk files called waveform or ".w" (dotw) files as a sequence of sample values (usually in a binary representation, as defined in the data-type column).

Pk/Fk/I	Column	Storage type	Description	Nullable
I	sta	*varchar2(6)*	Station code	N
I	chan	*varchar2(8)*	Channel name	N
I	time	*float(53)*	Start epochal time of the waveform segment	N
PI	wfid	*number(10,0)*	Unique waveform identifier (integer)	N
	chanid	*number(8,0)*	Channel recording identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
I	endtime	*float(53)*	End epochal time of the waveform segment	Y
	nsamp	*number(8,0)*	Number of samples in the waveform segment	Y
	samprate	*float(24)*	Sample rate for this record (samples per second)	Y
	calib	*float(24)*	Nominal calibration	Y
	calper	*float(24)*	Calibration period (seconds)	Y
	instype	*varchar2(6)*	Instrument type (e.g. CMG3ES, Akashi, HMP45A)	Y
	segtype	*varchar2(1)*	Waveform segment type indicating if the waveform is original, segmented, virtual or duplicate	Y
	datatype	*varchar2(2)*	Data type (e.g. s3, s4, t4)	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
I	dir	*varchar2(64)*	Absolute path to the data file	Y
I	dfile	*varchar2(32)*	Waveform data file name	Y
	foff	*number(10,0)*	Byte offset of data segment within file	Y
	commid	*number(10,0)*	Comment identifier	Y
I	lddate	*date*	Load date	Y

Primary key	PK_WFDISC(wfid)
Indexes	WF_TIME_ENDTIME_IDX(time, endtime), WFDISC_LDDATEX(lddate), PK_WFDISC(wfid), WFSTAX(sta, time, endtime, chan), WFTIMEX(time), WFDIRX(dir, dfile)
Triggers	WFDISC_ENDTIME, WFDISC_NVIAR_ENDTIME

3.1.157. WFDISC_NOMIG

Owner(s)	SEGMENT
Description	Obsolete.

Pk/Fk/I	Column	Storage type	Description	Nullable
	sta	*varchar2(6)*	Station code	N

	chan	*varchar2(8)*	Channel name	N
	time	*float(53)*	Start epochal time of the waveform segment	N
	wfid	*number(8,0)*	Unique waveform identifier (integer)	N
	chanid	*number(8,0)*	Channel recording identifier	Y
	jdate	*number(8,0)*	Julian date for this data	Y
	endtime	*float(53)*	End epochal time of the waveform segment	Y
	nsamp	*number(8,0)*	Number of samples in the waveform segment	Y
	samprate	*float(24)*	Sample rate for this record (samples per second)	Y
	calib	*float(24)*	Nominal calibration	Y
	calper	*float(24)*	Calibration period (seconds)	Y
	instype	*varchar2(6)*	Instrument type (e.g. CMG3ES, Akashi, HMP45A)	Y
	segtype	*varchar2(1)*	Waveform segment type indicating if the waveform is original, segmented, virtual or duplicate	Y
	datatype	*varchar2(2)*	Data type (e.g. s3, s4, t4)	Y
	clip	*varchar2(1)*	Flag indicating whether or not the data are clipped	Y
	dir	*varchar2(64)*	Absolute path to the data file	Y
	dfile	*varchar2(32)*	Waveform data file name	Y
	foff	*number(10,0)*	Byte offset of data segment within file	Y
	commid	*number(8,0)*	Comment identifier	Y
	lddate	*date*	Load date	Y

3.1.158. WFTAG

Owner(s)	SEL2, IDCX
Category	Core, Fundamental
Description	The WFTAG table links various identifiers (for example, orid, arid, and stassid to wfid). Linkages can also be determined indirectly using sta, chan, and time. However, it is more efficient to use the WFTAG table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	tagname	*varchar2(8)*	Key (arid, orid, evid, and so on)	N
PI	tagid	*number(10,0)*	Tagname value	N
PI	wfid	*number(10,0)*	Waveform identifier	N
	lddate	*date*	Load date	Y

Primary key	PK_WFTAG(tagname, tagid, wfid)
Indexes	WFTWFIDX(wfid), PK_WFTAG(tagname, tagid, wfid), WFTTAGIDX(tagid)

3.1.159. WFTAG

Owner(s)	SEGMENT
Category	Core, Fundamental
Description	The WFTAG table links various identifiers (for example, orid, arid, and stassid to wfid). Linkages can also be determined indirectly using sta, chan, and time. However, it is more efficient to use the WFTAG table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	tagname	*varchar2(8)*	Key (arid, orid, evid, and so on)	N
PI	tagid	*number(8,0)*	Tagname value	N
PI	wfid	*number(10,0)*	Waveform identifier	N
	lddate	*date*	Load date	Y

Primary key	PK_WFTAG(tagname, tagid, wfid)
Indexes	WFTWFIDX(wfid), PK_WFTAG(tagname, tagid, wfid), WFTTAGIDX(tagid)

3.1.160. WFTAG

Owner(s)	SEL3, SEL1
Category	Core, Fundamental
Description	The WFTAG table links various identifiers (for example, orid, arid, and stassid to wfid). Linkages can also be determined indirectly using sta, chan, and time. However, it is more efficient to use the WFTAG table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	tagname	*varchar2(8)*	Key (arid, orid, evid, and so on)	N
PI	tagid	*number(10,0)*	Tagname value	N
PI	wfid	*number(10,0)*	Waveform identifier	N
	lddate	*date*	Load date	Y

Primary key	PK_WFTAG(tagname, tagid, wfid)
Indexes	WFTWFIDX(wfid), PK_WFTAG(tagname, tagid, wfid), WFTTAGIDX(tagid)

3.1.161. XTAG

Owner(s)	IDCX
Category	Database and Utility
Description	The XTAG table links various identifiers (for example, orid, arid, stassid, and wfid to other identifiers). This table is a generalization of the WFTAG table, which is limited to linking exclusively to the wfid. The thisdb column describes the database account

	for the record specified by thisid and thisname; thatdb describes the database account for the record specified by thatid and thatname. When a parent/child relationship exists between the records, thisid should designate the parent, and thatid should designate the child.

Pk/Fk/I	Column	Storage type	Description	Nullable
	thisid	*number(10,0)*	Thisname identifier	Y
I	thatid	*number(10,0)*	Thatname identifier	Y
	thisname	*varchar2(8)*	Key for thisid (grid, orid, ntid, and so on)	Y
	thatname	*varchar2(8)*	Key for thatid (arid, orid, nfid, and so on)	Y
	thisdb	*varchar2(32)*	Database account for the records specified by thisid and thisname	Y
	thatdb	*varchar2(32)*	Database account for the records specified by thatid and thatname	Y
	lddate	*date*	Load date	Y

Indexes	XTAGTHATIDNDX(thatid)

3.2. S/H/I Column Descriptions

3.2.1. account

Table(s)	DATAREADY (IDCX)			
Description	Database account name. The tagid and tagname pointing to the data to be distributed by the Subscription Subsystem are in this database account.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	N
Range	any string up to 24 characters			

3.2.2. acoef

Table(s)	SPVAR (REB)			
Description	Coefficient "a" of the quadratic trend of the log spectrum between frequencies fmin and fmax. The spectrum is measured in nm-sec.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	any floating point value			

3.2.3. action

Table(s)	ALLOCATE_HOUR (LEB), ALLOCATE_HOUR (REB)			
Description	Analyst task performed on this time-block of data.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	N
Range	action IN {AfterShock, Allocate, Bull_QC, Del_Pass, RebDone, Scan_Pass}			

3.2.4. address

Table(s)	SUBS (IDCX)			
Description	FTP and email address (normally fully qualified in the form user@domain) for a user of the Subscription system.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	Y
Range	any valid email address			

3.2.5. algorithm

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)
Description	Location algorithm used. This column is a brief textual description of the al-

	gorithm used for computing a seismic origin.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any string up to the character field width			

3.2.6. amcor

Table(s)	SITEAUX (STATIC)			
Description	Site-dependent log amplitude correction.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f10.1	Y
Range	amcor > -999.0			

3.2.7. amcorsd

Table(s)	SITEAUX (STATIC)			
Description	Standard deviation for log amplitude correction.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f5.2	Y
Range	amcorsd > 0.0			

3.2.8. amp

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Measured amplitude defined by amptype.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers or dimensionless depending on the type of channel	-1.0	f11.2	Y
Range	amp > 0.0			

3.2.9. amp

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Measured amplitude defined by amptype. Amplitude is greater than zero in all cases except amptype=FmSig, which can be negative (and is for around 50% of cases).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers or dimensionless depending on the type of channel	-1.0	f11.2	Y
Range	amp > 0.0 (except FmSig: can be negative)			

3.2.10. amp_hi

Table(s)	BREGION_TAB (STATIC)			
Description	Upper bound of the amplitude window. Never used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	nanometers	-1	f8.3	Y
Range				

3.2.11. amp_lo

Table(s)	BREGION_TAB (STATIC)			
Description	Lower bound of the amplitude window. Never used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	nanometers	-1	f8.3	Y
Range				

3.2.12. ampid

Table(s)	STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Amplitude identifier. Every amplitude measure is assigned a unique positive integer that identifies it in the database. If an associated stamag record exists, then ampid links it to amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	ampid > 0			

3.2.13. ampid

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Amplitude identifier. Every amplitude measure is assigned a unique positive integer that identifies it in the database. If an associated stamag record exists, then ampid links it to amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	ampid > 0			

3.2.14. amplr

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)
Description	Maximum 3-component amplitude for all overlapping time windows used in the polarization analysis. This column is equal to the sum of the square roots of the eigenvalues. The only difference between amps and amplr is in the definition of the overlapping time windows.

Storage type	Unit	NA value	External	Nullable
float(24)	nanometers	-1.0	f7.2	Y
Range	amplr > 0.0			

3.2.15. ampp

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	3-component amplitude measured at the time of the maximum rectilinearity. This column is equal to the sum of the square roots of the eigenvalues (that is, it is the sum of the amplitudes measured along the three axes of the polarization ellipsoid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers	-1.0	f7.2	Y
Range	ampp > 0.0			

3.2.16. amps

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Maximum 3-component amplitude for all overlapping time windows used in the polarization analysis. This column is equal to the sum of the square roots of the eigenvalues. The only difference between amps and amplr is in the definition of the overlapping time windows.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers	-1.0	f7.2	Y
Range	amps > 0.0			

3.2.17. amptime

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Epoch time of amplitude measure.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.18. amptype

Table(s)	AMPDESCRIPT (IDCX)			
Description	Amplitude measure descriptor. This descriptor is used to uniquely identify an amplitude measurement and link the description in ampdescript with actual measurements in amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	N
Range	any free-format string up to eight characters			

3.2.19. amptype

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Amplitude measure descriptor. This descriptor is used to uniquely identify an amplitude measurement and link the description in ampdescript with actual measurements in amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	any free-format string up to eight characters			

3.2.20. apmarid

Table(s)	APMA (IDCX), APMA (LEB)			
Description	Amplitude measure descriptor. This descriptor is used to uniquely identify an amplitude measurement and link the description in ampdescript with actual measurements in amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		- (hyphen)	a8	Y
Range	any free-format string up to eight characters			

3.2.21. apmarid

Table(s)	APMA (REB)			
Description	Unique apma recipe identifier. Each arrival in apma is assigned a positive integer identifying it with the recipe used in the polarization analysis.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	apmarid > 0			

3.2.22. archid

Table(s)	ARCH_DATA_TYPE (IDCX)			
Description	Archive identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	archid > 0			

3.2.23. archiveport

Table(s)	DLMAN (IDCX)			
Description	Archiver port.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(6,0)		-1	i6	Y

Range	0 <= archiveport <= 16383

3.2.24. arid

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB)			
Description	Arrival identifier. Each arrival is assigned a positive integer (if available) identifying it with a unique sta, chan, and time. Note that arid in the AMP3C, AMPLITUDE and STAMAG tables is also allowed to have a NA value of -1.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	arid >= -1			

3.2.25. arid

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), FKDISC (LEB), HYDRO_ASSOC (IDCX), HYDRO_ASSOC (LEB), HYDRO_ASSOC (REB), HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB), INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB), PMCC_FEATURES (IDCX), SPVAR (REB), THIRDMOM (REB)			
Description	Arrival identifier. Each arrival is assigned a positive integer identifying it with a unique sta, chan, and time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	arid > 0			

3.2.26. arid

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Arrival identifier. Each arrival is assigned a positive integer (if available) identifying it with a unique sta, chan, and time. Note that arid in the AMP3C, AMPLITUDE and STAMAG tables is also allowed to have a NA value of -1.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	arid >= -1			

3.2.27. array

Table(s)	REQUEST (IDCX)
Description	Array code. Network or station name.

Storage type	Unit	NA value	External	Nullable
varchar2(8)		NOT ALLOWED	a8	Y
Range	any string matching the net column in Affiliation (Stanet)			

3.2.28. attenid

Table(s)	ATTENCOEF (STATIC)			
Description	Attenuation coefficient set identifier. Each set of attenuation coefficients is assigned a unique positive integer to identify the change history.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		- (hyphen)	a20	N
Range	any valid string up to 20 characters			

3.2.29. attribute

Table(s)	NA_VALUE (IDCX)			
Description	Name of the column to which a NA value is to be assigned.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)		NOT ALLOWED	a30	N
Range	any valid string up to the character width			

3.2.30. auth

Table(s)	REVAUDIT (LEB)			
Description	Author, the originator of the data. Auth may also identify an application generating or modifying the record, such as an automated interpretation or signal-processing program.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	N
Range	any string up to the column character width			

3.2.31. auth

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), APMA (IDCX), APMA (LEB), APMA (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), EVENT (LEB), EVENT (REB), EVENT (SEL1), EVENT (SEL2), EVENT (SEL3), NETWORK (STATIC), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), OUTAGE (IDCX), PMCC_FEATURES (IDCX), QCSTATS (IDCX), SITEAUX (STATIC), STASSOC (IDCX)			
Description	Author, the originator of the data. Auth may also identify an application generating or modifying the record, such as an automated interpretation or signal-processing program.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y

Range	any string up to the column character width

3.2.32. auth

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB), NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Author, the originator of the data. Auth may also identify an application generating the record, such as an automated interpretation or signal-processing program.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any string up to the column character width			

3.2.33. auth

Table(s)	DISCARD (LEB), DISCARD (REB)			
Description	Name of the Analyst discarding the event			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any string up to the column character width			

3.2.34. auth

Table(s)	ALLOCATE_HOUR (LEB), ALLOCATE_HOUR (REB)			
Description	Name of the Analyst processing the time block			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y
Range	any string up to the column character width			

3.2.35. auth

Table(s)	PMCC_RECIPE (IDCX)			
Description	Author, the originator of the data. Auth may also identify an application generating or modifying the record, such as an automated interpretation or signal-processing program.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(17)		- (hyphen)	a17	Y
Range	any string up to the column character width			

3.2.36. author

Table(s)	FILEPRODUCT (IDCX)

Description	Author of the data pointed to by the dir and dfile attributes			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y
Range	any string up to the character field width			

3.2.37. author

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	Y
Range				

3.2.38. author

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)				Y
Range				

3.2.39. auxid

Table(s)	OUTAGE (IDCX)			
Description	Auxiliary identification code.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	Y
Range	any valid auxiliary code up to four characters			

3.2.40. available

Table(s)	OUTAGE (IDCX)			
Description	Flag to specify if data are available (t) or unavailable (f).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		NOT ALLOWED	a1	Y
Range	available IN {t, f}			

3.2.41. ave_noise

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)
Description	Average pressure of the noise segment.

Storage type	Unit	NA value	External	Nullable
float(24)	dB re mPa	-1.0	f9.4	Y
Range	ave_noise > 0.0			

3.2.42. avgconstval

Table(s)	QCSTATS (IDCX)			
Description	Average value of data in masked constant segments.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)		-999.0	f17.5	Y
Range	aveconstval >= 0.0			

3.2.43. aweight

Table(s)	WEIGHTS (STATIC)			
Description	Azimuth weight.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		0.0	f5.2	Y
Range	aweight >= 0.0			

3.2.44. az1

Table(s)	HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB)			
Description	Azimuth estimated from the time lags of arrivals in a hydro-arrival group.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= az1 < 360.0			

3.2.45. az2

Table(s)	HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB)			
Description	Azimuth estimated from the time lags of arrivals in a hydro-arrival group. This second azimuth estimate is only needed when only two arrivals exist in a group, which results in an ambiguity between two equally likely azimuths. The error is the same for the two azimuths.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= az2 < 360.0			

3.2.46. az_rel

Table(s)	STATION_PHASE_RELY (STATIC)

Description	Azimuth reliability estimate			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)		-1	f8.3	Y
Range	0.0 <= az_rel <= 1.0			

3.2.47. az_rely

Table(s)	STA_RELY (STATIC)			
Description	Azimuth reliability flag. The value is a single-character flag to indicate whether (y) or not (n) the azimuth is considered to be reliable.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	Y
Range	az_rely IN { 0,1}			

3.2.48. az_tol1

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.49. az_tol2

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.50. az_weight

Table(s)	PHASE_WEIGHT_EVENT_DEFINE (STATIC)			
Description	Weight given to defining azimuth for this phase type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	N
Range				

3.2.51. azcontrib

Table(s)	HYDRO_ASSOC (IDCX), HYDRO_ASSOC (LEB), HYDRO_ASSOC (REB)
Description	Flag that specifies if an arrival that belongs to a hydro arrival group was used to calculate the azimuth.

Storage type	Unit	NA value	External	Nullable
varchar2(1)		- (hyphen)	a1	Y
Range	azcontrib IN {y, n}			

3.2.52. azdef

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Azimuth-defining code. The one-character flag indicates whether or not the azimuth of a phase was used to constrain the event location solution. This column is defining (azdef = d) if it was used in the location, non defining (azdef = n) if it was not.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	azdef IN {d, n}			

3.2.53. azi_hi

Table(s)	BREGION_TAB (STATIC)			
Description	Maximal azimuth to consider.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	0.0 <= azi_hi <= 360.0			

3.2.54. azi_lo

Table(s)	BREGION_TAB (STATIC)			
Description	Minimal azimuth to consider.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	0.0 <= azi_lo <= 360.0			

3.2.55. azimuth

Table(s)	STASSOC (IDCX)			
Description	Estimated azimuth. This value is the estimated station-to-event azimuth measured clockwise from North.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= azimuth < 360.0			

3.2.56. azimuth

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Observed azimuth. This value is the estimated station-to-event azimuth measured clockwise from North. The estimate can be made in a variety of ways, including f-k or polarization analysis.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= azimuth < 360.0			

3.2.57. azimuth

Table(s)	PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3)			
Description	Predicted azimuth. This value is the estimated station-to-event azimuth measured clockwise from North.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= azimuth < 360.0			

3.2.58. azres

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC)			
Description	Allowable azimuth residual			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	NOT ALLOWED	f7.1	N
Range	-180.0 <= azres <= 180.0			

3.2.59. azres

Table(s)	ALLOW_RESID (LEB), ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Azimuth residual. The value is the difference between the measured station-to-event azimuth for an arrival and the true azimuth. The true azimuth is the bearing to the inferred event origin.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f7.1	Y
Range	-180.0 <= azres <= 180.0			

3.2.60. band

Table(s)	INSTRUMENT (STATIC)
Description	Frequency band. The value is a qualitative indicator of frequency passband for an instrument. Values should reflect the response curve rather than just the

	sample rate. Recommended values are as follows: s (short-period) m (mid-period) i (intermediate-period) l (long-period) b (broadband) h (high-frequency, very short-period) v (very long-period) For a better notion of the instrument characteristics, see the instrument response curve.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	band IN {s, m, i, l, b, h, v}			

3.2.61. bandw

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	Difference between the high and low frequency bands used in the processing detection or amplitude processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f7.3	Y
Range	bandw > 0.0			

3.2.62. bandw

Table(s)	FKDISC (LEB)			
Description	Frequency bandwidth.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f7.3	Y
Range	bandw > 0.0			

3.2.63. bcoef

Table(s)	SPVAR (REB)			
Description	Coefficient "b" of the quadratic trend of the log spectrum between frequencies fmin and fmax. The spectrum is measured in nm-sec.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	any floating point value			

3.2.64. begin_date

Table(s)	PARTICIPATION (STATIC)			
Description	Beginning date of station participation in a particular net.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	any valid Julian date			

3.2.65. belief

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Phase identification confidence level. The value is a qualitative estimate of the confidence that a seismic phase is correctly identified.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f4.2	Y
Range	0.0 <= belief <= 1.0			

3.2.66. bmtyp

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	String indicating a coherent (coh), incoherent (inc), or horizontal (hor) beam type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	Y
Range	bmtyp IN {coh, inc, hor}			

3.2.67. bordercolor

Table(s)	MAPDISC (MAP)			
Description	Map border color name. A solid colored border may appear on the top, bottom, and right of any raster map.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	any string up to 32 characters that forms a valid X11 color name (for example, black)			

3.2.68. bregid

Table(s)	BREGION_DEF (STATIC), BREGION_TAB (STATIC)			
Description	Region identification number			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	bregid > 0			

3.2.69. bregname

Table(s)	BREGION_DEF (STATIC), BREGION_TAB (STATIC)			
Description	Name of the region.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		NOT ALLOWED	a32	N

Range	any valid string up to 32 characters

3.2.70. calib

Table(s)	OLD_WFPROTO (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Calibration factor. The value is the conversion factor that maps digital data to earth displacement. The factor holds true at the oscillation period specified by the column calper. A positive value means ground motion increasing in component direction (up, north, east) is indicated by increasing counts. A negative value means the opposite. Calib generally reflects the best calibration information available at the time of recording, but refinement may be given in Sensor, reflecting a subsequent recalibration of the instrument (see calratio).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers/digital count	NOT ALLOWED	f16.6	Y
Range	any nonzero floating point number			

3.2.71. calper

Table(s)	OLD_WFPROTO (IDCX), SENSOR (STATIC), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Calibration period. This gives the period for which calib, ncalib, and calratio are valid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	NOT ALLOWED	f16.6	Y
Range	calper > 0.0			

3.2.72. calratio

Table(s)	SENSOR (STATIC)			
Description	Calibration conversion ratio. The value is a dimensionless calibration correction factor that permits small refinements to the calibration correction made using calib and calper from the Wfdisc table. Often, the wfdisc calib contains the nominal calibration assumed at the time of data recording. If the instrument is recalibrated, calratio provides a mechanism to update calibrations from wfdisc with the new information without modifying the wfdisc table. A positive value means ground motion increasing in component direction (up, north, east) is indicated by increasing counts. A negative value means the opposite. Calratio is meant to reflect the most accurate calibration information for the time period for which the sensor record is appropriate, but the nominal value may appear until other information is available.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f16.6	Y
Range	any nonzero floating point number			

3.2.73. capability

Table(s)	NOOPS_CHANNAME (STATIC)			
Description	Capability. These values are used by the Station Quality application, squal. cd2w only processes channels with a capability less than ten (10). In addition, cd2w only updates the waveinterval table for channels with a capability equal to one (1).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	Y
Range	-1 <= capability < 18			

3.2.74. capability

Table(s)	CHANNAME (STATIC)			
Description	Not used by CD2WNG. These values were used by the Station Quality application, squal.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	Y
Range	-1 <= capability < 18			

3.2.75. ccoef

Table(s)	SPVAR (REB)			
Description	Coefficient "c" of the quadratic trend of the log spectrum between frequencies fmin and fmax. The spectrum is measured in nm-sec.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	any floating point value			

3.2.76. cep_delay_time_signal

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Bubble pulse delay time estimate where the cepstrum is computed from the NSE-detrended spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f9.4	Y
Range	cep_delay_time_signal > 0.0			

3.2.77. cep_delay_time_trend

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)
Description	Bubble pulse delay time estimate where the cepstrum is computed from the trend of NSE-detrended spectrum.

Storage type	Unit	NA value	External	Nullable
float(24)	seconds	-1.0	f9.4	Y
Range	cep_delay_time_trend > 0.0			

3.2.78. cep_peak_std_signal

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Number of standard deviations from the mean for the largest cepstrum amplitude where the cepstrum is computed from the NSE-detrended spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f9.4	Y
Range	cep_peak_std_signal > 0.0			

3.2.79. cep_peak_std_trend

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Number of standard deviations from the mean for the largest cepstrum amplitude where the cepstrum is computed with the trend of the NSE-detrended spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f9.4	Y
Range	cep_peak_std_trend > 0.0			

3.2.80. cep_var_signal

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Variance of the cepstrum computed using NSE-detrended spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f9.4	Y
Range	cep_var_signal > 0.0			

3.2.81. cep_var_trend

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Variance of the cepstrum computed using the trend from the NSE-detrended spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f9.4	Y
Range	cep_var_trend > 0.0			

3.2.82. cfreq

Table(s)	PMCC_FEATURES (IDCX)			
Description	Central frequency (Hz)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f7.2	Y
Range	cfreq > 0.0			

3.2.83. cfreq

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	Central frequency of a beam or f-k spectrum (Hz)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f7.2	Y
Range	cfreq > 0.0			

3.2.84. cfreq

Table(s)	FKDISC (LEB)			
Description	Centre frequency of a beam or f-k spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f7.2	Y
Range	cfreq > 0.0			

3.2.85. chan

Table(s)	SITEPOLL (STATIC)			
Description	Channel identifier. The value is an eight-character code that specifies a particular channel within a network (station), which, taken together with sta and time, uniquely identifies seismic timeseries data, including the geographic location, spatial orientation, sensor, and subsequent data processing (beam channel descriptor).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any alphanumeric string up to the character field width			

3.2.86. chan

Table(s)	SITEPOLLCA (STATIC)
Description	Channel identifier. The value is an eight-character code that specifies a particular channel within a network (station), which, taken together with sta and time, uniquely identifies seismic timeseries data, including the geographic location, spatial orientation, sensor, and subsequent data processing (beam channel descriptor).

Storage type	Unit	NA value	External	Nullable
varchar2(6)		NOT ALLOWED	a6	Y
Range	any alphanumeric string up to the character field width			

3.2.87. chan

Table(s)	ATTENCOEF (STATIC), BEAMAUX (STATIC), CHAN_GROUPS (STATIC), DERVDISC (IDCX), EVSC_REGIONAL (IDCX), FOR-BEAMAUX (STATIC), OLD_WFPROTO (IDCX), PRODUCTTYPESTA (IDCX), QCSTATS (IDCX), REQUEST (IDCX), SENSOR (STATIC), SITEAUX (STATIC), SITECHAN (STATIC), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Channel identifier. The value is an eight-character code that specifies a particular channel within a network (station), which, taken together with sta and time, uniquely identifies seismic timeseries data, including the geographic location, spatial orientation, sensor, and subsequent data processing (beam channel descriptor).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	any alphanumeric string up to the character field width			

3.2.88. chan

Table(s)	WFCONV (IDCX)			
Description	Channel identifier. The value is an eight-character code that specifies a particular channel within a network (station), which, taken together with sta and time, uniquely identifies seismic timeseries data, including the geographic location, spatial orientation, sensor, and subsequent data processing (beam channel descriptor).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	any lower-case string up to eight characters			

3.2.89. chan

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), CHAN_GROUPS_AUXWF (STATIC), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), FILEPRODUCT (IDCX), OUTAGE (IDCX)			
Description	Channel identifier. The value is an eight-character code that specifies a particular channel within a network (station), which, taken together with sta and time, uniquely identifies seismic timeseries data, including the geographic location, spatial orientation, sensor, and subsequent data processing (beam channel descriptor).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	Y

Range	any alphanumeric string up to the character field width

3.2.90. chan

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

3.2.91. chan

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)				Y
Range				

3.2.92. chanid

Table(s)	WFCONV (IDCX)			
Description	Channel recording identifier. The value is a surrogate key used to uniquely identify a specific recording. Chanid duplicates the information of the compound key sta, chan, time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i10	Y
Range	chanid > 0			

3.2.93. chanid

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), OLD_WFPROTO (IDCX), SENSOR (STATIC), SITECHAN (STATIC), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Channel recording identifier. The value is a surrogate key used to uniquely identify a specific recording. Chanid duplicates the information of the compound key sta, chan, time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	chanid > 0			

3.2.94. class

Table(s)	INTERVAL (IDCX), INTERVAL (SEGMENT)
Description	Problem class or object class for interval and request.

Storage type	Unit	NA value	External	Nullable
varchar2(16)		- (hyphen)	a16	N
Range	any free-format string up to the character limit			

3.2.95. class

Table(s)	CHAN_GROUPS (STATIC)			
Description	Type of channel group.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	N
Range	any free-format string up to the character limit			

3.2.96. class

Table(s)	REQUEST (IDCX)			
Description	Problem class or object class for interval and request.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y
Range	any free-format string up to the character limit			

3.2.97. class

Table(s)	CHAN_GROUPS_AUXWF (STATIC)			
Description	Type of channel group.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y
Range	any free-format string up to the character limit			

3.2.98. class

Table(s)	PROBLEM (IDCX)			
Description	Problem class or object class for interval and request.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(33)		- (hyphen)	a33	Y
Range	any free-format string up to the character limit			

3.2.99. clip

Table(s)	OLD_WFPROTO (IDCX)			
Description	Clipped data flag. The value is a single-character flag to indicate whether (c) or not (n) the data were clipped.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

char(1)		- (hyphen)	a1	Y
Range	clip IN {c, n}			

3.2.100. clip

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Clipped data flag. The value is a single-character flag to indicate whether (c) or not (n) the data were clipped.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	clip IN {c, n}			

3.2.101. clrpth

Table(s)	EVSC_PROD (IDCX)			
Description	Hydroacoustic clear-path flag. Set to 1 if all paths within the location error ellipse are clear to at least one hydroacoustic station in a given set.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		-1	i1	Y
Range	clrpth IN {0,1}			

3.2.102. coh_deldur

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Estimated standard deviation in coh_dur.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f7.2	Y
Range	coh_deldur > 0.0			

3.2.103. coh_dur

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Time period during which the spatial coherence (reckoned as F-Statistic) “beam” exceeded its threshold for the current time interval of spatial coherence in excess of its threshold value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	coh_dur > 0.0			

3.2.104. coh_per

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Dominant period of the spatially coherent signal. The value is estimated from the inverse of the frequency of the maximum power in a windowed spectrum of the traditional beamformed waveform data during the time interval defined by coinc_time and coh_dur.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	coh_per > 0.0			

3.2.105. coh_snr

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Equivalent, equal channel, coherent signal-to-noise ratio; the inband power in the spatially coherent fraction of the arrival divided by the total inband non-coherent noise.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f7.2	Y
Range	coh_snr > 0.0			

3.2.106. coh_time

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Earliest time at which the spatial coherence (determined as F-Statistic) “beam” exceeded its threshold for the current time interval of spatial coherence in excess of its threshold value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	–9999999999.999	f17.5	Y
Range	coh_time > –9999999999.999			

3.2.107. coinc_deldur

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Estimated standard deviation in coinc_dur.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f7.2	Y
Range	coinc_deldur > 0.0			

3.2.108. coinc_dur

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Time period during which both the STA/LTA on the traditional beam and the sample-by-sample F-Statistic beams exceed their respective thresholds for the current time interval of coincidence detection.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	coinc_dur > 0.0			

3.2.109. coinc_time

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Earliest time at which the STA/LTA applied to the traditional beam and the spatial coherence “beam” both exceeded their respective thresholds for the current time interval of coincidence detection.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	–9999999999.999	f17.5	Y
Range	coinc_time > –9999999999.999			

3.2.110. colormapid

Table(s)	COLORDISC (MAP), MAPCOLOR (MAP)			
Description	Colordisc identifier. Each colordisc is assigned a unique positive integer that identifies it in a database. The colormapid identifies color-lookup tables available to maps.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	colormapid > 0			

3.2.111. colormapname

Table(s)	COLORDISC (MAP)			
Description	Colormap name that identifies the color-lookup table in a listing of available tables.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	Y
Range	any string up to 64 characters			

3.2.112. colorname

Table(s)	OVERLAYDISC (MAP)
Description	Overlay color name.

Storage type	Unit	NA value	External	Nullable
varchar2(32)		- (hyphen)	a32	N
Range	any valid color name			

3.2.113. command

Table(s)	MSGAUX (IDCX)			
Description	Command that was being executed when the failure occurred. If this cannot be determined, such as a caught signal from UNIX, then the value is set to "signal caught."			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any string up to the character field width that is a valid user name			

3.2.114. commid

Table(s)	OLD_WFPROTO (IDCX)			
Description	Comment identifier. The value is a key that points to free-form comments entered in the REMARK table. These comments store additional information about a record in another table. The REMARK table can have many records with the same commid and different lineno, but the same commid will appear in only one other record among the rest of the tables in the database (see lineno)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		-1	i8	Y
Range	commid > 0			

3.2.115. commid

Table(s)	REMARK (IDCX), REMARK (LEB), REMARK (REB), REMARK (SEL1), REMARK (SEL2), REMARK (SEL3), REMARK (STATIC)			
Description	Comment identifier. The value is a key that points to free-form comments entered in the REMARK table. These comments store additional information about a record in another table. The REMARK table can have many records with the same commid and different lineno, but the same commid will appear in only one other record among the rest of the tables in the database (see lineno)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	commid > 0			

3.2.116. commid

Table(s)	ALLOCATE_HOUR (LEB), ALLOCATE_HOUR (REB), APMA (IDCX), APMA (LEB), APMA (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), DATAUSER (IDCX), DERVDISC (IDCX), DE-

	TECTION (IDCX), DETECTION (LEB), DETECTION (REB), DLMAN (IDCX), EVENT (LEB), EVENT (REB), EVENT (SEL1), EVENT (SEL2), EVENT (SEL3), FKDISC (LEB), INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB), MSGDISC (IDCX), NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), NETWORK (STATIC), ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), OUTAGE (IDCX), PMCC_FEATURES (IDCX), SITEAUX (STATIC), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3), STASSOC (IDCX), WFCONV (IDCX)
Description	Comment identifier. The value is a key that points to free-form comments entered in the REMARK table. These comments store additional information about a record in another table. The REMARK table can have many records with the same commid and different lineno, but the same commid will appear in only one other record among the rest of the tables in the database (see lineno)

Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	commid > 0			

3.2.117. commid

Table(s)	WFDISC (IDCX)
Description	Comment identifier. The value is a key that points to free-form comments entered in the REMARK table. These comments store additional information about a record in another table. The REMARK table can have many records with the same commid and different lineno, but the same commid will appear in only one other record among the rest of the tables in the database (see lineno)

Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i8	Y
Range	commid > 0			

3.2.118. commid

Table(s)	WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)
Description	Comment identifier. The value is a key that points to free-form comments entered in the REMARK table. These comments store additional information about a record in another table. The REMARK table can have many records with the same commid and different lineno, but the same commid will appear in only one other record among the rest of the tables in the database (see lineno)

Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	commid > 0			

3.2.119. complete

Table(s)	REQUEST (IDCX)			
Description	Percentage complete. Percentage of waveform data acquired for this request.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)	percent	NOT ALLOWED	i8	Y
Range	0 <= complete <= 100			

3.2.120. complexity

Table(s)	COMPLEXITY (REB)			
Description	Signal complexity measure estimated by DFX application.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f11.4	Y
Range	complexity >= 0.0			

3.2.121. conf

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Confidence measure for a particular event identification method.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f5.3	Y
Range	0.5 <= conf <= 1.0			

3.2.122. connmanport

Table(s)	DLMAN (IDCX)			
Description	Connection manager (ConnMan) port number used to send messages to the disk-loop manager application.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(6,0)		0	i6	Y
Range	1 <= connmanport <= 16383			

3.2.123. consider

Table(s)	EVSC_PROD (IDCX)			
Description	Flag indicating whether (1) or not (0) an event was considered for screening.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	consider IN {0, 1}			

3.2.124. consistency

Table(s)	PMCC_FEATURES (IDCX)			
Description	Mean consistency value of the detection pixels in the PMCC family (seconds)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.125. const

Table(s)	QCSTATS (IDCX)			
Description	Amount of data in the detection processing interval masked due to constant segments.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999.0	f17.5	Y
Range	const >= 0.0			

3.2.126. constrain_depth

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that tells location process whether or not to fix (constrain) the current hypocentral depth. If TRUE (1), the depth will be fixed to the value specified on the first (summary) line of the DATA file or as specified by the depth column of the Origin (Origin_temp_ga) table. If FALSE (0), the depth is an independent solution parameter. Default is TRUE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	constrain_depth IN {0, 1}			

3.2.127. constrain_latlon

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that tells location process whether or not to fix (constrain) the current epicentral location. If TRUE (1), the latitude and longitude will be fixed to the value specified by the lat and lon columns of the Origin (Origin_temp_ga) table. If FALSE (0), the latitude and longitude are independent solution parameters. Default is FALSE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	constrain_latlon IN {0, 1}			

3.2.128. constrain_ot

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that tells location process whether or not to fix (constrain) the current origin time. If TRUE (1), the origin time will be fixed to the value specified by the time column of the Origin (Origin_temp_ga) table. If FALSE (0), the origin time is an independent solution parameter. Default is FALSE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	constrain_ot IN {0, 1}			

3.2.129. controlport

Table(s)	DLMAN (IDCX)			
Description	DataControl port. Number used to send commands to the diskloop manager application.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(6,0)		0	i6	Y
Range	0 <= controlport <= 16383			

3.2.130. correlation

Table(s)	PMCC_FEATURES (IDCX)			
Description	Mean correlation coefficient of the detection pixels in the PMCC family			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.6	Y
Range				

3.2.131. country

Table(s)	SITE_ADDRESS (STATIC)			
Description	Full country name for a sensor site.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)		- (hyphen)	a40	Y
Range	any string up to 40 characters			

3.2.132. cov_depth_time

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)
Description	Coverage ellipse depth/time conversion factor. The value is the conversion factor to be multiplied by the depth and origin time axes (sdepth and stime) of the confidence ellipse to recover the coverage ellipse without having to do a complete relocation.

Storage type	Unit	NA value	External	Nullable
float(24)		-999.0	f9.4	Y
Range	cov_depth_time > 0.0			

3.2.133. cov_sm_axes

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Coverage ellipse semi-axis conversion factor. The value is the conversion factor to be multiplied by the semi-major and semi-minor axes of the confidence ellipse to recover the coverage ellipse without having to do a complete relocation.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f9.4	Y
Range	cov_sm_axes > 0.0			

3.2.134. cp_broad_band

Table(s)	EVSC_HYDRO (IDCX)			
Description	The cep_peak_std_signal from table Hydro_features in the 2-80 Hz band for a given orid/sta pair.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	cp_broad_band > 0.0			

3.2.135. crnr_delfreq

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Estimated standard deviation in crnr_freq.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	–1.0	f7.2	Y
Range	crnr_delfreq > 0.0			

3.2.136. crnr_freq

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Corner frequency estimated from a windowed traditional FFT spectrum (rendered on a dB power scaling) applied to the traditional beam-formed waveform data.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	–1.0	f7.2	Y
Range	crnr_freq > 0.0			

3.2.137. ctype

Table(s)	SITECHAN (STATIC)			
Description	Channel type. This column specifies the type of data channel: n (normal, a normal instrument response), b (beam, a coherent beam formed with array data), or i (an incoherent beam or energy stack).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	Y
Range	ctype IN {n, b, i}			

3.2.138. cycle_size

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Size to which a subscription product should be allowed to grow before distributing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)	Kilobytes	-1	i8	Y
Range	cycle_size > 0			

3.2.139. cycle_time

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Period of cyclicity for a product in the Subscription Subsystem that is how often a product should be delivered.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)	Hours	-1	i4	Y
Range	cycle_time > 0			

3.2.140. data_info

Table(s)	DATAREADY (IDCX)			
Description	Miscellaneous data information for the Subscription Subsystem, such as the station name for station processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any character string up to 24 characters			

3.2.141. dataday

Table(s)	REBDONE_DATADAY_FLAG (LEB)			
Description	Julian day of this dataday.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(12,0)		NOT ALLOWED	i12	Y

Range	Julian dates are of the form yyyyddd

3.2.142. dataid

Table(s)	DATAREADY (IDCX)			
Description	Unique identifier for the DATAREADY table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	dataid >= 0			

3.2.143. dataid

Table(s)	PRODTRACK (IDCX)			
Description	Unique identifier for the dataready table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	dataid >= 0			

3.2.144. datatype

Table(s)	DERVDISC (IDCX)			
Description	Data type (bi: binary as:ASCII)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	datatype IN {bi, as}			

3.2.145. datatype

Table(s)	OLD_WFPROTO (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Data type of waveform segment			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	any valid data type in lowercase			

3.2.146. datatype

Table(s)	ARCH_DATA_TYPE (IDCX)			
Description	Data type (or class of data).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any uppercase string up to the character field width			

3.2.147. datsw

Table(s)	FKDISC (LEB)			
Description	A switch to indicate a data format data type. Proper values will be defined through the software libraries and include files. A manual (man) page will explain the meaning of datsw values and reference related software man pages.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	datsw > 0			

3.2.148. deast

Table(s)	SITE (STATIC)			
Description	Distance east. This column gives the "easting," or the relative position of an array element east of the location of the array centre specified by the value of refsta (see dnorth).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	kilometers	0.0	f9.4	Y
Range	-20,000.0 <= deast <= 20,000.0			

3.2.149. deflt

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i1	Y
Range				

3.2.150. delaz

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB)			
Description	Azimuth uncertainty. This column is an estimate of the standard deviation of the azimuth of a signal.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	delaz > 0.0			

3.2.151. deliv_meth

Table(s)	SUBS (IDCX)			
Description	Delivery method for subscription products.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(6)		- (hyphen)	a6	Y
Range	deliv_meth IN {ftp, email, www}			

3.2.152. delivid

Table(s)	PRODTRACK (IDCX)			
Description	Identifier that is unique and consecutive for each product constraint.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	delivid >= 0			

3.2.153. delivid

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Last delivery identifier for this product			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	delivid >= 0			

3.2.154. delslo

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Slowness uncertainty. This column is an estimate of the standard deviation of the slowness of a signal. Note that different units are used in ARRIVAL and DETECTION tables.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/degree	-1.0	f7.2	Y
Range	delslo > 0.0			

3.2.155. delslo

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	Slowness uncertainty. This column is an estimate of the standard deviation of the slowness of a signal. Note that different units are used in ARRIVAL and DETECTION tables.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/kilometer	-1.0	f7.2	Y
Range	delslo > 0.0			

3.2.156. delta

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), EVSC_REGIONAL (IDCX), PARRIVAL (LEB), PARRIVAL

	(REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Source-receiver distance in degrees. This column is the arc length, over the Earth's surface, of the path the seismic phase follows from source to receiver. The location of the origin is specified in the origin (origin_temp_ga) record referenced by the column orid. The column arid points to the record in the arrival table that identifies the receiver. The value of the column can exceed 360 degrees. The geographic distance between source and receiver is delta modulo(180).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f8.3	Y
Range	delta >= 0.0			

3.2.157. deltim

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	Arrival time uncertainty. This column is an estimate of the standard deviation of an arrival time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f6.3	Y
Range	deltim > 0.0			

3.2.158. depdp

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Depth as estimated from depth phases. The value is a measure of event depth estimated from a depth phase or an average of several depth phases. Depth is measured positive in a downwards direction starting from the earths surface (see ndp).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f9.4	Y
Range	0.0 <= depdp < 1000.0			

3.2.159. depth

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Source depth in kilometers. This column gives the depth (positive down) of the event origin. Negative depth implies an atmospheric event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f9.4	N

Range	-100.0 <= depth < 1000.0 0

3.2.160. depth

Table(s)	EVSC_PROD (IDCX)			
Description	Depth estimate in kilometers. This column gives the depth of the event origin. Must be >= 0. Atmospheric events, with negative depth are not allowed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f9.4	Y
Range	0 <= depth < 1000.0			

3.2.161. depth

Table(s)	STASSOC (IDCX)			
Description	Source depth in kilometers. This column gives the depth (positive down) of the event origin. Negative depth implies an atmospheric event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f9.4	Y
Range	-100.0 <= depth < 1000.0 0			

3.2.162. depth_conf

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Confidence level of a one-sided confidence interval of the source depth, used for the depth screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0 < depth_conf < 1.0			

3.2.163. depth_kvalue

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Depth uncertainty k-value, used to specify additional uncertainty in seismic free-depth solutions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f7.2	Y
Range	0.0 < depth_kvalue < 1000.0			

3.2.164. depth_thresh

Table(s)	PRODUCTTYPEEVSC (IDCX)
Description	Depth screening threshold. An event with a one-sided confidence interval for its source depth that is deeper than this threshold is screened out at the confidence level given by depth_conf.

Storage type	Unit	NA value	External	Nullable
float(24)	kilometers	-999.0	f7.2	Y
Range	0.0 < depth_thresh < 1000.0			

3.2.165. deptherr

Table(s)	EVSC_PROD (IDCX)			
Description	Depth error. This is the size of the one-sided depth confidence interval used for event screening for a confidence level given by depth_conf in table PRODUCTTYPEEVSC. A model uncertainty term, given by depth_kvalue in table PRODUCTTYPEEVSC, is included for free-depth solutions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f7.2	Y
Range	deptherr > 0.0			

3.2.166. dervid

Table(s)	DERVDISC (IDCX)			
Description	Derived data unique identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	dervid > 0			

3.2.167. dervtype

Table(s)	DERVDISC (IDCX)			
Description	Nature of the derived data (pix:PMCC detection pixels, fam: PMCC detection family)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	Y
Range	dervtype IN { fam, pix }			

3.2.168. descr

Table(s)	AMPDESCRIPT (IDCX)			
Description	Text description. Describes amplitude measurement parameters.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(255)		- (hyphen)	a255	Y
Range	any free-format string up to column size			

3.2.169. descrip

Table(s)	PROBLEM (IDCX)

Description	Text description.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(129)		- (hyphen)	a129	Y
Range	any free-format string up to the character limit			

3.2.170. descrip

Table(s)	SITECHAN (STATIC)			
Description	Text description.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)		- (hyphen)	a50	Y
Range	any free-format string up to the character limit			

3.2.171. detendtime

Table(s)	QCSTATS (IDCX)			
Description	End time of actual interval used for detection processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999.0	f17.5	Y
Range	detendtime < 9999999999.999			

3.2.172. dettime

Table(s)	QCSTATS (IDCX)			
Description	Start time of actual interval used for detection processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999.0	f17.5	Y
Range	dettime < 9999999999.999			

3.2.173. dfile

Table(s)	COLORDISC (MAP), MAPDISC (MAP), OVERLAYDISC (MAP)			
Description	Name of the data file. Must conform to UNIX file naming conventions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		NOT ALLOWED	a32	N
Range	any string up to the character limit that conforms to UNIX filename syntax			

3.2.174. dfile

Table(s)	FKDISC (LEB)
Description	Name of data file. Name of the heap file (fwfile). In fkdisc, the value is the file-name of an f-k disk file. In fsdisc, the value is the filename of a Fourier Spec-trum disk file (see dir), and so on.

Storage type	Unit	NA value	External	Nullable
varchar2(32)		NOT ALLOWED	a32	Y
Range	any string up to the character limit that conforms to UNIX filename syntax			

3.2.175. dfile

Table(s)	DERVDISC (IDCX), FILEPRODUCT (IDCX), FS_STAGEPRODUCT (IDCX), INSTRUMENT (STATIC), MSGDISC (IDCX), OLD_WFPROTO (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Name of the data file. Must conform to UNIX file naming conventions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		NOT ALLOWED	a32	Y
Range	any string up to the character limit that conforms to UNIX filename syntax			

3.2.176. dfile

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)				Y
Range				

3.2.177. dfile

Table(s)	PROBLEMLOG (IDCX)			
Description	Name of the data file. Must conform to UNIX file naming conventions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(65)		NOT ALLOWED	a65	Y
Range	any string up to the character limit that conforms to UNIX filename syntax			

3.2.178. digital

Table(s)	INSTRUMENT (STATIC)			
Description	Flag denoting whether this instrument record describes an analog (a) or digital (d) recording system.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	digital IN {d, a}			

3.2.179. dimx

Table(s)	MAPDISC (MAP)

Description	Width (or x-dimension) of the map in pixels.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)	pixels	NOT ALLOWED	i8	N
Range	dimx > 0			

3.2.180. dimy

Table(s)	MAPDISC (MAP)			
Description	Height (or y-dimension) of the map in pixels.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)	pixels	NOT ALLOWED	i8	N
Range	dimy > 0			

3.2.181. dir

Table(s)	PROBLEMLOG (IDCX)			
Description	Directory. This column is the directory part of a path name and must conform to UNIX directory naming conventions. Relative path names or "." (dot), the notation for the current directory, may be used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(129)		NOT ALLOWED	a129	Y
Range	any string that conforms to UNIX directory name syntax			

3.2.182. dir

Table(s)	COLORDISC (MAP), MAPDISC (MAP), OVERLAYDISC (MAP)			
Description	Directory. This column is the directory part of a path name and must conform to UNIX directory naming conventions. Relative path names or "." (dot), the notation for the current directory, may be used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	N
Range	any string that conforms to UNIX directory name syntax			

3.2.183. dir

Table(s)	DERVDISC (IDCX), FILEPRODUCT (IDCX), FS_STAGEPRODUCT (IDCX), INSTRUMENT (STATIC), MSGDISC (IDCX), OLD_WFPROTO (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Directory. This column is the directory part of a path name and must conform to UNIX directory naming conventions. Relative path names or "." (dot), the notation for the current directory, may be used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	Y

Range	any string that conforms to UNIX directory name syntax

3.2.184. dir

Table(s)	FKDISC (LEB)			
Description	Directory. This column is the directory part of a path name. Relative path names or "." (dot), the notation for the current directory, may be used. Directory where the heap file is located (fwfile). Directory to find file (msgdisc).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	Y
Range	any string that conforms to UNIX directory name syntax			

3.2.185. dir

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)				Y
Range				

3.2.186. dist

Table(s)	STASSOC (IDCX)			
Description	Estimated distance. This column gives the approximate source-receiver distance as calculated from slowness (array measurements only), incident angle, or (S-P) times.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= dist <= 180.0			

3.2.187. dlat

Table(s)	DSEISINDEX (STATIC), SEISINDEX (STATIC)			
Description	Latitude increment between grid cells in the related (d)seisgrid file			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	NOT ALLOWED	f9.4	Y
Range	dlat >= 0.0			

3.2.188. dlid

Table(s)	DLMAN (IDCX)			
Description	Diskloop manager identification.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number(10,0)		NOT ALLOWED	i10	N
Range	dlid > 0			

3.2.189. dlon

Table(s)	DSEISINDEX (STATIC), SEISINDEX (STATIC)			
Description	Longitude increment between grid cells in Seisgrid (Dseisgrid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	NOT ALLOWED	f9.4	Y
Range	dlon >= 0.0			

3.2.190. dmax

Table(s)	ATTENCOEF (STATIC)			
Description	Maximum distance for which the regional distance correction is applicable.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f7.2	Y
Range	0.0 < dmax < 2500.0			

3.2.191. dmin

Table(s)	ATTENCOEF (STATIC)			
Description	Minimum distance for which the regional distance correction is applicable.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f7.2	Y
Range	0.0 < dmin < 2500.0			

3.2.192. dnorth

Table(s)	SITE (STATIC)			
Description	Distance north. This column gives the "northing," or relative position of array element north of the array centre specified by the value of refsta (see deast).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	kilometers	0.0	f9.4	Y
Range	-20,000.0 <= dnorth <= 20,000.0			

3.2.193. dom_to_send

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Day of month to send product.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(2,0)		-1	i2	Y

Range	0 < dom_to_send <= 31

3.2.194. domain

Table(s)	DATAUSER (IDCX)			
Description	Domain name for a Subscription Subsystem user.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(48)		- (hyphen) NOT ALLOWED for datauser	a48	N
Range	any string up to 48 characters that is a valid internet domain			

3.2.195. domain

Table(s)	SUBSUSER (IDCX)			
Description	Domain name for a Subscription Subsystem user.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(48)		- (hyphen) NOT ALLOWED for datauser	a48	Y
Range	any string up to 48 characters that is a valid internet domain			

3.2.196. donetime

Table(s)	REBDONE_DATADAY_FLAG (LEB)			
Description	Epoch time that REB was completed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)		NOT ALLOWED	f17.5	Y
Range	any valid epoch time			

3.2.197. dow_to_send

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Day of week to deliver a product.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(2,0)		-1	i2	Y
Range	1 <= dow_to_send <= 7			

3.2.198. dropped

Table(s)	QCSTATS (IDCX)			
Description	Flag that indicates if the interval was dropped by DFX due to excessive masking.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number(8,0)		-1	i8	Y
Range	dropped IN {0, 1}			

3.2.199. dscore

Table(s)	EVSC_PROD (IDCX)			
Description	Score for the depth event-screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	dscore > -999.0			

3.2.200. dsize

Table(s)	FILEPRODUCT (IDCX)			
Description	Size of data file in bytes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)	bytes	NOT ALLOWED	i10	Y
Range	0 < dsize < 99,999,999			

3.2.201. dsize

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)				Y
Range				

3.2.202. dsize

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	Y
Range				

3.2.203. dtype

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)
Description	Depth determination flag. This single-character flag indicates the method by which the depth was determined or constrained during the location process. The recommended values are f (free), d (from depth phases), r (restrained by location program) or g (restrained by geophysicist). In cases r or g, either the auth

	column should indicate the agency or person responsible for this action, or the commid column should point to an explanation in the Remark table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	dtype IN {f, d, r, g}			

3.2.204. duration

Table(s)	PMCC_FEATURES (IDCX)			
Description	Duration of the PMCC family (seconds)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f8.3	Y
Range	duration > 0			

3.2.205. duration

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Total duration of amplitude window. Combined with start_time, the entire amplitude time window is specified. May also be employed to compute a coda duration magnitude if amp and per columns contain NA values.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	duration >= 0			

3.2.206. duration

Table(s)	CHAN_GROUPS (STATIC), CHAN_GROUPS_AUXWF (STATIC)			
Description	Not used. Always set to -1			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				Y
Range				

3.2.207. eavcep

Table(s)	TIMEFREQ (REB)			
Description	Average maximum value in the two-dimensional cepstrum of the east component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	eavcep >= 0			

3.2.208. eavcor

Table(s)	TIMEFREQ (REB)			
Description	Average autocorrelation along the time axis across all frequencies excluding randomized points of the east component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	eavcor >= 0			

3.2.209. eavpct

Table(s)	TIMEFREQ (REB)			
Description	Average ratio of bad points to total of the east component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	0 <= eavpct <= 1			

3.2.210. ec_consider

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				Y
Range				

3.2.211. ec_depthco

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.212. ec_deptherr

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.213. ec_depthp

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.214. ec_dscore

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.215. ec_grpname

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)				Y
Range				

3.2.216. ec_hscore

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.217. ec_icat

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)				Y
Range				

3.2.218. ec_ievcat

Table(s)	EVCHAR_PROD (IDCX)

Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)				Y
Range				

3.2.219. ec_irej

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)				Y
Range				

3.2.220. ec_iscore

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.221. ec_ldpflg

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)				Y
Range				

3.2.222. ec_lscore

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.223. ec_mbmscal

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(24)				Y
Range				

3.2.224. ec_mbmsco

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.225. ec_mbmsfac

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.226. ec_mbmsns

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				Y
Range				

3.2.227. ec_mbmsp

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.228. ec_mscore

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.229. ec_msmb

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.230. ec_msmberr

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.231. ec_ndef

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				Y
Range				

3.2.232. ec_nsta_mb

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				Y
Range				

3.2.233. ec_nsta_ms

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				Y
Range				

3.2.234. ec_onshorep

Table(s)	PRODUCTTYPEEVPARS (IDCX)

Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.235. ec_pctoffsh

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.236. ec_popcrit

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(60)				Y
Range				

3.2.237. ec_popmagc

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.238. ec_popmint

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				Y
Range				

3.2.239. ec_popsimp

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(24)				Y
Range				

3.2.240. ec_popsnco

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.241. ec_pscore

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.242. ec_pvaluemin

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.243. ec_regname

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)				Y
Range				

3.2.244. ec_score

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.245. ec_smajax

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.246. ec_sminax

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.247. ec_stamin

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)				Y
Range				

3.2.248. ec_strike

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)				Y
Range				

3.2.249. ec_udat

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)				Y
Range				

3.2.250. edepth

Table(s)	SITECHAN (STATIC)

Description	Emplacement depth. This column gives the depth at which the instrument is positioned, relative to the value of elev in the Site table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	kilometers	NOT ALLOWED	f9.4	Y
Range	edepth >= 0.0			

3.2.251. elev

Table(s)	SITE (STATIC)			
Description	Surface elevation. This column is the elevation of the surface of the earth above a seismic station relative to mean sea level.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	kilometers	-999.0	f9.4	Y
Range	-10.0 <= elev <= 10.0			

3.2.252. elevation_angle

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.253. ema

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Emergence angle. This column is the emergence angle of an arrival, as observed at a three-component station or array. The value increases from the vertical direction towards the horizontal.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= ema <= 90.0			

3.2.254. emaillimit

Table(s)	DATAUSER (IDCX)			
Description	Maximum size of message that will be delivered via email in the Message Subsystem.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)	bytes	-1	i8	Y
Range	emaillimit >= 0			

3.2.255. emailto

Table(s)	MSGDEST (IDCX)			
Description	Destination email address.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		- (hyphen)	a64	Y
Range	any string up to 64 characters			

3.2.256. emares

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Emergence angle residual. This column is the difference between an observed emergence angle and the theoretical prediction for the same phase, assuming an event location as specified by the accompanying orid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f7.1	Y
Range	-90.0 <= emares <= 90.0			

3.2.257. end_date

Table(s)	PARTICIPATION (STATIC)			
Description	Date a station was discontinued from a network.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	any valid Julian date where end_date > begin_date (yyyyddd)			

3.2.258. end_time

Table(s)	REQUEST (IDCX)			
Description	Ending time of comparison or processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	N
Range	any valid epoch time			

3.2.259. endtime

Table(s)	BEAMAUX (STATIC), SENSOR (STATIC)			
Description	End epochal time given as seconds and fractions of a second since 00:00:00.000 01-Jan-1970; stored in a double-precision floating number. This represents the endtime of the data segment.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	9999999999.999	f17.5	N

Range	endtime < +9999999999.999

3.2.260. endtime

Table(s)	OLD_WFPROTO (IDCX)			
Description	End epochal time given as seconds and fractions of a second since 00:00:00.000 01-Jan-1970; stored in a double-precision floating number. This represents the endtime of the data segment.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	N
Range	endtime < +9999999999.999			

3.2.261. endtime

Table(s)	FILEPRODUCT (IDCX), INTERVAL (IDCX), INTERVAL (SEGMENT), OUTAGE (IDCX), QCSTATS (IDCX)			
Description	End epochal time given as seconds and fractions of a second since 00:00:00.000 01-Jan-1970; stored in a double-precision floating number. This represents the endtime of the data segment.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	9999999999.999	f17.5	Y
Range	endtime < +9999999999.999			

3.2.262. endtime

Table(s)	WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	End epochal time given as seconds and fractions of a second since 00:00:00.000 01-Jan-1970; stored in a double-precision floating number. This represents the endtime of the data segment.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	Y
Range	endtime < +9999999999.999			

3.2.263. endtime

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	Epoch time. Epochal time is given as seconds and fractions of a second since 00:00:00.000 01-Jan-1970, and is stored in a double-precision floating number. This represents the endtime of the data segment.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds		f17.5	Y
Range	endtime < +9999999999.999			

3.2.264. endtime

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)				Y
Range				

3.2.265. endtime

Table(s)	PROBLEM (IDCX)			
Description	End epochal time given as seconds and fractions of a second since 00:00:00.000 01-Jan-1970; stored in a double-precision floating number. This represents the endtime of the data segment.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(17,5)	seconds	9999999999.999	f17.5	Y
Range	endtime < +9999999999.999			

3.2.266. eng_deldur

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Estimated standard deviation in eng_dur.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f7.2	Y
Range	eng_deldur > 0.0			

3.2.267. eng_dur

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Time period during which the STA/LTA on the traditional beam exceeded its threshold for the current time interval.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	eng_dur > 0.0			

3.2.268. eng_time

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Earliest time at which the STA/LTA applied to the traditional beam exceeded its threshold for the current time interval.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(53)	seconds	–9999999999.999	f17.5	Y
Range	eng_time > –9999999999.999			

3.2.269. esaz

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Event-to-station azimuth measured in degrees clockwise from North.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f7.2	Y
Range	0.0 <= esaz <= 360.0			

3.2.270. etype

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), STASSOC (IDCX)			
Description	Event type. Describes the type of event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(7)		- (hyphen)	a7	Y
Range	etype IN {QUAKE, BLAST, EXPL, IND}			

3.2.271. evid

Table(s)	DISCARD (LEB), DISCARD (REB)			
Description	Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the origin table can have the same evid. this can accommodate opinions about the location of the event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	evid > 0			

3.2.272. evid

Table(s)	EVENT (LEB), EVENT (REB), EVENT (SEL1), EVENT (SEL2), EVENT (SEL3)			
Description	Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the origin table can have the same evid. this can accommodate opinions about the location of the event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	evid > 0			

3.2.273. evid

Table(s)	IN_EVENT_CONTROL (LEB)			
Description	Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the origin table can have the same evid. Analyst have several opinions about the location of the event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1 NOT ALLOWED for event	i10	Y
Range	evid > 0			

3.2.274. evid

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3), REQUEST (IDCX), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the origin table can have the same evid. this can accommodate opinions about the location of the event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	evid > 0			

3.2.275. evname

Table(s)	EVENT (LEB), EVENT (REB), EVENT (SEL1), EVENT (SEL2), EVENT (SEL3)			
Description	Event name. This is the common name of the event identified by evid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any free-format string up to 15 characters			

3.2.276. extern_chan

Table(s)	CHANNAME (STATIC), NOOPS_CHANNAME (STATIC)			
Description	The channel name used by the sending station.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	any string up to the character field width			

3.2.277. *extern_sta*

Table(s)	CHANNAME (STATIC), NOOPS_CHANNAME (STATIC)			
Description	The site name used by the sending station.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any upper-case string up to the character field width			

3.2.278. *extmsgid*

Table(s)	MSGDISC (IDCX)			
Description	Value of the msgid column in a message that is received by the message system.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		-1	a20	Y
Range	any string up to 20 characters			

3.2.279. *famsize*

Table(s)	PMCC_FEATURES (IDCX)			
Description	Number of detection pixels in the PMCC family			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	Y
Range				

3.2.280. *fgroup*

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	Y
Range				

3.2.281. *fileoff*

Table(s)	MSGDISC (IDCX)			
Description	Number of bytes to the first character of the email file (first character of the email header). fileoff will always be 0 on the operations system, but will be reset when archived.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)	bytes	-1	i10	Y
Range	fileoff > 0			

3.2.282. *filesize*

Table(s)	MSGDISC (IDCX)			
Description	Size of file.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)	bytes	-1	i10	Y
Range	fileoff > 0			

3.2.283. *fkid*

Table(s)	FKDISC (LEB)			
Description	Uniquely identifies a f-k spectrum file.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	fkid > 0			

3.2.284. *fkqual*

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	An integer quantifying the quality of the f-k spectrum. An fkqual = 1 is high quality; an fkqual = 4 is low quality.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		-1	f7.6	Y
Range	1 <= fkqual <= 4			

3.2.285. *fkrid*

Table(s)	FKDISC (LEB)			
Description	Uniquely defines a f-k spectrum recipe.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	fkrid > 0			

3.2.286. *fktyp*

Table(s)	FKDISC (LEB)			
Description	String that identifies the type of f-k spectrum; examples are monochromatic (mono) and broadband (broa).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		NOT ALLOWED	a4	Y
Range	any lower-case string up to four characters			

3.2.287. fm

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	First motion. This is a two-character indication of first motion. The first character describes first motion seen on short-period channels and the second holds for long-period instruments. Compression on a short-period sensor is denoted by c, dilatation by a d; and compression on a long-period sensor is denoted by u, dilatation by an r. Empty character positions will be indicated by dots (for example, ".r" for dilatation on a long-period sensor).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	all two-letter permutations of {c | d | .}, {u | r | .}			

3.2.288. fmax

Table(s)	SPVAR (REB)			
Description	Maximum frequency of a band with snr > 3 dB used for the spectral variance calculation.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	NOT ALLOWED	f7.2	N
Range	fmax > fmin			

3.2.289. fmin

Table(s)	SPVAR (REB)			
Description	Minimum frequency of a band with snr > 3 dB used for the spectral variance calculation.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	NOT ALLOWED	f7.2	N
Range	0 < fmin < fmax			

3.2.290. foff

Table(s)	OLD_WFPROTO (IDCX)			
Description	File offset; the byte offset of a data segment within a physical data file. This column is nonzero if the data reference does not occur at the beginning of the file.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	bytes	NOT ALLOWED	i10	Y
Range	foff >= 0			

3.2.291. foff

Table(s)	DERVDISC (IDCX), FILEPRODUCT (IDCX), FKDISC (LEB), MSGDISC (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEG-

	MENT)			
Description	File offset; the byte offset of a data segment within a physical data file. This column is nonzero if the data reference does not occur at the beginning of the file.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)	bytes	NOT ALLOWED	i10	Y
Range	foff >= 0			

3.2.292. foff

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)				Y
Range				

3.2.293. foff

Table(s)	MSGDATATYPE (IDCX)			
Description	File offset; the byte offset of a data segment within a physical data file. This column is nonzero if the data reference does not occur at the beginning of the file.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)	bytes	NOT ALLOWED	i8	N
Range	foff >= 0			

3.2.294. foff

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	Y
Range				

3.2.295. ford

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Filter order. In infra_features, this column identifies the order of the Infinite Impulse Response (IIR) filter used to extract the spectral band in which the standard fk spectrum calculation was performed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		0	i8	Y

Range	ford > 0

3.2.296. ford

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Filter order. In infra_features, this column identifies the order of the Infinite Impulse Response (IIR) filter used to extract the spectral band in which the standard fk spectrum calculation was performed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		0	i8	Y
Range	ford > 0			

3.2.297. forwardport

Table(s)	DLMAN (IDCX)			
Description	Forwarder port used by the diskloop manager.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(6,0)		0	i6	Y
Range	1 <= forwardport <= 16383			

3.2.298. fpid

Table(s)	FILEPRODUCT (IDCX)			
Description	File product identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	fpid > 0			

3.2.299. fpid

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	Y
Range				

3.2.300. fpid

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)				Y

Range	

3.2.301. freq

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Centre frequency of the wideband polarization analysis. For example, if only the 2 - 4 Hz and 4 - 8 Hz bands satisfy the signal-to-noise ratio criterion, then freq is set to 5.0 Hz.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f7.2	Y
Range	freq > 0.0			

3.2.302. freq_tol

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.303. fsid

Table(s)	SPVAR (REB)			
Description	Fourier spectrum identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	fsid > 0			

3.2.304. fstat

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	F-statistic; a measure of the signal-to-noise ratio at the peak in the f-k spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f8.5	Y
Range	fstat >= 0			

3.2.305. ftp_address

Table(s)	FTPLOGIN (IDCX)			
Description	FTP address of site from which IDC is attempting to transfer data message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	N
Range	any string up to 64 characters			

3.2.306. ftp_address

Table(s)	FTPFAILED (IDCX)			
Description	FTP address of site from which IDC is attempting to transfer data message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	Y
Range	any string up to 64 characters			

3.2.307. ftype

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Filter type. This indicates the type of filtering that was performed for a Butterworth filter. "BP" indicates a band-pass filter between low_cut and high_cut; "LP" indicates a low-pass filter below high_cut; "HP" indicates a high-pass filter above low_cut; "BR" indicates a band-reject filter outside low_cut and high_cut.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	ftype IN {BP, LP, HP, BR}			

3.2.308. fzp

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Filter causality. This flag indicates zero-phase filtering. If it is 1, the filter is zero-phase (noncausal); if it is 0, the filter is not zero-phase (causal).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i1	Y
Range	fzp IN {0,1}			

3.2.309. gctp1

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.310. gctp10

Table(s)	MAPDISC (MAP)
Description	General cartographic transformation package variable. The default for all values is 0.

Storage type	Unit	NA value	External	Nullable
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.311. gctp11

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.312. gctp12

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.313. gctp13

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.314. gctp14

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.315. gctp15

Table(s)	MAPDISC (MAP)

Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.316. gctp2

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.317. gctp3

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.318. gctp4

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.319. gctp5

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.320. gctp6

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.321. gctp7

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.322. gctp8

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.323. gctp9

Table(s)	MAPDISC (MAP)			
Description	General cartographic transformation package variable. The default for all values is 0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.4	Y
Range				

3.2.324. grdname

Table(s)	DSEISGRID (STATIC), DSEISINDEX (STATIC), SEISGRID (STATIC), SEISINDEX (STATIC)			
Description	Name for identifying the basis of a natural seismicity grid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N

Range	any free-format string up to six characters

3.2.325. grn

Table(s)	GREGION (STATIC)			
Description	Geographic region number, as defined by [Fli74].			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	N
Range	grn > 0			

3.2.326. grn

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Geographic region number, as defined by [Fli74].			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	grn > 0			

3.2.327. grname

Table(s)	GREGION (STATIC)			
Description	Geographic region name. This column is the common name of a geographic region, as given in [Fli74]. Names may have changed due to changing political circumstances (for example, old RHODESIA = new ZIMBABWE) (see grn and srname).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)		NOT ALLOWED	a40	Y
Range	any upper-case string up to 40 characters			

3.2.328. grpname

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Group name for subscriptions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any valid string up to 24 characters			

3.2.329. gvhi

Table(s)	AMPDESCRIPT (IDCX)
Description	High group velocity for determining a time window. This column defines the start time of an amplitude measurement window if toff is null. If gvhi is used,

	then gvlo must be used to define the end time of the window.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers/second	-999.0	f5.2	Y
Range	gvhi > gvlo			

3.2.330. gvlo

Table(s)	AMPDESCRIPT (IDCX)			
Description	Low group velocity for determining a time window. This column defines the end time of an amplitude measurement window if tlen is null or if gvhi is used to define the start time of the window.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers/second	-999.0	f5.2	Y
Range	gvlo > 0			

3.2.331. hamp

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB)			
Description	Horizontal amplitude. Absolute maximum amplitude (nm) measured on the root of the sum of the squares of two horizontally-oriented components filtered in a frequency band centred at cfreq Hz.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers	-999.0	f11.2	Y
Range	hamp >= 0.0			

3.2.332. hang

Table(s)	SITECHAN (STATIC)			
Description	Horizontal orientation of seismometer. This column specifies the orientation of the seismometer in the horizontal plane, measured clockwise from North. For a North-South orientation with the seismometer pointing toward the north, hang = 0.0; for East-West orientation with the seismometer pointing toward the west, hang = 270.0 (see vang).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees	NOT ALLOWED	f6.1	Y
Range	0.0 <= hang < 360.0			

3.2.333. header_fpid

Table(s)	FPDESCRIPTION (IDCX)			
Description	Fpid pointing to the header row for this product type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	header_fpid > 0			

3.2.334. high_cut

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	High-cut frequency of filter.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f9.4	N
Range	0.0 <= high_cut <= sample rate/2			

3.2.335. hmxmn

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Maximum-to-minimum horizontal amplitude ratio defined as (l1/l2)1/2 where l1 and l2 are the maximum and minimum eigenvalues obtained by solving the 2-D eigensystem using only the horizontal components. This S-type value is calculated at the time of maximum 3-component amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	hmxmn >= 0.0			

3.2.336. hour

Table(s)	ALLOCATE_HOUR (LEB), ALLOCATE_HOUR (REB)			
Description	Starting hour of a time block for interactive analysis.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(2,0)	hours	NOT ALLOWED	i2	N
Range	0 <= hour <= 23			

3.2.337. hour_to_send

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Hour at which the product is to be sent.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(2,0)	hours	NOT ALLOWED	i2	Y
Range	0 <= hours_to_send <= 23			

3.2.338. hscore

Table(s)	EVSC_PROD (IDCX)			
Description	Score for the hydroacoustic event-screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	hscore > -999.0			

3.2.339. hsnr

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB)			
Description	Horizontal signal-to-noise ratio. Ratio of hamp to the root-mean-square amplitude of the root of the sum of the squares of two horizontally oriented components filtered in a frequency band centred at cfreq Hz.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f10.2	Y
Range	hsnr >= 0.0			

3.2.340. htov

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB)			
Description	Horizontal-to-vertical power ratio. One-half times the square of the ratio of hamp to vamp.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f10.2	Y
Range	htov >= 0.0			

3.2.341. hvrat

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Horizontal-to-vertical power ratio defined as (C3 + C2)/2C1 where C1, C2, and C3 are the diagonal elements of the covariance matrix (C1 corresponds to the vertical component). This is an S-phase-type value that is calculated at the time of the maximum 3-component amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	hvrat >= 0.0			

3.2.342. hvratp

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Horizontal-to-vertical power ratio defined as (C3 + C2)/2C1 where C1, C2, and C3 are the diagonal elements of the covariance matrix (C1 corresponds to the vertical component). This is a P-phase-type value that is calculated at the time of maximum rectilinearity.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	hvratp >= 0.0			

3.2.343. hyd_grp_phase

Table(s)	HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB)

Description	Phase name resulting from hydro group processing by the application HASE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a4	Y
Range	Any valid hydro or seismic phase name. currently hyd_grp_phase IN {H, T, N, P}			

3.2.344. hydro_cp_thresh

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Threshold of the hydroacoustic cepstral peak screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	dB re microPascals	-999.0	f7.2	Y
Range	0.0 < hydro_cp_thresh < 999.0			

3.2.345. hydro_id

Table(s)	HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB), HYDRO_ASSOC (IDCX), HYDRO_ASSOC (LEB), HYDRO_ASSOC (REB)			
Description	Identifier which is the primary key in the hydro_arr_group table and the foreign key in hydro_assoc.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	hydro_id > 0			

3.2.346. hydro_te_thresh

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Threshold of the hydroacoustic total energy screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	dB re microPascals	-999.0	f7.2	Y
Range	0.0 < hydro_te_thresh < 999.0			

3.2.347. icell

Table(s)	DSEISGRID (STATIC), SEISGRID (STATIC)			
Description	Grid cell index.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	icell > 0			

3.2.348. id

Table(s)	GA_TAG (SEL1)			
Description	Arrival or origin identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	id > 0			

3.2.349. idate

Table(s)	MSGDISC (IDCX)			
Description	Initial Julian date message that was received.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	any valid Julian date			

3.2.350. imb

Table(s)	STASSOC (IDCX)			
Description	Initial body wave magnitude. This column is an analysts estimate of the body wave magnitude using data from a single station (see iml, ims, magnitude, magtype, mb, ml, and ms).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	imb > -2.0			

3.2.351. imethod

Table(s)	MSGDISC (IDCX)			
Description	Method of transmission for a message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	imethod IN {email, ftp}			

3.2.352. iml

Table(s)	STASSOC (IDCX)			
Description	Initial local magnitude. This column is an analysts estimate of the local using data from a single station (see imb, ims, magnitude, magtype, mb, ml, and ms).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	iml > -2.0			

3.2.353. ims

Table(s)	STASSOC (IDCX)			
Description	Initial surface wave magnitude. The value in this column is an analysts estimate of surface wave magnitude using data from a single station (see iml, imb, magnitude, magtype, mb, ml, and ms).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	ims > -2.0			

3.2.354. inang1

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Apparent incidence angle (measured from the vertical) of the eigenvector (e1) associated with the largest eigenvalue (l1). This column is also called the long-axis incidence angle or the emergence angle. This P-type value is calculated at the time of maximum rectilinearity.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= inang1 < 90.0			

3.2.355. inang3

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Apparent incidence angle (measured from the vertical) of the eigenvector (e3) associated with the smallest eigenvalue (l3). This column is also called the short-axis incidence angle. This S-type value is measured at the time of the maximum 3-component amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f7.2	Y
Range	0.0 <= inang3 < 90.0			

3.2.356. inarrival

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Flag to indicate whether or not amp is the same as it is in the ARRIVAL table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		NOT ALLOWED	a1	Y
Range	inarrival IN {y, n}			

3.2.357. inauth

Table(s)	WFCONV (IDCX)
Description	Flag showing if input data are authenticated.

Storage type	Unit	NA value	External	Nullable
varchar2(1)		- (hyphen)	a1	Y
Range	inauth IN {y, n}			

3.2.358. incomp

Table(s)	WFCONV (IDCX)			
Description	Input data compression type. The only type currently supported is Canadian compression (CA).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	incomp IN {CA}			

3.2.359. inid

Table(s)	INSTRUMENT (STATIC), SENSOR (STATIC)			
Description	Instrument identifier. This column is a unique key to the instrument table and inid provides the only link between sensor and instrument.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED -1 for sensor.	i8	N
Range	inid > 0			

3.2.360. initial_time

Table(s)	PMCC_FEATURES (IDCX)			
Description	Time of the first detection pixel in the PMCC family (epochal)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds		f17.5	Y
Range				

3.2.361. initialdate

Table(s)	SUBS (IDCX)			
Description	Initial date of a subscription.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	ORACLE Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

3.2.362. insamp

Table(s)	WFCONV (IDCX)
Description	Number of input samples per packet.

Storage type	Unit	NA value	External	Nullable
number(8,0)		0	i8	Y
Range	insamp > 0			

3.2.363. insname

Table(s)	INSTRUMENT (STATIC)			
Description	Instrument name. This character string contains the name of the instrument.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)		- (hyphen)	a50	Y
Range	any free-format string up to 50 characters			

3.2.364. instant

Table(s)	SENSOR (STATIC)			
Description	Snapshot indicator. When instant = y, the snapshot was taken at the time of a discrete procedural change, such as an adjustment of the instrument gain; when instant = n, the snapshot is of a continuously changing process, such as calibration drift. This value is important for tracking time corrections and calibrations. The default value is y.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		NOT ALLOWED	a1	N
Range	instant IN {y, n}			

3.2.365. instype

Table(s)	INSTRUMENT (STATIC), OLD_WFPROTO (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Instrument type. This character string is used to indicate the instrument type. Some examples are: W200P, A100L2, MB2000, PTB101 and CMG3ES.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen)	a6	Y
Range	any alphanumeric string up to the character field width			

3.2.366. intern_chan

Table(s)	CHANNAME (STATIC)			
Description	The channel name used by CD2WNG. Links to the chan column in the SENSOR and WFDISC tables.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	any lower-case string up to the character field width			

3.2.367. intern_chan

Table(s)	NOOPS_CHANNAME (STATIC)			
Description	The name of the channel as chosen by the data consumer. The translation is from extern_chan to intern_chan.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	any lower-case string up to the character field width			

3.2.368. intern_sta

Table(s)	CHANNAME (STATIC)			
Description	The site name used by CD2WNG. Links to the sta column in the SENSOR and WFDISC tables.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any upper-case string up the character field width			

3.2.369. intern_sta

Table(s)	NOOPS_CHANNAME (STATIC)			
Description	The site name used by cd2w. Links to the sta column in the sensor and wfdisc tables. If the waveSaveFormat parameter is SAVE_SEPARATE, then this name is also used in the waveform file names (e.g. I26H3.BDF.20080223.20-24.w).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any upper-case string up the character field width			

3.2.370. intid

Table(s)	MSGDISC (IDCX), SUBS (IDCX)			
Description	Either the locally generated msgid of an earlier table entry that evoked the creation of this table entry or the reqid from the request table of an internally generated request			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	intid > 0			

3.2.371. intidtype

Table(s)	MSGDISC (IDCX)
Description	Identifier type for the intid (usually msgid,but can be reqid, delivid, wmreq_msgid). Set to - (hyphen) for incoming radionuclide data messages and VDMS requests.

Storage type	Unit	NA value	External	Nullable
varchar2(16)		- (hyphen)	a16	Y
Range	intidtype IN { msgid, reqid, delivid, wmreq_msgid }			

3.2.372. intidtype

Table(s)	SUBS (IDCX)			
Description	Message identification number (msgid), linking back to the entry in the msgdisc table that initiated the subscription request.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	Y
Range	intidtype IN { msgid }			

3.2.373. intvlid

Table(s)	INTERVAL (IDCX)			
Description	Interval identifier. Each interval is assigned a unique positive integer that identifies it in the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	intvlid > 0			

3.2.374. intvlid

Table(s)	INTERVAL (SEGMENT)			
Description	Interval identifier. Each interval is assigned a unique positive integer that identifies it in the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i10	N
Range	intvlid > 0			

3.2.375. intype

Table(s)	WFCONV (IDCX)			
Description	Input fixed width datatype.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	same as datatype			

3.2.376. inwfactivity

Table(s)	CHAN_GROUPS (STATIC), CHAN_GROUPS_AUXWF (STATIC)
Description	Not used. Always set to -1

Storage type	Unit	NA value	External	Nullable
number(1,0)				Y
Range				

3.2.377. iphase

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Reported phase. This eight-character column holds the name initially given to a seismic phase. Standard seismological labels for the types of signals (or phases) are used (for example, P, PKP, PcP, pP). Both upper- and lower-case letters are available and should be used when appropriate, for example, pP or PcP (see phase).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	any string up to eight characters that conforms to seismological practice			

3.2.378. is_coda

Table(s)	PHASE_DESCRIPTION (STATIC)			
Description	Flag to specify if phase is a coda phase (y) or not (n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	N
Range	is_coda IN { y, n }			

3.2.379. is_depth_phase

Table(s)	PHASE_DESCRIPTION (STATIC)			
Description	Flag to specify if phase is a depth phase (y) or not (n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	N
Range	is_depth_phase IN { y, n }			

3.2.380. is_depth_sensitive

Table(s)	PHASE_DESCRIPTION (STATIC)			
Description	Flag to specify if phase is depth sensitive (y) or not (n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	N
Range	is_depth_sensitive IN { y, n }			

3.2.381. is_selected

Table(s)	PHASE_DESCRIPTION (STATIC)

Description	Flag to specify if phase is selected (y) or not (n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	N
Range	is_selected IN { y, n }			

3.2.382. is_timeavail

Table(s)	PHASE_DESCRIPTION (STATIC)			
Description	Flag to specify whether travel time information is available (y) or not (n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	N
Range	is_timeavail IN { y, n }			

3.2.383. is_unique

Table(s)	PHASE_DESCRIPTION (STATIC)			
Description	Flag to specify if phase is a coda phase (y) or not (n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	N
Range	is_unique IN { y, n }			

3.2.384. isrc

Table(s)	MSGDISC (IDCX)			
Description	Initial source of message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		- (hyphen)	a64	Y
Range	any string up to 64 characters			

3.2.385. itime

Table(s)	MSGDEST (IDCX)			
Description	Initial time when message was inserted into the MSGDEST table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.386. itime

Table(s)	MSGDISC (IDCX)
Description	Initial time when message was inserted into the msgdisc table by the MessageReceive application.

Storage type	Unit	NA value	External	Nullable
float(53)	seconds	-9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.387. jdate

Table(s)	ALLOCATE_HOUR (LEB), ALLOCATE_HOUR (REB)			
Description	Julian date; date of an arrival, origin, seismic recording, and so on. The same information is available in epoch time, but the Julian date format is more convenient for many types of searches. Dates B.C. are negative. The year will never equal 0000, and the day will never equal 000. Where only the year is known, the day of the year is 001; where only year and month are known, the day of year is the first day of the month. Only the year is negated for B.C., so 1 January of 10 B.C. is -0010001 (see time). A jdate begins at midnight and ends just before midnight the following day. The jdate must be consistent with the time in the same record (if specified).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	N
Range	Julian dates are of the form yyyyddd			

3.2.388. jdate

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), DERVDISC (IDCX), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), FKDISC (LEB), OLD_WFPROTO (IDCX), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), QCSTATS (IDCX), SENSOR (STATIC), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Julian date; date of an arrival, origin, seismic recording, and so on. The same information is available in epoch time, but the Julian date format is more convenient for many types of searches. Dates B.C. are negative. The year will never equal 0000, and the day will never equal 000. Where only the year is known, the day of the year is 001; where only year and month are known, the day of year is the first day of the month. Only the year is negated for B.C., so 1 January of 10 B.C. is -0010001 (see time). A jdate begins at midnight and ends just before midnight the following day. The jdate must be consistent with the time in the same record (if specified).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	Julian dates are of the form yyyyddd			

3.2.389. keyname

Table(s)	LASTID (IDCX), LASTID (SEGMENT), PROBLASTID (IDCX), RMS_LASTID (IDCX)
Description	Identifier type. This column contains the actual name of a key whose last assigned numeric value is saved in keyvalue.

Storage type	Unit	NA value	External	Nullable
varchar2(15)		NOT ALLOWED	a15	N
Range	keyname IN {arid, chanid, commid, evid, inid, orid, stassid, wfid, dataid, prodid, subsid, and so on}			

3.2.390. keyvalue

Table(s)	LASTID (IDCX), LASTID (SEGMENT), PROBLASTID (IDCX), RMS_LASTID (IDCX)			
Description	Current identifier value. This column maintains the last assigned value (a positive integer) of the counter for the specified keyname. The keyvalue is the last counter value used for the column keyname. Key values are maintained in the database to ensure uniqueness.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	keyvalue > 0			

3.2.391. kurtosis

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Kurtosis of the estimated signal energy between onset_time and termination_time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-9999999999999	f9.4	Y
Range	-10000000.0 < kurtosis< 10000000.0			

3.2.392. label

Table(s)	MAPDISC (MAP)			
Description	Header for Map listing. A label, such as world, categorizes each Map. Label is used to build a sorted list of maps in the Map program.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(65)		- (hyphen)	a65	Y
Range	any string up to 65 characters			

3.2.393. lastfailedtime

Table(s)	FTPFAILED (IDCX)			
Description	Time of most recent attempt to retrieve data message by FTP.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)		NOT ALLOWED	f17.5	Y
Range	any valid epoch time			

3.2.394. lat

Table(s)	MAPPOINT (MAP), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Geographic latitude. Locations north of the equator have positive latitudes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f9.4	N
Range	-90.0 <= lat <= 90.0			

3.2.395. lat

Table(s)	STASSOC (IDCX)			
Description	Geographic latitude. Locations north of the equator have positive latitudes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f9.4	Y
Range	-90.0 <= lat <= 90.0			

3.2.396. lat

Table(s)	SITE (STATIC)			
Description	Geographic latitude. Locations north of the equator have positive latitudes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees	-999.0	f9.4	Y
Range	-90.0 <= lat <= 90.0			

3.2.397. lat1

Table(s)	DSEISINDEX (STATIC), SEISINDEX (STATIC)			
Description	Southern-most latitude of the first grid cell in Seisgrid (Dseisgrid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	NOT ALLOWED	f9.4	Y
Range	-90.0 <= lat1 <= 90.0			

3.2.398. lat_hi

Table(s)	BREGION_DEF (STATIC)			
Description	Maximum latitude of the region. Locations north of the equator have positive latitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	-90.0 <= lat_hi <= 90.0			

3.2.399. lat_lo

Table(s)	BREGION_DEF (STATIC)			
Description	Minimum latitude of the region. Locations north of the equator have positive latitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	-90.0 <= lat_lo <= 90.0			

3.2.400. latmajor

Table(s)	MAPDISC (MAP)			
Description	Latitude interval for displaying major grid lines in the Map application.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999	f9.4	Y
Range	0.0 < latmajor < 90.0			

3.2.401. latminor

Table(s)	MAPDISC (MAP)			
Description	Latitude interval for displaying minor grid lines in the Map application.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999	f9.4	Y
Range	0 < latminor < 90.0			

3.2.402. latorigradians

Table(s)	MAPDISC (MAP)			
Description	Latitude origin radians. Coordinates in radians of the lower left corner in the Map application. The Map application uses this for mercator projections only.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	radians	-999	f9.4	Y
Range	-pi/2 < latorigradians < pi/2			

3.2.403. lddate

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC), DISCARD (LEB), DISCARD (REB), PHASE_DESCRIPTION (STATIC), PHASE_WEIGHT_EVENT_DEFINE (STATIC), PHASOR_DIST_DEPTH_RANGES (STATIC), STATION_PHASE_RELY (STATIC), STA_RELY (STATIC)			
Description	Load date. The date and time the record was inserted into the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	ORACLE Date	NOT ALLOWED	a21	N

Range	any valid ORACLE date

3.2.404. lddate

Table(s)	AFFILIATION (STATIC), AFFILIATION_EXCL (STATIC), AFFILIATION_LP (STATIC), AFFILIATION_SUB (STATIC), ALLOCATE_HOUR (LEB), ALLOCATE_HOUR (REB), ALLOW_RESID (LEB), AMP3C (IDCX), AMP3C (LEB), AMP3C (REB), AMPDESCRIPT (IDCX), AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), APMA (IDCX), APMA (LEB), APMA (REB), ARCH_DATA_TYPE (IDCX), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), ATTENCOEF (STATIC), BREGION_DEF (STATIC), BREGION_TAB (STATIC), CEPPKS (REB), CHANNAME (STATIC), CHAN_GROUPS (STATIC), CHAN_GROUPS_AUXWF (STATIC), COLORDISC (MAP), COMPLEXITY (REB), DATAREADY (IDCX), DATAUSER (IDCX), DERVDISC (IDCX), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), DLMAN (IDCX), DSEISGRID (STATIC), DSEISINDEX (STATIC), EVCHAR_PROD (IDCX), EVENT (LEB), EVENT (REB), EVENT (SEL1), EVENT (SEL2), EVENT (SEL3), EVENT_CONTROL (LEB), EVENT_CONTROL (REB), EVSC_HYDRO (IDCX), EVSC_PROD (IDCX), EVSC_REGIONAL (IDCX), FILEPRODUCT (IDCX), FKDISC (LEB), FPDESCRIPTION (IDCX), FS_STAGEPRODUCT (IDCX), FTPFAILED (IDCX), FTPLOGIN (IDCX), GREGION (STATIC), HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB), HYDRO_ASSOC (IDCX), HYDRO_ASSOC (LEB), HYDRO_ASSOC (REB), HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB), INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB), INSTRUMENT (STATIC), INTERVAL (IDCX), INTERVAL (SEGMENT), IN_EVENT_CONTROL (LEB), LASTID (IDCX), LASTID (SEGMENT), MAPCOLOR (MAP), MAPDISC (MAP), MAPOVER (MAP), MAPPOINT (MAP), MSGAUX (IDCX), MSGDATATYPE (IDCX), MSGDEST (IDCX), MSGDISC (IDCX), NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), NETWORK (STATIC), NOOPS_CHANNAME (STATIC), OLD_WFPROTO (IDCX), ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), OUTAGE (IDCX), OVERLAYDISC (MAP), PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3), PARTICIPATION (STATIC), PMCC_FEATURES (IDCX), PMCC_RECIPE (IDCX), PROBLASTID (IDCX), PROBLEM (IDCX), PROBLEMLOG (IDCX), PROBLEMMAIL (IDCX), PRODTRACK (IDCX), PRODUCTCRITERIA (IDCX), PRODUCTTYPEEVSC (IDCX), QCSTATS (IDCX), REBDONE_DATADAY_FLAG (LEB), REGCOEF (STATIC), REMARK (IDCX), REMARK (LEB), REMARK (REB), REMARK (SEL1), REMARK (SEL2), REMARK (SEL3), REMARK (STATIC), REQUEST (IDCX), REVAUDIT (LEB), RMS_LASTID (IDCX), RULESET (STATIC), SEISGRID (STATIC), SEISINDEX (STATIC), SEMAPHORE (IDCX), SENSOR (STATIC), SITE

	(STATIC), SITEAUX (STATIC), SITECHAN (STATIC), SITEPOLL (STATIC), SITEPOLLCA (STATIC), SITE_ADDRESS (STATIC), SPLP (REB), SPVAR (REB), SREGION (STATIC), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3), STANET (STATIC), STASSOC (IDCX), SUBS (IDCX), SUBSUSER (IDCX), THIRDMOM (REB), TIMEFREQ (REB), TIMESTAMP (IDCX), WEIGHTS (STATIC), WFCONV (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT), WFTAG (IDCX), WFTAG (SEGMENT), WFTAG (SEL1), WFTAG (SEL2), WFTAG (SEL3), XTAG (IDCX)			
Description	Load date. The date and time the record was inserted into the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	ORACLE Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

3.2.405. lddate

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date				Y
Range				

3.2.406. length

Table(s)	WFAUX (IDCX)			
Description	Length of file, bytes (dlfile), waveform length, bytes (wfaux).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)	bytes	NOT ALLOWED	i10	N
Range	length > 0			

3.2.407. lg_snr

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Signal-to-noise ratio of the Lg measurement.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	lg_snr > 0.0			

3.2.408. lineno

Table(s)	REMARK (IDCX), REMARK (LEB), REMARK (REB), REMARK (SEL1), REMARK (SEL2), REMARK (SEL3), REMARK (STATIC)
Description	Line number. This integer is assigned as a sequence number for multiple line comments.

Storage type	Unit	NA value	External	Nullable
number(8,0)		NOT ALLOWED	i8	N
Range	lineno > 0			

3.2.409. loc_all_stas

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that informs the location process whether or not it should use only stations with source-dependent corrections in event locations. If TRUE (1), use all stations in event location. If FALSE (0), only use phase data from stations possessing either an SSSC, SRST, or test-site correction. Any data without a valid correction will not be included in the final event location. Only meaningful if src_dpnt_corr is > 0. Default is TRUE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	loc_all_stas IN {0, 1}			

3.2.410. loc_alpha_only

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that restricts phase data to be used in event location to only those stations contained in the substation list. If FALSE (0), use all stations provided in the Site table. If TRUE (1), only "PRIMARY" station data are used to locate events. This is option desirable in cases where the station network has varying station qualities contributing to events. Default is FALSE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	loc_alpha_only IN {0, 1}			

3.2.411. loc_conf

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Confidence level of location error ellipse used to assess whether the error ellipse was onshore, offshore or mixed (in other words, partially onshore and offshore).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0.0 < loc_conf < 1.0			

3.2.412. loc_dist_varwgt

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)
Description	Logical descriptor that informs the location process if predefined distance variance weighting should be applied to the event location. The predefined weight-

	ing is a set of data variances as a function of distance. If FALSE (0) and both user_var_wgt and srst_var_wgt are also set to FALSE, then variances are determined by the deltim, delslo, and delaz from Arrival. Default is FALSE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	loc_dist_varwgt IN {0, 1}			

3.2.413. loc_sdv_mult

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Large residual multiplier factor. This column is only meaningful when loc_sdv_screen is set to TRUE (1). If TRUE, all data with travel-time/azimuth/slowness residuals greater than this factor times its data variance (standard error) will be ignored during any given iteration of the location process. Default is 3.0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f5.2	Y
Range	loc_sdv_mult > 0.0			

3.2.414. loc_sdv_screen

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that tells the location process whether or not to ignore data with travel-time/azimuth/slowness residuals greater than loc_sdv_mult times its data standard error in determining an event location. If FALSE (0), include data regardless of its residuals, provided it meets other pertinent conditions. Default is FALSE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	loc_sdv_screen IN {0, 1}			

3.2.415. loc_src_dpnt_reg

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Source-dependent region identifier for event location. If source-dependent corrections are applied as part of event location process (src_dpnt_corr > 0), then loc_src_dpnt_reg indicates region name.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any free-format string up to 15 characters			

3.2.416. locality

Table(s)	SITE_ADDRESS (STATIC)

Description	Geographical location within a country.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)		- (hyphen)	a40	Y
Range	any string up to 40 characters			

3.2.417. location

Table(s)	STASSOC (IDCX)			
Description	Location description. This character string describes the location of an event identified from data recorded at a single station. An examples is Fiji-Tonga.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	any free-format string up to 32 characters			

3.2.418. logat

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Log of amplitude divided by period. This measurement of signal size is often reported instead of the amplitude and period separately. This column is only filled if the separate measurements are not available.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	log (nanometers/seconds)	-999.0	f7.2	Y
Range	logat > 0.0			

3.2.419. login

Table(s)	PROBLEMMAIL (IDCX)			
Description	Login name of originator of log entry or a mail subscriber.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(33)		NOT ALLOWED	a33	N
Range	any valid string up to the field width that is a valid email address			

3.2.420. login

Table(s)	PROBLEM (IDCX), PROBLEMLOG (IDCX)			
Description	Login name of originator of log entry or a mail subscriber.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(33)		NOT ALLOWED	a33	Y
Range	any valid string up to the field width that is a valid email address			

3.2.421. lon

Table(s)	MAPPOINT (MAP), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB),

	ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Geographic longitude. Longitudes are measured positive east of the Greenwich meridian.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f9.4	N
Range	-180.0 <= lon <= 180.0			

3.2.422. lon

Table(s)	STASSOC (IDCX)			
Description	Geographic longitude. Longitudes are measured positive east of the Greenwich meridian.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f9.4	Y
Range	-180.0 <= lon <= 180.0			

3.2.423. lon

Table(s)	SITE (STATIC)			
Description	Geographic longitude. Longitudes are measured positive east of the Greenwich meridian.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees	-999.0	f9.4	Y
Range	-180.0 <= lon <= 180.0			

3.2.424. lon1

Table(s)	DSEISINDEX (STATIC), SEISINDEX (STATIC)			
Description	Western-most longitude of the first grid cell in Seisgrid (Dseisgrid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	NOT ALLOWED	f9.4	Y
Range	-180.0 <= lon1 <= 180.0			

3.2.425. lon_hi

Table(s)	BREGION_DEF (STATIC)			
Description	Maximum longitude of the region. Locations east of the Greenwich Meridian have positive longitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	-180.0 <= lon_hi <= 180.0			

3.2.426. lon_lo

Table(s)	BREGION_DEF (STATIC)			
Description	Minimum longitude of the region. Locations east of the Greenwich Meridian have positive longitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	-180.0 <= lon_lo <= 180.0			

3.2.427. lonmajor

Table(s)	MAPDISC (MAP)			
Description	Longitude interval (in degrees) for displaying major grid lines in the Map application.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999	f9.4	Y
Range	0 < lonmajor < 180.0			

3.2.428. lonminor

Table(s)	MAPDISC (MAP)			
Description	Longitude interval (in degrees) for displaying minor grid lines in the Map application.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999	f9.4	Y
Range	-180.0 < lonminor < 180.0			

3.2.429. lonorigradians

Table(s)	MAPDISC (MAP)			
Description	Longitude origin radians. Coordinates in radians of the lower left corner in the Map application. Map uses this for mercator projections only.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	radians	-999	f9.4	Y
Range	-p <= latorigradians <= p			

3.2.430. low_cut

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Low-cut frequency of filter.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	-1.0	f9.4	N

Range	low_cut > 0.0

3.2.431. machine

Table(s)	DLMAN (IDCX)			
Description	Fully qualified domain name of the computer where the connection manager resides.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	N
Range	any string up to 32 characters that points to a valid machine			

3.2.432. mag_all_stas

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that informs magnitude process whether or not it should only use amplitude information from stations with magnitude test-site corrections. If TRUE (1), use all amplitude information in event magnitudes. If FALSE (0), use only amplitude data from stations possessing a magnitude test-site correction; any data without a valid correction will not be included in the magnitude determination. Only meaningful if mag_src_dpnt_reg is not null or "-." Default is TRUE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	mag_all_stas IN {0, 1}			

3.2.433. mag_alpha_only

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that restricts amplitude data to be used in the magnitude determination to only those stations contained in the substation list. If FALSE (0), use all valid amplitudes. If TRUE (1), only "Primary seismic" station data are used for the magnitude calculation. This option is desirable in cases where the station network has varying station qualities contributing to the magnitude. Default is FALSE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	mag_alpha_only IN {0, 1}			

3.2.434. mag_cutoff

Table(s)	PHASOR_DIST_DEPTH_RANGES (STATIC)
Description	Lower bound of body wave magnitude for a phase to be considered for processing by Phasor, if the event mb is >= than this value the program will add it to the phase list if mb-based Automated Phase Selection

	mode of the program is selected.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	mb >= -2.0			

3.2.435. mag_sdv_mult

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Magnitude large residual multiplier factor. This column is meaningful only when mag_sdv_screen is set to TRUE (1). If TRUE, an amplitude with magnitude residuals greater than this factor times its data variance (standard error) will be ignored by the magnitude process. Default is 3.0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f5.2	Y
Range	mag_sdv_mult > 0.0			

3.2.436. mag_sdv_screen

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Logical descriptor that tells magnitude process whether or not to ignore amplitude data with magnitude residuals greater than mag_sdv_mult times its data standard error in determining the given magnitude. If FALSE (0), include data regardless of its residuals, provided it meets other pertinent conditions. Default is FALSE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	mag_sdv_screen IN {0, 1}			

3.2.437. mag_src_dpnt_reg

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Source-dependent region identifier for magnitude determination. If source-dependent corrections are applied as part of the event magnitude determination process, then mag_src_dpnt_reg indicates the region name.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any free-format string up to 15 characters			

3.2.438. magdef

Table(s)	STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)

Description	Magnitude defining switch. A one-character flag indicating whether or not a station magnitude for a given stamag record was used in determining the network magnitude. This column is defining (magdef = d) if it is used in network magnitude calculation or nondefining (magdef = n) if it is not used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	magdef IN {d, n}			

3.2.439. magid

Table(s)	NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3)			
Description	Network magnitude identifier. This value is assigned to identify a network magnitude in the NETMAG table. This column is required for every network magnitude. Origins are linked to their corresponding mb, Ms and ML magnitudes via the mbid, msid and mlid columns respectively.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	magid > 0			

3.2.440. magid

Table(s)	STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Station magnitude identifier. This value is assigned to identify a station magnitude in the STAMAG table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	magid > 0			

3.2.441. magnitude

Table(s)	NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Magnitude. This column gives the magnitude value of the type indicated in magtype. The value is derived in a variety of ways, which are not necessarily linked directly to an arrival (see imb, iml, ims, magtype, mb, ml, and ms).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	NOT ALLOWED -999.0 for netmag	f7.2	Y
Range	magnitude > -2.0			

3.2.442. magpref_mb

Table(s)	PRODUCTTYPEEVSC (IDCX)

Description	Mb magnitude type used for screening (see Netmag.magtype). Default is mb_ave.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	Y
Range	any mb magnitude type up to six characters			

3.2.443. magpref_ms

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Ms magnitude type used for screening (see Netmag.magtype). Default is ms_ave.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	Y
Range	any Ms magnitude type up to six characters			

3.2.444. magres

Table(s)	STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Magnitude residual. This column is the difference between the magnitude for a given stamag record and the network magnitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	-10.0 < magres < 10.0			

3.2.445. magth

Table(s)	DSEISGRID (STATIC), SEISGRID (STATIC)			
Description	Magnitude threshold used to calculate the number of events per year in each grid cell.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	any valid magnitude value			

3.2.446. magtype

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Magnitude type, for example mb, Ms, ML			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		NOT ALLOWED	a4	Y
Range	any valid magnitude type up to the character field width			

3.2.447. magtype

Table(s)	DSEISGRID (STATIC), NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), SEISGRID (STATIC), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Magnitude type, for example mb, Ms, ML			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	Y
Range	any valid magnitude type up to the character field width			

3.2.448. mapfiletype

Table(s)	MAPDISC (MAP)			
Description	Specifies how the Map program handles the referenced Map file. If mapfiletype = all, then the program reads the file in its entirety. If mapfiletype = blk, then the program reads only the blocks necessary for the display area.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		NOT ALLOWED	a4	N
Range	mapfiletype IN {all, blk}			

3.2.449. mapid

Table(s)	FDSNCHAN (STATIC)			
Description	Mapping identifier between old and new channel names			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	N
Range	mapid > 0			

3.2.450. mapid

Table(s)	MAPCOLOR (MAP), MAPDISC (MAP), MAPOVER (MAP)			
Description	Mapdisc identifier. Each mapdisc is assigned a unique positive integer that identifies it in a database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	mapid > 0			

3.2.451. mapname

Table(s)	MAPDISC (MAP)			
Description	Name of the map. Each map in the Map application is assigned a name for identifying the map in a list of all maps.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(64)		NOT ALLOWED	a64	N
Range	any string up to 64 characters			

3.2.452. maptype

Table(s)	MAPDISC (MAP)			
Description	Type of map. A positive integer enumerator for identifying the output graphic type, either raster or vector (maptype = 1 for raster and maptype = 2 for vector).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	maptype IN {1, 2}			

3.2.453. masked

Table(s)	QCSTATS (IDCX)			
Description	Amount of data in the processing interval masked due to point-spikes, spikes, or constant value segments.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999.0	f17.5	Y
Range	masked >= 0			

3.2.454. max_azerr

Table(s)	STATION_PHASE_RELY (STATIC)			
Description	Maximum azimuth error			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	0.0 <= max_azerr <= 360.0			

3.2.455. max_depth

Table(s)	PHASOR_DIST_DEPTH_RANGES (STATIC)			
Description	Maximum allowable depth (km)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	NOT ALLOWED	f9.4	Y
Range	0 <= max_depth <= 700			

3.2.456. max_dist

Table(s)	PHASOR_DIST_DEPTH_RANGES (STATIC)			
Description	Maximum allowable distance (degrees)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees		f8.3	Y

Range	

3.2.457. maxdelta

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC), PHASE_DESCRIPTION (STATIC)			
Description	Maximum allowable distance (degrees)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f8.3	N
Range	maxdelta >= 0.0			

3.2.458. maxdelta

Table(s)	ALLOW_RESID (LEB)			
Description	Maximum station to event distance.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f8.3	Y
Range	maxdelta >= 0.0			

3.2.459. maxdep_err

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Maximum of depth_error product constraint.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	999.0	f9.4	Y
Range	mindeperr < maxdep_err < 999.0			

3.2.460. maxdepth

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC), PHASE_DESCRIPTION (STATIC)			
Description	Maximum allowable depth (km)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	NOT ALLOWED	f9.4	N
Range	0 <= maxdepth <= 700			

3.2.461. maxdepth

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Maximum depth value for a product constraint.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	999.0	f9.4	Y
Range	mindepth < maxdepth < 999.0			

3.2.462. maxesd

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Maximum distance between the event and a station.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1	f6.2	Y
Range	minesd <= maxesd <= 180.0			

3.2.463. maxfreq

Table(s)	PMCC_FEATURES (IDCX)			
Description	Maximum frequency of the detection pixels in the PMCC family (Hz)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz		f9.4	Y
Range				

3.2.464. maxkx

Table(s)	FKDISC (LEB)			
Description	Maximum wavenumber along x-axis in an f-k spectrum. F-k spectra are assumed to be symmetrical, ranging from -maxkx to maxkx.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers-1	-1.0 (Either maxkx or maxsx must be set.)	f7.4	Y
Range	maxkx > 0.0			

3.2.465. maxky

Table(s)	FKDISC (LEB)			
Description	Maximum wavenumber along y-axis of an f-k spectrum. F-k spectra are assumed to be symmetrical, ranging from -maxky to maxky.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers-1	-1.0 (Either maxky or maxsy must be set.)	f7.4	Y
Range	maxky > 0.0			

3.2.466. maxlat

Table(s)	PRODUCTTYPEORIGIN (IDCX), PRODUCTTYPESTA (IDCX)			
Description	Maximum latitude. Locations north of the equator have positive latitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	999.0	f9.4	Y

Range	-90.0 <= maxlat <= 90.0

3.2.467. maxlon

Table(s)	PRODUCTTYPEORIGIN (IDCX), PRODUCTTYPESTA (IDCX)			
Description	Maximum longitude. Locations east of the Greenwich Meridian have positive longitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	999.0	f9.4	Y
Range	-180.0 <= maxlon <= 180.0			

3.2.468. maxmag

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Maximum value of magnitude for a product constraint.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	999.0	f7.2	Y
Range	minmag < maxmag < 999.0			

3.2.469. maxmb_ms

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Maximum value of mb - Ms.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	999.0	f7.2	Y
Range	minmb_ms < maxmb_ms < 999.0			

3.2.470. maxsx

Table(s)	FKDISC (LEB)			
Description	Maximum slowness along the x-axis in a broadband f-k spectrum. F-k spectra are assumed to be symmetrical, ranging from -maxsx to maxsx.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/kilometer	-1.0 (Either maxkx or maxsx must be set.)	f7.4	Y
Range	maxsx > 0.0			

3.2.471. maxsy

Table(s)	FKDISC (LEB)			
Description	Maximum slowness along the y-axis in a broadband f-k spectrum. F-k spectra are assumed to be symmetrical, ranging from -maxsy to maxsy.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(24)	seconds/kilometer	-1.0 (Either maxsy or maxky must be set.)	f7.4	Y
Range	maxsy > 0.0			

3.2.472. mb

Table(s)	EVSC_PROD (IDCX), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Body wave magnitude, mb. (origin) This is the body wave magnitude of an event. The identifier mbid that points to magid in the Netmag table is associated with this column. The information in that record summarizes the method of analysis and data used (see imb, iml, ims, magnitude, magtype, ml, and ms). (evsc_prod) Body wave magnitude for event screening. The type of measurement is indicated by magtype_mb.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	mb > -2.0			

3.2.473. mb_err

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Uncertainty of single-station mb magnitude estimates, used in the computation of the confidence interval of the network estimate of mb minus Ms.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	0.0 < mb_err < 10.0			

3.2.474. mb_max_dist

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Body wave magnitude (mb) station magnitudes at distances greater than mb_max_dist will not be used in network magnitude calculations.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f9.4	Y
Range	0.0 <= mb_max_dist <= 180.0			

3.2.475. mb_min_dist

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Body wave magnitude (mb) station magnitudes at distances less than mb_min_dist will not be used in network magnitude calculations.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(24)	degrees	-999.0	f9.4	Y
Range	0.0 <= mb_min_dist <= 180.0			

3.2.476. mbid

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Magnitude identifier for mb. This column stores the magid for a record in NETMAG. mbid is a foreign key joining ORIGIN to NETMAG where ORIGIN.mbid = NETMAG.magid (see magid, mlid, and msid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	mbid > 0			

3.2.477. mbms

Table(s)	EVSC_PROD (IDCX)			
Description	Difference of body and surface wave magnitudes, mb minus Ms, including a slope term given by mbms_slope of table PRODUCTTYPEEVSC.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	mbms > -999.0			

3.2.478. mbms_conf

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Confidence level for the mb minus Ms screening criterion. This confidence level is for a one-sided confidence interval for mb minus Ms.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0.0 < mbms_conf < 1.0			

3.2.479. mbms_slope

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Slope term (A) for the mb minus Ms relation (Amb - Ms) to account for calibration of magnitude dependence of the screening criterion. The standard default value is 1.25. A typical range of reasonable values is from +1.0 to +2.0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0.0 < mbms_slope < 999.0			

3.2.480. mbms_thresh

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Threshold for the mb minus Ms screening criterion. Events with one-sided confidence intervals for mb minus Ms (Amb - Ms, including the slope term) less than this threshold are screened out at the confidence level given by mbms_conf. Typical values of this threshold range from +1.0 to +4.0, depending on the slope term given by mbms_slope.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	0.0 < mbms_thresh < 999.0			

3.2.481. mbmserr

Table(s)	EVSC_PROD (IDCX)			
Description	Error of mb minus Ms. This is the size of the one-sided confidence interval used for the mb minus Ms screening criterion for a confidence level given by mbms_conf of table Producttypeevsc.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	mbmserr > -999.0			

3.2.482. mean_arrival_time

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Mean arrival time of the estimated signal energy, in epoch time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	Y
Range	any valid epoch time			

3.2.483. merge_adjacent

Table(s)	ARCH_DATA_TYPE (IDCX)			
Description	Merge flag. If y, Archive should merge adjacent rows.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	merge_adjacent IN {y, n}			

3.2.484. method

Table(s)	DERVDISC (IDCX)			
Description	Method or Application associated to the dervdisc entry.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(17)		- (hyphen)	a17	Y
Range	any string up to the character field width			

3.2.485. mfoff

Table(s)	MSGDISC (IDCX)			
Description	Offset in bytes to beginning of message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)	bytes	-1	i10	Y
Range	mfoff > 0			

3.2.486. min_azerr

Table(s)	STATION_PHASE_RELY (STATIC)			
Description	Minimum azimuth error			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	degrees		f8.3	Y
Range	0.0 <= min_azerr <= 360.0			

3.2.487. min_depth

Table(s)	PHASOR_DIST_DEPTH_RANGES (STATIC)			
Description	Minimum allowable depth (km)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	NOT ALLOWED	f9.4	Y
Range	-999.0 < min_depth < max_depth			

3.2.488. min_dist

Table(s)	PHASOR_DIST_DEPTH_RANGES (STATIC)			
Description	Minimum allowable distance (degrees)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees		f8.3	Y
Range				

3.2.489. min_dp_snr_pp

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum pP depth phase signal-to-noise ratio required for depth phase analysis.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0 < min_dp_snr_pp < 999			

3.2.490. min_dp_snr_sp

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum sP depth phase signal-to-noise ratio required for depth phase analysis.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0 < min_dp_snr_sp < 999			

3.2.491. min_dt_pp

Table(s)	EVSC_PROD (IDCX)			
Description	PP-P travel-time difference at nearest station beyond 25 deg.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	0 < min_dt_pp < 999			

3.2.492. min_dt_sp

Table(s)	EVSC_PROD (IDCX)			
Description	SP-P travel-time difference at nearest station beyond 25 deg.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	0 < min_dt_sp < 999			

3.2.493. min_mb

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum mb magnitude for application of event-screening criteria.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	min_mb > -999.0			

3.2.494. min_moveout_pp

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum (pP-P) travel time moveout required for the depth screening criterion to be applied to depth-phase solutions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	-999.0 < min_moveout_pp < 999.0			

3.2.495. min_moveout_sp

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum (sP-P) travel time moveout required for the depth screening criterion to be applied to depth-phase solutions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	-999.0 < min_moveout_sp < 999.0			

3.2.496. min_ndef

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum number of defining phases required for the event-screening criteria to be applied to a seismic-acoustic event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	min_ndef > 0			

3.2.497. min_ndp_pp

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum number of pP depth phases required for the depth screening criterion to be applied to depth-phase solutions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	min_ndp_pp > 0			

3.2.498. min_ndp_sp

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum number of sP depth phases required for the depth screening criterion to be applied to depth-phase solutions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	min_ndp_sp > 0			

3.2.499. min_nsta_ms

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum number of stations required to contribute to the network Ms estimate for the mb minus Ms screening criterion to be applied.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y

Range	0 < min_nsta_ms < 1000

3.2.500. min_wdepth

Table(s)	EVSC_PROD (IDCX)			
Description	Minimum depth of water within the location error ellipse defined by smaj_sc, smin_sc, and strike.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	meters	-999.0	f7.2	Y
Range	min_wdepth > 0.0			

3.2.501. min_wdepth_thresh

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Minimum water depth threshold for application of the hydroacoustic screening criteria.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f7.2	Y
Range	0.0 < min_wdepth_thresh < 10.0			

3.2.502. minchan

Table(s)	RULESET (STATIC)			
Description	Minimum number of channels. It can be a number of channels or a percentage.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

3.2.503. mindelta

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC)			
Description	Minimum allowable distance (degrees)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees		f8.3	N
Range				

3.2.504. mindelta

Table(s)	PHASE_DESCRIPTION (STATIC)			
Description	Minimum distance (degrees)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees		f8.3	N
Range				

3.2.505. mindelta

Table(s)	ALLOW_RESID (LEB)			
Description	Minimum station to event distance.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f8.3	N
Range	mindelta >= 0.0			

3.2.506. mindep_err

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Minimum value of depth_error constraint.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f9.4	Y
Range	-999.0 < mindep_err < maxdep_err			

3.2.507. mindepth

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC), PHASE_DESCRIPTION (STATIC)			
Description	Minimum allowable depth (km)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	NOT ALLOWED	f9.4	N
Range	-999.0 < mindepth < maxdepth			

3.2.508. mindepth

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Minimum depth value for a product constraint (km).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f9.4	Y
Range	-999.0 < mindepth < maxdepth			

3.2.509. minesd

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Minimum distance between an event and stations.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1.0	f6.2	Y
Range	0.0 <= minesd <= maxesd			

3.2.510. minfreq

Table(s)	PMCC_FEATURES (IDCX)

Description	Minimum frequency of the detection pixels in the PMCC family (Hz)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz		f9.4	Y
Range				

3.2.511. minlat

Table(s)	PRODUCTTYPEORIGIN (IDCX), PRODUCTTYPESTA (IDCX)			
Description	Minimum latitude. Locations north of the equator have positive latitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f9.4	Y
Range	-90.0 <= minlat <= 90.0			

3.2.512. minlon

Table(s)	PRODUCTTYPEORIGIN (IDCX), PRODUCTTYPESTA (IDCX)			
Description	Minimum longitude defining a reference area. Locations east of the Greenwich Meridian have positive longitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f9.4	Y
Range	-180.0 <= minlon <= 180.0			

3.2.513. minmag

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Minimum value of the magnitude constraint.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	-999.0 < minmag < maxmag			

3.2.514. minmb_ms

Table(s)	PRODUCTTYPEORIGIN (IDCX)			
Description	Minimum value of mb - Ms product constraint.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	-999.0 < minmb_ms < maxmb_ms			

3.2.515. missing

Table(s)	QCSTATS (IDCX)
Description	Amount of missing data in the interval, measured in seconds.

Storage type	Unit	NA value	External	Nullable
float(53)	seconds	NOT ALLOWED	f17.5	Y
Range	missing < 9999999999.999			

3.2.516. ml

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Local magnitude (ML) of an event. The identifier mlid, which points to magid in the Netmag table, is associated with this column. The information in that record summarizes the method of analysis and the data used (see imb, iml, ims, magnitude, magtype, mb, and ms).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	ml > -2.0			

3.2.517. mlid

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Magnitude identifier for ML. This column stores the magid for a record in NETMAG. mlid is a foreign key joining ORIGIN to NETMAG, where ORIGIN.mlid = NETMAG.magid (see magid, msid, and mbid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	mlid > 0			

3.2.518. mmodel

Table(s)	IN_EVENT_CONTROL (LEB)			
Description	Magnitude model. This character string identifies the magnitude model employed for station (stamag) or overall network magnitude calculation (event_control). In stamag, mmodel is the unique magnitude model as extracted from the magnitude correction file. In event_control, mmodel indicates only whether or not mixed models were employed (mixed) or a unique magnitude model was used for all stations. In the latter case, it would be identical to STAMAG.mmodel.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any free-format string up to 15 characters			

3.2.519. mmodel

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), STAMAG (LEB),

	STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Magnitude model. This character string identifies the magnitude model employed for station (stamag) or overall network magnitude calculation (event_control). In stamag, mmodel is the unique magnitude model as extracted from the magnitude correction file. In event_control, mmodel indicates only whether or not mixed models were employed (mixed) or a unique magnitude model was used for all stations. In the latter case, it would be identical to stamag.mmodel.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any free-format string up to 15 characters			

3.2.520. modauthor

Table(s)	REQUEST (IDCX)			
Description	Author of last state change.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any string up to 15 characters			

3.2.521. moddate

Table(s)	INTERVAL (IDCX), INTERVAL (SEGMENT)			
Description	Modification date. The date and time the record was last updated (state column) in the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	ORACLE Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

3.2.522. modtime

Table(s)	REQUEST (IDCX)			
Description	Modification time. The epoch time that the record was last updated in the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	Y
Range	any valid epoch time			

3.2.523. moveout_pp

Table(s)	EVSC_PROD (IDCX)			
Description	Moveout of (pP-P) travel times for seismic stations between 25 and 100 degrees from an event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(24)	seconds	-999.0	f7.2	Y
Range	moveout_pp > -999.0			

3.2.524. moveout_sp

Table(s)	EVSC_PROD (IDCX)			
Description	Moveout of (sP-P) travel times for seismic stations between 25 and 100 degrees from an event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	moveout_sp > -999.0			

3.2.525. mpdescrip

Table(s)	MAPPOINT (MAP)			
Description	Arbitrary string describing the referenced geographic point.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)		- (hyphen)	a50	Y
Range	any string up to 50 characters			

3.2.526. mplabel

Table(s)	MAPPOINT (MAP)			
Description	String used as a label for the geographic point described by a record in the MAP-POINT table (for example, Paris, London and so on).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(65)		- (hyphen)	a65	Y
Range	any string up to 65 characters			

3.2.527. mptype

Table(s)	MAPPOINT (MAP)			
Description	String specifying the type of geographic point described by a record in the MAP-POINT table. Examples include cities, mines, and so on.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	N
Range	any free-format (spaces allowed) string up to 20 characters			

3.2.528. ms

Table(s)	EVSC_PROD (IDCX), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)
Description	Surface wave magnitude.

Storage type	Unit	NA value	External	Nullable
float(24)	magnitude	-999.0	f7.2	Y
Range	Ms > -2.0			

3.2.529. ms_err

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Uncertainty of single-station Ms magnitude estimates used in the computation of the confidence interval of the network estimate of mb minus Ms.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	magnitude	-999.0	f7.2	Y
Range	0.0 < ms_err < 10.0			

3.2.530. mscore

Table(s)	EVSC_PROD (IDCX)			
Description	Score for the mb minus Ms event-screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	mscore > -999.0			

3.2.531. msgdformat

Table(s)	MSGDATATYPE (IDCX)			
Description	Format of the data, currently one of GSE2.0, IMS1.0 or IMS2.0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	Y
Range	any string up to the character field width			

3.2.532. msgdformat

Table(s)	FPDESCRIPTION (IDCX)			
Description	Format of the data, for example, binary, ps, IMS1.0, TXT, CD-1.1, CD-1.0, GSE2.0, IMS2.0).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	Y
Range	any string up to the character field width			

3.2.533. msgdid

Table(s)	MSGDEST (IDCX)			
Description	Message destination identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number(10,0)		-1	i10	N
Range	msgdid > 0			

3.2.534. msgdtype

Table(s)	FPDESCRIPTION (IDCX), MSGDATATYPE (IDCX)			
Description	Data type of a data section within an VDMS message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	Y
Range	any string up to 16 characters that is a recognized data type			

3.2.535. msgid

Table(s)	FTPFAILED (IDCX), MSGAUX (IDCX), MSGDATATYPE (IDCX), MSG-DISC (IDCX), PRODTRACK (IDCX)			
Description	Message identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	msgid > 0			

3.2.536. msgid

Table(s)	MSGDEST (IDCX)			
Description	The message identifier of the response message created by VDMS.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	msgid > 0			

3.2.537. msgrow

Table(s)	MSGAUX (IDCX)			
Description	Number of lines in a message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)	lines	-1	i4	N
Range	msgrow > 0			

3.2.538. msgsrc

Table(s)	MSGDISC (IDCX)			
Description	Source code from where the message was sent.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y

Range	any string up to 16 characters

3.2.539. msgtype

Table(s)	DATAUSER (IDCX)			
Description	Message type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	N
Range	any string up to 16 characters			

3.2.540. msgtype

Table(s)	MSGDISC (IDCX)			
Description	Message type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y
Range	any string up to 16 characters			

3.2.541. msgver

Table(s)	MSGDISC (IDCX)			
Description	Message Subsystem version number.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	any string up to eight characters			

3.2.542. msid

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Magnitude identifier for Ms. This column stores the magid for a record in NETMAG. msid is a foreign key joining ORIGIN to NETMAG, where ORIGIN.msid = NETMAG.magid (see magid, mlid, and mbid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	msid > 0			

3.2.543. msize

Table(s)	MSGDISC (IDCX)			
Description	Size of bytes of message or section of a message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number(10,0)	bytes	-1	i10	Y
Range	msize > 0			

3.2.544. msize

Table(s)	MSGDATATYPE (IDCX)			
Description	Size of bytes of message or section of a message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)	bytes	-1	i8	Y
Range	msize > 0			

3.2.545. mtype

Table(s)	AMPDESCRIPT (IDCX)			
Description	Measurement type. This column defines how the amplitude is measured in a given time window. The following values are allowed: peak (maximum amplitude), stav (maximum short-term average amplitude), rms (root-mean-squared amplitude), peak2tr (maximum peak-to-trough amplitude), and 1stpeak (first peak amplitude).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	mtype IN {peak, stav, rms, peak2tr, 1stpeak}			

3.2.546. na_value

Table(s)	NA_VALUE (IDCX)			
Description	Value to be inserted for a column when no other value is available.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)		NOT ALLOWED	a30	Y
Range	any valid string up to the character width			

3.2.547. name

Table(s)	INTERVAL (IDCX), INTERVAL (SEGMENT)			
Description	Name of the station or processing pipeline described by this row. Examples are sel1, sel2, sel3, ANALYSIS, RECALL, REB, TMPR, MIG, EVCH, EVSC, SEG and QTR.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	N
Range	any string up to the character limit			

3.2.548. name

Table(s)	CHAN_GROUPS (STATIC)

Description	Name of the station.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	N
Range	any alphanumeric string up to the character field width			

3.2.549. name

Table(s)	SITE_ADDRESS (STATIC)			
Description	Name of a constant, variable, or parameter.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		- (hyphen)	a20	Y
Range	any string up to the character limit			

3.2.550. name

Table(s)	CHAN_GROUPS_AUXWF (STATIC)			
Description	Name of the station.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	Y
Range	any alphanumeric string up to the character field width			

3.2.551. name

Table(s)	RULESET (STATIC)			
Description	Name of the rule.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		NOT ALLOWED	a32	Y
Range	any valid string up to 32 characters			

3.2.552. name

Table(s)	FPDESCRIPTION (IDCX)			
Description	Name of a constant, variable, or parameter.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		- (hyphen)	a64	Y
Range	any string up to the character limit			

3.2.553. nass

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)
Description	Number of associated arrivals. This column gives the number of arrivals associ-

	ated with the origin.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		-1	i4	Y
Range	nass > 0			

3.2.554. navcep

Table(s)	TIMEFREQ (REB)			
Description	Average maximum value in the two-dimensional cepstrum of the north component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	navcep >= 0			

3.2.555. navcor

Table(s)	TIMEFREQ (REB)			
Description	Average autocorrelation along the time axis across all frequencies excluding randomized points of the north component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	navcor >= 0			

3.2.556. navpct

Table(s)	TIMEFREQ (REB)			
Description	Average ratio of bad points to total of the north component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	0 <= navpct <= 1			

3.2.557. ncalib

Table(s)	BEAMAUX (STATIC), INSTRUMENT (STATIC)			
Description	Nominal calibration factor. This conversion factor maps digital data to earth displacement. The factor holds true at the oscillation period specified by ncalper. A positive value means ground motion increasing in component direction (up, north, east) is indicated by increasing counts. A negative value means the opposite. Actual calibration for a particular recording is determined using the WFDISC and SENSOR tables (see calratio).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers per digital count	NOT ALLOWED	f16.6	Y
Range	any nonzero floating point number			

3.2.558. ncalper

Table(s)	BEAMAUX (STATIC), INSTRUMENT (STATIC)			
Description	Calibration period. This column is the period for which ncalib is valid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	NOT ALLOWED	f16.6	Y
Range	ncalper > 0.0			

3.2.559. nconstseg

Table(s)	QCSTATS (IDCX)			
Description	Number of constant valued segments in the detection processing interval.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	nconstseg >= 0			

3.2.560. ndef

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	For a given origin, this is the number of time-defining phases that contributed to the location of the origin. A time-defining phase is a phase whose time is used as a constraint in the location.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	Y
Range	0 < ndef <= nass			

3.2.561. ndp

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Number of depth phases. This column gives the number of depth phases used in calculating depth/depdp (see depdp).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		-1	i4	Y
Range	ndp >= 0			

3.2.562. ndp_snr_pp

Table(s)	EVSC_PROD (IDCX)
Description	Number of pP depth phases meeting the snr criterion defined by min_dp_snr_pp of table PRODUCTTYPEEVSC.

Storage type	Unit	NA value	External	Nullable
number(8,0)		-1	i8	Y
Range	ndp_snr_pp > -1			

3.2.563. ndp_snr_sp

Table(s)	EVSC_PROD (IDCX)			
Description	Number of sP depth phases meeting the snr criterion defined by min_dp_snr_sp of table PRODUCTTYPEEVSC.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	ndp_snr_sp > -1			

3.2.564. net

Table(s)	SITEPOLL (STATIC)			
Description	Unique network identifier. This character string is the name of a seismic network. Examples include YKA, I02AR, H01W2, SADO, CUR_IDC and IMS_PRI.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any string up to the character field width			

3.2.565. net

Table(s)	SITEPOLLCA (STATIC)			
Description	Unique network identifier. This character string is the name of a seismic network. Examples include YKA, I02AR, H01W2, SADO, CUR_IDC and IMS_PRI.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	Y
Range	any string up to the character field width			

3.2.566. net

Table(s)	AFFILIATION (STATIC), AFFILIATION_EXCL (STATIC), NETWORK (STATIC), PARTICIPATION (STATIC), STANET (STATIC), WEIGHTS (STATIC)			
Description	Unique network identifier. This character string is the name of a seismic network. Examples include YKA, I02AR, H01W2, SADO, CUR_IDC and IMS_PRI.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	any string up to the character field width			

3.2.567. net

Table(s)	AFFILIATION_SUB (STATIC)			
Description	Unique network identifier. This character string is the name of a seismic network. Examples include YKA, I02AR, H01W2, SADO, CUR_IDC and IMS_PRI.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a6	N
Range	any upper-case string up to the character field width			

3.2.568. net

Table(s)	DERVDISC (IDCX)			
Description	Infrasound network name (collection of stations)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	Y
Range	any string up to the character field width			

3.2.569. net

Table(s)	NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3)			
Description	Unique network identifier. This character string is the name of a seismic network. Examples include YKA, I02AR, H01W2, SADO, CUR_IDC and IMS_PRI.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	any string up to eight characters			

3.2.570. net

Table(s)	AFFILIATION_LP (STATIC), HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB)			
Description	Unique network identifier. This character string is the name of a seismic network. Examples include YKA, I02AR, H01W2, SADO, CUR_IDC and IMS_PRI.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	Y
Range	any string up to the character field width			

3.2.571. netname

Table(s)	NETWORK (STATIC)
Description	Network name. String containing the name of a network.

Storage type	Unit	NA value	External	Nullable
varchar2(80)		- (hyphen)	a80	Y
Range	any string up to 80 characters			

3.2.572. nettype

Table(s)	NETWORK (STATIC)			
Description	Network type. This four-character string specifies the type of network (ar = array, lo = local area, ww = world-wide) for the given value of net.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	Y
Range	any lower-case string up to four characters			

3.2.573. nevyr

Table(s)	DSEISGRID (STATIC), SEISGRID (STATIC)			
Description	Average number of events per year with magnitude above magth whose location is within the grid cell defined by icell.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f9.2	Y
Range	nevyr >= 0			

3.2.574. newname

Table(s)	FDSNCHAN (STATIC)			
Description	FDSN channel identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	N
Range	any alphanumeric string up to the character field width			

3.2.575. nhydarr

Table(s)	HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB)			
Description	Number of arrivals in the group.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		-1	i4	Y
Range	nhydarr <= number of stations in the group			

3.2.576. nlat

Table(s)	DSEISINDEX (STATIC), SEISINDEX (STATIC)
Description	Number of latitudes in Seisgrid (Dseisgrid).

Storage type	Unit	NA value	External	Nullable
number(8,0)		NOT ALLOWED	i8	Y
Range	nlat > 0.0			

3.2.577. nlon

Table(s)	DSEISINDEX (STATIC), SEISINDEX (STATIC)			
Description	Number of longitudes in Seisgrid (Dseisgrid).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	Y
Range	nlon > 0.0			

3.2.578. nois

Table(s)	SITEAUX (STATIC)			
Description	Nominal background seismic noise level.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers	-1.0	f10.1	Y
Range	nois >= 0.0			

3.2.579. noise_high_band

Table(s)	EVSC_HYDRO (IDCX)			
Description	Hydroacoustic noise level in the high-frequency band (32-64 Hz) from table amplitude for a given orid/sta pair.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	dB re mPa	-1.0	f7.2	Y
Range	noise_high_band > 0.0			

3.2.580. noissd

Table(s)	SITEAUX (STATIC)			
Description	Standard deviation of the log noise amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f5.2	Y
Range	noissd > 0.0			

3.2.581. note_missing_data

Table(s)	ARCH_DATA_TYPE (IDCX)			
Description	Missing data flag. If y, there are missing data for the datatype indicated in the arch_data_type table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(2)		- (hyphen)	a2	Y
Range	note_missing_data IN {y, n}			

3.2.582. nsamp

Table(s)	OLD_WFPROTO (IDCX)			
Description	Number of samples in the waveform segment. This must be greater than one, otherwise the time and endtime will be equal and the time interval will be zero.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		NOT ALLOWED	i8	Y
Range	nsamp > 1			

3.2.583. nsamp

Table(s)	WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Number of samples in the waveform segment. This must be greater than one, otherwise the time and endtime will be equal and the time interval will be zero.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	Y
Range	nsamp > 1			

3.2.584. nseg

Table(s)	QCSTATS (IDCX)			
Description	Number of masked segments in the detection processing interval.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	nseg >= 0			

3.2.585. nsta

Table(s)	NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3)			
Description	Number of stations contributing to the network magnitude estimate.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	nsta > -1			

3.2.586. nsta_mb

Table(s)	EVSC_PROD (IDCX)			
Description	Number of stations contributing to the mb estimate used in event screening.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number(8,0)		-1	i8	Y
Range	nsta_mb > -1			

3.2.587. nsta_ms

Table(s)	EVSC_PROD (IDCX)			
Description	Number of stations contributing to the Ms estimate used for event screening.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	nsta_ms > -1			

3.2.588. num_cross

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Number of times that the estimated signal pressure squared crosses noise_onset_thresh between onset_time and termination_time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	num_cross > 0			

3.2.589. num_defining

Table(s)	PHASE_WEIGHT_EVENT_DEFINE (STATIC)			
Description	Number of defining phases			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	N
Range				

3.2.590. numfailedattempt

Table(s)	FTPFAILED (IDCX)			
Description	Number of failed attempts to retrieve message via FTP.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		0	i4	Y
Range	numfailedattempt > 0			

3.2.591. nx

Table(s)	FKDISC (LEB)			
Description	Total number of X sample points (either slowness or wavenumber, depending on fktyp) in a f-k spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number(4,0)		NOT ALLOWED	i4	Y
Range	nx > 0			

3.2.592. ny

Table(s)	FKDISC (LEB)			
Description	Total number of Y sample points (either slowness or wavenumber, depending on fktyp) in a f-k spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	Y
Range	ny > 0			

3.2.593. objtype

Table(s)	GA_TAG (SEL1)			
Description	Defines the id as either an arid (a) or an orid (o).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		NOT ALLOWED	a1	Y
Range	objtype IN {a, o}			

3.2.594. obs_type

Table(s)	PHASE_WEIGHT_EVENT_DEFINE (STATIC)			
Description	Observation type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	N
Range	any string up to (16, 24) characters that is a valid user name			

3.2.595. obsolete

Table(s)	FILEPRODUCT (IDCX)			
Description	Flag to indicate if the data in the fileproduct file are obsolete. 0 indicates that the data are not obsolete, and 1 indicates that the data are obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)		NOT ALLOWED	i1	Y
Range	obsolete IN {0, 1}			

3.2.596. obsolete

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)			i1	Y

Range	

3.2.597. obsolete

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(1,0)				Y
Range				

3.2.598. offdate

Table(s)	SUBS (IDCX)			
Description	Indicates the last date at which the subscription was active. For currently active subscriptions, this is set to 01-Jan-4700. Unlike other uses of ondate in the IDC Database Schema, the column accepts any valid ORACLE date.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	ORACLE Date	01 Jan 4700	a21	Y
Range	any valid ORACLE date			

3.2.599. offdate

Table(s)	AFFILIATION_SUB (STATIC), CHAN_GROUPS (STATIC)			
Description	Turn off date. This column is the date after which the archive specifications, regional coefficient, station, sensor, or subscription indicated was no longer applicable, turned off, dismantled, or moved (see ondate).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	N
Range	Julian date of the form yyyyddd			

3.2.600. offdate

Table(s)	ARCH_DATA_TYPE (IDCX), CHAN_GROUPS_AUXWF (STATIC), REGCOEF (STATIC), RULESET (STATIC), SITE (STATIC), SITECHAN (STATIC), WEIGHTS (STATIC)			
Description	Turn off date. This column is the date after which the archive specifications, regional coefficient, station, sensor, or subscription indicated was no longer applicable, turned off, dismantled, or moved (see ondate).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	Julian date of the form yyyyddd			

3.2.601. oldname

Table(s)	FDSNCHAN (STATIC)

Description	Old channel name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	N
Range	any alphanumeric string up to the character field width			

3.2.602. ondate

Table(s)	SUBS (IDCX)			
Description	Indicates the first date at which the subscription became active. Unlike other uses of ondate in the IDC Database Schema, the column accepts any valid ORACLE date.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	ORACLE Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

3.2.603. ondate

Table(s)	CHAN_GROUPS (STATIC)			
Description	Turn on date. This column is the date on which the archive specifications, regional coefficient, station, sensor, or subscription indicated became applicable or began operating. Offdate and ondate are not intended to accommodate temporary downtimes, but rather to indicate the time period for which the columns of the station (lat, lon, elev) are valid for the given station code. Stations are often moved, but with the station code remaining unchanged.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	N
Range	Julian date of the form yyyyddd			

3.2.604. ondate

Table(s)	AFFILIATION_SUB (STATIC), RULESET (STATIC), SITE (STATIC), SITECHAN (STATIC)			
Description	Turn on date. This column is the date on which the archive specifications, regional coefficient, station, sensor, or subscription indicated became applicable or began operating. Offdate and ondate are not intended to accommodate temporary downtimes, but rather to indicate the time period for which the columns of the station (lat, lon, elev) are valid for the given station code. Stations are often moved, but with the station code remaining unchanged.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	Julian date of the form yyyyddd			

3.2.605. ondate

Table(s)	REGCOEF (STATIC)

Description	Turn on date. This column is the date on which the archive specifications, regional coefficient, station, sensor, or subscription indicated became applicable or began operating. Offdate and ondate are not intended to accommodate temporary downtimes, but rather to indicate the time period for which the columns of the station (lat, lon, elev) are valid for the given station code. Stations are often moved, but with the station code remaining unchanged.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		- (hyphen)	i8	Y
Range	Julian date of the form yyyyddd			

3.2.606. ondate

Table(s)	CHAN_GROUPS_AUXWF (STATIC)			
Description	Turn on date. This column is the date on which the archive specifications, regional coefficient, station, sensor, or subscription indicated became applicable or began operating. Offdate and ondate are not intended to accommodate temporary downtimes, but rather to indicate the time period for which the columns of the station (lat, lon, elev) are valid for the given station code. Stations are often moved, but with the station code remaining unchanged.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	Julian date of the form yyyyddd			

3.2.607. ondate

Table(s)	ARCH_DATA_TYPE (IDCX), WEIGHTS (STATIC)			
Description	Turn on date. This column is the date on which the archive specifications, regional coefficient, station, sensor, or subscription indicated became applicable or began operating. Offdate and ondate are not intended to accommodate temporary downtimes, but rather to indicate the time period for which the columns of the station (lat, lon, elev) are valid for the given station code. Stations are often moved, but with the station code remaining unchanged.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	Y
Range	Julian date of the form yyyyddd			

3.2.608. onset_time

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Estimated onset time of signal.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	Y
Range	any valid epoch time			

3.2.609. orderby

Table(s)	DSEISINDEX (STATIC), SEISINDEX (STATIC)			
Description	Grid order. The seismic grid can either be ordered by latitude or by longitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	Y
Range	orderby IN {lat, lon}			

3.2.610. orid

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)				N
Range				

3.2.611. orid

Table(s)	IN_EVENT_CONTROL (LEB)			
Description	Origin identifier that relates a record in these tables to a record in the origin table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	orid > 0			

3.2.612. orid

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), CEPPKS (REB), COMPLEXITY (REB), EVENT_CONTROL (LEB), EVENT_CONTROL (REB), EVSC_HYDRO (IDCX), EVSC_PROD (IDCX), EVSC_REGIONAL (IDCX), NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), SPLP (REB), TIMEFREQ (REB)			
Description	Origin identifier that relates to a record in the origin table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	orid > 0			

3.2.613. orid

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORI-

	GIN (SEL2), ORIGIN (SEL3)			
Description	Unique origin identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	orid > 0			

3.2.614. orid

Table(s)	PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3), REQUEST (IDCX), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Origin identifier that relates to a record in the origin table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	orid > 0			

3.2.615. otgid

Table(s)	OUTAGE (IDCX)			
Description	Outage identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	otgid > 0			

3.2.616. outauth

Table(s)	WFCONV (IDCX)			
Description	Flag showing if output data are authenticated.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	outauth IN {y, n}			

3.2.617. outcomp

Table(s)	WFCONV (IDCX)			
Description	Output data compression type. The only type currently supported is Canadian compression (CA).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		0	a2	Y
Range	outcomp IN {CA}			

3.2.618. outsamp

Table(s)	WFCONV (IDCX)			
Description	Number of output samples per packet.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		0	i8	Y
Range	outsamp > 0			

3.2.619. outtype

Table(s)	WFCONV (IDCX)			
Description	Output fixed width datatype.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	same as datatype			

3.2.620. overlayid

Table(s)	MAPOVER (MAP), OVERLAYDISC (MAP)			
Description	Overlay identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	overlayid > 0			

3.2.621. overlayname

Table(s)	OVERLAYDISC (MAP)			
Description	Name of the map overlay.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		- (hyphen)	a64	N
Range	any free format character string up to 64 characters			

3.2.622. parid

Table(s)	PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3)			
Description	Predicted arrival identifier. Every event-based parrival measure is assigned a unique positive integer that identifies it in the database. If an associated amplitude record exists, then parid links it to parrival.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	parid > 0			

3.2.623. parid

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Predicted arrival identifier. Every event-based parrival measure is assigned a unique positive integer that identifies it in the database. If an associated amplitude record exists, then parid links it to parrival.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	parid >= -1			

3.2.624. password

Table(s)	FTPLOGIN (IDCX)			
Description	Password for remote FTP site for pushing data messages from PIDC.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	Y
Range	any string up to 16 characters			

3.2.625. pctoffsh

Table(s)	EVSC_PROD (IDCX)			
Description	Percentage of location error ellipse that is determined to be offshore.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0.0 <= pctoffsh < 100.0			

3.2.626. peak_level

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Pressure of largest absolute signal value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	dB re mPa	-1.0	f9.4	Y
Range	peak_level >= 0.0			

3.2.627. peak_time

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Epoch time of largest absolute signal value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	Y
Range	any valid epoch time			

3.2.628. per

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Measured period at the time of the amplitude measurement.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f7.2	Y
Range	per > 0.0			

3.2.629. per_hi

Table(s)	BREGION_TAB (STATIC)			
Description	Upper bound of the period window. Never used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	seconds	-1	f8.3	Y
Range				

3.2.630. per_lo

Table(s)	BREGION_TAB (STATIC)			
Description	Lower bound of the period window. Never used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	seconds	-1	f8.3	Y
Range				

3.2.631. phase

Table(s)	COMPLEXITY (REB)			
Description	Phase type. The identity of a phase that has been associated to an arrival. Standard labels for phases are used (for example, P, PKP, PcP, pP, and so on). Both upper- and lower-case letters are available and should be used when appropriate; for example, pP or PcP.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen) if this column does not apply to seismic phases	a6	N
Range	any string up to the character field width that conforms to scientific practice			

3.2.632. phase

Table(s)	ALLOW_RESID (LEB), DEFINE_PHASE_RESIDUAL (STATIC), PHASE_DESCRIPTION (STATIC), WEIGHTS (STATIC)
Description	Phase type. The identity of a phase that has been associated to an arrival. Standard labels for phases are used (for example, P, PKP, PcP, pP, and so on). Both

	upper- and lower-case letters are available and should be used when appropriate; for example, pP or PcP.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen) if this column does not apply to seismic phases	a8	N
Range	any string up to the character field width that conforms to scientific practice			

3.2.633. phase

Table(s)	PHASOR_DIST_DEPTH_RANGES (STATIC)			
Description	Phase type. The identity of a phase that has been associated to an arrival. Standard labels for phases are used (for example, P, PKP, PcP, pP, and so on). Both upper- and lower-case letters are available and should be used when appropriate; for example, pP or PcP.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen) if this column does not apply to seismic phases	a8	Y
Range	any string up to the character field width that conforms to scientific practice			

3.2.634. phase

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB), ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Phase type. The identity of a phase that has been associated to an arrival. Standard labels for phases are used (for example, P, PKP, PcP, pP, and so on). Both upper- and lower-case letters are available and should be used when appropriate; for example, pP or PcP.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen) if this column does not apply to seismic phases	a8	Y
Range	any string up to the character field width that conforms to scientific practice			

3.2.635. phase_type

Table(s)	PHASE_DESCRIPTION (STATIC), PHASE_WEIGHT_EVENT_DEFINE (STATIC)

Description	Phase type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	N
Range	any valid string up to 24 characters			

3.2.636. pid

Table(s)	SEMAPHORE (IDCX)			
Description	Process id of the ParseData handling the station			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	0 < pid < system-dependent pid limit			

3.2.637. pkamp

Table(s)	CEPPKS (REB)			
Description	Amplitude of consistent cepstral peak. This column value is set to zero if there are no consistent peaks.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometer-seconds	0.0	f7.5	Y
Range	pkamp > 0.0			

3.2.638. pkqf

Table(s)	CEPPKS (REB)			
Description	Frequency of consistent cepstral peak. This column value is set to zero if no consistent peaks exist.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	0.0	f11.8	N
Range	pkqf > 0.0			

3.2.639. planlr

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Planarity of an S-type polarization column defined as 1 - l3/l2, where l2 and l3 are eigenvalues from the decomposition of the covariance matrix. Planarity is measured at the time of maximum 3-component amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	0.0 <= planlr <= 1.0			

3.2.640. plans

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Planarity of an S-type polarization column defined as 1 - l3/l2, where l2 and l3 are eigenvalues from the decomposition of the covariance matrix. Planarity is measured at the time of maximum 3-component amplitude. The only difference between plans and planlr is in the definition of the overlapping time windows.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	0.0 <= plans <= 1.0			

3.2.641. plid

Table(s)	PROBLEMLOG (IDCX)			
Description	Problem log identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	plid > 0			

3.2.642. pmcc3d

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i1	Y
Range				

3.2.643. pmccrecid

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	N
Range				

3.2.644. pn_snr

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Signal-to-noise ratio of the Pn measurement.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	pn_snr > 0.0			

3.2.645. pnlg

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Logarithm (base 10) of the distance corrected Pn/Lg ratio.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	pnlg > -999.0			

3.2.646. pnlg_qual

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Data quality indicator for the Pn/Lg measurement. A string of zeros indicates good data quality. See quality flag note for pnsn_qual.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)		- (hyphen)	a10	Y
Range	any valid ORACLE string			

3.2.647. pnsmax

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Maximum of the Pn/Sn and Pn/Lg ratios.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	pnsmax > -999.0			

3.2.648. pnsmax_corr

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Correction term of the maximum of the Pn/Sn and Pn/Lg ratios.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	pnsmax_corr > -999.0			

3.2.649. pnsmax_err

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Standard error of the maximum of the Pn/Sn and Pn/Lg ratios.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	pnsmax_err > -999.0			

3.2.650. pnsn

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Logarithm (base 10) of the distance-corrected Pn/Sn amplitude ratio.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	pnsn > -999.0			

3.2.651. pnsn_qual

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Data quality indicator for the Pn/Sn amplitude ratio measurement. A string of zeros indicates good data quality. Each position in the quality string indicates the state of data quality test. A 0 character indicates the data passed the test, a 1 indicates the data failed the test. Reading from left-to-right, the test indicators are ordered as follows: 1. Missing data - 1 indicates that Pn, Sn, Lg signal or noise values are not available for the calculation. 2. Signal-to-noise - 1 indicates that the snr for a given amplitude ratio does not satisfy the input criteria. 3. No distance correction - 1 indicates that there is no distance correction data available to apply to this amplitude ratio (see Attencoef). 4. Exceeds valid range of distance correction - 1 indicates the event is outside the valid range of applicability for the distance correction given in Attencoef. 5. No correction grid file - 1 indicates there is no correction grid file provided for the station under consideration. 6. Exceeds valid range of correction grid file - 1 indicates the event is outside the valid range of the correction grid file provided for the station. 7. Spare field (not currently used) - set to 0. 8. Anomalous data value - 1 indicates an anomalous regional phase amplitude measurement, invalidating the P/S measurement.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)		- (hyphen)	a10	Y
Range	any valid ORACLE string			

3.2.652. pocid

Table(s)	DATAUSER (IDCX), SUBSUSER (IDCX)			
Description	Point of contact identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	pocid > 0			

3.2.653. pointspike

Table(s)	QCSTATS (IDCX)			
Description	Amount of data in the detection processing interval masked due to point-spikes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999.0	f17.5	Y

Range	pointspike >= 0.0

3.2.654. position

Table(s)	CHANNAME (STATIC)			
Description	Not used by CD2WNG. The number of this channel in the mask column of the waveinterval and clf table. The first station channel should have a position zero, the second channel position one and so on.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	Y
Range	position >= 0			

3.2.655. position

Table(s)	NOOPS_CHANNAME (STATIC)			
Description	The number of this channel in the mask column of the waveinterval and clf table. The first station channel should have a position zero, the second channel position one and so on.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	Y
Range	position >= 0			

3.2.656. prefer_loc

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Preferred location identifier. This column indicates which of three possible location solutions is the chosen location. The hypocentre can be either held to a surface location (S), determined with no constraints at all (free depth, F), or restrained based on the settings of constrain_ot, constrain_latlon, and constrain_depth, (R). The constrained location (R) can be fixed in origin time/ latitude and longitude/depth. Prefer_loc takes precedence to the actual constraint settings (constrain_ot, constrain_latlon, and constrain_depth) when prefer_loc indicates a surface (S) or free depth (F) location. Default is "S."			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	prefer_loc IN {F, S, R}			

3.2.657. prefor

Table(s)	EVENT (LEB), EVENT (REB), EVENT (SEL1), EVENT (SEL2), EVENT (SEL3)			
Description	Preferred origin. This column holds the origin identifier (orid) that points to the preferred origin for a seismic event.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y

Range	prefor > 0

3.2.658. prid

Table(s)	PROBLEM (IDCX)			
Description	Problem identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	prid > 0			

3.2.659. prid

Table(s)	PROBLEMLOG (IDCX)			
Description	Problem identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	prid > 0			

3.2.660. primarykey

Table(s)	ARCH_DATA_TYPE (IDCX)			
Description	Not used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any string up to 24 characters			

3.2.661. priority

Table(s)	DATAUSER (IDCX)			
Description	Priority assigned to process.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(2,0)		NOT ALLOWED	i2	N
Range	priority > 0			

3.2.662. priority

Table(s)	SUBSUSER (IDCX)			
Description	Priority assigned to process.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(2,0)		-1	i2	Y
Range	priority > 0			

3.2.663. prob_weight_time

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Probability weighted time. The weight for a sample is the probability that the samples amplitude is the maximum amplitude associated with the arrival within the frequency band defined by low_cut, high_cut, ford, and ftype.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999.0	f17.5	Y
Range	any valid epoch time			

3.2.664. procclass

Table(s)	TIMESTAMP (IDCX)			
Description	Process class used to group processes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	N
Range	any string up to the character field width			

3.2.665. procclass

Table(s)	PROBLEMMAIL (IDCX)			
Description	Category of problem, such as Stations, Software, Message or Interactive.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(17)		NOT ALLOWED	a17	N
Range	any string up to the character field width			

3.2.666. procclass

Table(s)	PROBLEM (IDCX)			
Description	Category of problem, such as Stations, Software, Message or Interactive.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(17)		NOT ALLOWED	a17	Y
Range	any string up to the character field width			

3.2.667. procname

Table(s)	TIMESTAMP (IDCX)			
Description	Process name that identifies a process within a process class.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	N
Range	any string up to character limit			

3.2.668. procname

Table(s)	PROBLEM (IDCX)			
Description	Process name that identifies a process within a process class.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(17)		NOT ALLOWED	a17	Y
Range	any string up to 16 characters			

3.2.669. prodfmt

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	The format of the product. Must be one of the currently supported IDC product formats: GSE2.0, IMS1.0.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen)	a6	Y
Range	prodfmt IN { GSE2.0, IMS1.0 }			

3.2.670. prodid

Table(s)	EVCHAR_PROD (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)				N
Range				

3.2.671. prodid

Table(s)	PRODTRACK (IDCX)			
Description	Product identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	prodid > 0			

3.2.672. prodid

Table(s)	EVSC_PROD (IDCX), SUBS (IDCX)			
Description	Product identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	prodid > 0			

3.2.673. prodid

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Product identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i8	N
Range	prodid > 0			

3.2.674. prodid

Table(s)	PRODUCTTYPEEVPARS (IDCX)			
Description	Obsolete.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)				N
Range				

3.2.675. prodid

Table(s)	PRODUCTTYPEORIGIN (IDCX), PRODUCTTYPESTA (IDCX)			
Description	Product identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	N
Range	prodid > 0			

3.2.676. prodid

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Product identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	prodid > 0			

3.2.677. prodname

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Name of the product.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any valid string up to 24 characters			

3.2.678. prodsubtype

Table(s)	PRODUCTCRITERIA (IDCX)

Description	Subtype of the product. Typically corresponds to one of the IDC SHI event bulletins such as SEL1, SEL2, SEL3, LEB, REB etc.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(12)		- (hyphen)	a12	Y
Range	prodsubtype IN { sel1, sel2, sel3, reb, etc. }			

3.2.679. prodtype

Table(s)	FPDESCRIPTION (IDCX)			
Description	Product type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(12)		- (hyphen)	a12	Y
Range	prodtype IN {origin, event, bulletin, arrival, detection, waveform}			

3.2.680. prodtype

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Product type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	prodtype IN {origin, event, bulletin, arrival, detection, waveform}			

3.2.681. projection

Table(s)	MAPDISC (MAP)			
Description	Projection of the Map; a positive integer enumerator for uniquely classifying the stereographic projection of the Map. Azimuthal equidistant = 2; Mercator = 3.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	projection IN {2, 3}			

3.2.682. prop_mode

Table(s)	PHASE_DESCRIPTION (STATIC)			
Description	Indicates the propagation mode for this phase. One of P,S,H,I,V or N.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	N
Range	prop_mode IN { P,S,H,I,V,N }			

3.2.683. ptime

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)
Description	Epoch time at which P-type polarization columns are estimated. This column

	value is the centre of the time window with maximum rectilinearity.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-9999999999999	f17.5	Y
Range	any valid epoch time			

3.2.684. ptmcor

Table(s)	SITEAUX (STATIC)			
Description	P-wave arrival time correction.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f6.3	Y
Range	ptmcor > -999.0			

3.2.685. ptyp

Table(s)	CEPPKS (REB)			
Description	Consistent cepstral peak type. This column is FC-PHS if consistent Fourier cepstral peaks are found across two or more phases for one array and no peak is in the noise cepstrum at this quefrency. Otherwise, it is "-" if no consistent cepstral peaks are found.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen)	a6	Y
Range	ptyp IN {FC-PHS}			

3.2.686. qcstatsid

Table(s)	QCSTATS (IDCX)			
Description	Data quality statistics identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	qcstatsid > 0.0			

3.2.687. qfactor

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.688. qual

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)

Description	Onset quality. This single-character flag is used to denote the sharpness of the onset of a seismic phase. This relates to the timing accuracy as follows: i (impulsive) - accurate to +/-0.2 seconds e (emergent) - accuracy between +/-(0.2 to 1.0 seconds) w (weak) - timing uncertain to > 1 second			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	qual IN {i, e, w}			

3.2.689. ratio

Table(s)	SPLP (REB)			
Description	Ratio of short-period to long-period energy.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f10.8	Y
Range	ratio > 0			

3.2.690. ratiotype

Table(s)	ATTENCOEF (STATIC)			
Description	Amplitude ratio type. This identifies the type of P/S amplitude ratio (for example, Pn/Lg or Pn/Sn) for which the attenuation coefficients apply.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen)	a6	N
Range	any valid string up to 6 characters			

3.2.691. rcoefid

Table(s)	REGCOEF (STATIC)			
Description	Linear coefficient set identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		- (hyphen)	a20	N
Range	any valid string up to 20 characters			

3.2.692. rcoeftype

Table(s)	REGCOEF (STATIC)			
Description	Type of linear weighting coefficient used in regional P/S screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)		- (hyphen)	a10	N
Range	any valid string up to 10 characters			

3.2.693. rcoefvalue

Table(s)	REGCOEF (STATIC)			
Description	Value of linear weighting coefficient used in regional P/S screening criterion. Identified by rcoeftype.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	any valid floating point value			

3.2.694. reason

Table(s)	DISCARD (LEB), DISCARD (REB)			
Description	Reason an automated system event was discarded by an analyst.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)		NOT ALLOWED	a30	N
Range	any valid string up to the character width			

3.2.695. recid

Table(s)	DERVDISC (IDCX)			
Description	Unique record identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	recid > 0			

3.2.696. rect

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Signal rectilinearity defined as 1 - (l3 + l2)/2l1 where l1, l2, and l3 are the three eigenvalues from the decomposition of the covariance matrix. This column value is the maximum rectilinearity for all overlapping time windows.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.3	Y
Range	0.0 <= rect <= 1.0			

3.2.697. rectype

Table(s)	SPLP (REB)			
Description	Recipe type. This term identifies the algorithm used to calculate the parameter.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	N
Range	any string up to the character field width			

3.2.698. rectype

Table(s)	THIRDMOM (REB), TIMEFREQ (REB)			
Description	Recipe type. This term identifies the algorithm used to calculate the parameter.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	any string up to the character field width			

3.2.699. rectype

Table(s)	COMPLEXITY (REB)			
Description	Recipe type. This term identifies the algorithm used to calculate the parameter.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	any string up to the character field width			

3.2.700. reflat

Table(s)	MAPDISC (MAP)			
Description	Latitude reference; latitude of the center of the Map application.s projection (used for azimuthal equidistant projections only).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999	f9.4	Y
Range	-90.0 <= reflat <= 90.0			

3.2.701. reflon

Table(s)	MAPDISC (MAP)			
Description	Longitude reference. Longitude of the center of the Map application's projection (used for azimuthal equidistant projections only).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999	f9.4	Y
Range	-180.0 <= reflon <= 180.0			

3.2.702. refoffsetlat

Table(s)	MAPDISC (MAP)			
Description	Latitude offset reference. This column value is the reference (in pixels) from the lower left corner of the map to the center of the Map applications projection. In the case where the reference point is at the center of the map, the offsets are equal to half the map width and height. For azimuthal equidistant projections only.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	pixels	-1	f9.4	Y

Range	refoffsetlat > 0

3.2.703. refoffsetlon

Table(s)	MAPDISC (MAP)			
Description	Longitude offset reference. This column value is the reference (in pixels) from the lower left corner of the map to the center of the Map application.s projection. For azimuthal equidistant projections only.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	pixels	-1	f9.4	Y
Range	refoffsetlon > 0			

3.2.704. refsta

Table(s)	SITE (STATIC)			
Description	Reference station. This string specifies the reference station with respect to which array members are located (see deast, dnorth).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen)	a6	Y
Range	any sta from tables			

3.2.705. reg_conf

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Confidence level of the regional seismic P/S screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	0.0 < reg_conf < 1.0			

3.2.706. regexp

Table(s)	RULESET (STATIC)			
Description	Regular expression to describe the channels where the rule applies.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		NOT ALLOWED	a64	N
Range	any alphanumeric string up to the character field width			

3.2.707. regname

Table(s)	PRODUCTTYPEEVSC (IDCX)			
Description	Region name for subscriptions.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y

Range	any valid string up to 24 characters

3.2.708. rely

Table(s)	SITEAUX (STATIC)			
Description	Station reliability. This column is an estimate of the percentage of time that the station is up.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f5.2	Y
Range	0.0 <= rely <= 1.0			

3.2.709. remark

Table(s)	REMARK (IDCX), REMARK (LEB), REMARK (REB), REMARK (SEL1), REMARK (SEL2), REMARK (SEL3), REMARK (STATIC)			
Description	Descriptive text. This single line of text is an arbitrary comment about a record in the database. The comment is linked to its parent table only by forward reference from commid in the record of the table of interest (see commid, lineno).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(80)		- (hyphen)	a80	Y
Range	any string up to 80 characters			

3.2.710. req_azslo_dist

Table(s)	STATION_PHASE_RELY (STATIC)			
Description	Maximum distance from hypocenter to disregard azimuth and slowness residuals			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)	degrees	-1	i8	Y
Range	dis_dist >= 0.0			

3.2.711. req_azslo_snr

Table(s)	STATION_PHASE_RELY (STATIC)			
Description	Minimum signal-to-noise ratio required to disregard azimuth and slowness residuals			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	dis_allow >= 0.0			

3.2.712. reqid

Table(s)	REQUEST (IDCX)
Description	Request identifier. Unique key to allow tracking of requests by the Message

	Subsystem.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	reqid > 0			

3.2.713. requestor

Table(s)	REQUEST (IDCX)			
Description	Original requestor of this data. The requestor is the person or program that requests this waveform data.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any string up to 15 characters			

3.2.714. revfunction

Table(s)	REVAUDIT (LEB)			
Description	Name of the revision function used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		NOT ALLOWED	a32	N
Range	any function name in rebrevise			

3.2.715. revid

Table(s)	REVAUDIT (LEB)			
Description	Revision identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	revid > 0			

3.2.716. revision

Table(s)	CHANNAME (STATIC)			
Description	Link between the clf and channame table to indicate which position values in the channame table should be used. This column is needed because channels may be added and removed from a station. The channame table stores each combination of channels using different revision values. This allows historical data to be analysed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	N
Range	revision > 0			

3.2.717. revision

Table(s)	FILEPRODUCT (IDCX)			
Description	Revision number of the data file.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		-1	i4	Y
Range	revision > 0			

3.2.718. revision

Table(s)	NOOPS_CHANNAME (STATIC)			
Description	link between the clf and channame table to indicate which position values in the channame table should be used. This column is needed because channels may be added and removed from a station. The channame table stores each combination of channels using different revision values. This allows historical data to be analysed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	Y
Range	revision > 0			

3.2.719. revision

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)			i4	Y
Range				

3.2.720. revision

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)				Y
Range				

3.2.721. revstate

Table(s)	REVAUDIT (LEB)			
Description	State of the revision.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		NOT ALLOWED	a16	Y
Range	revstate IN {DONE|FAILED|ABORTED}			

3.2.722. revtagid1

Table(s)	REVAUDIT (LEB)			
Description	Value of the foreign key named in revtagname1.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	revtagid1 > 0			

3.2.723. revtagid2

Table(s)	REVAUDIT (LEB)			
Description	Value of the foreign key named in revtagname2.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	revtagid2 > 0			

3.2.724. revtagname1

Table(s)	REVAUDIT (LEB)			
Description	Name of the foreign key whose value is in revtagid1.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	revtagname1 IN {valid identifier}			

3.2.725. revtagname2

Table(s)	REVAUDIT (LEB)			
Description	Name of the foreign key whose value is in revtagid2.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	revtagname2 IN {valid identifier}			

3.2.726. rid

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB)			
Description	Recipe identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	N
Range	any string up to eight characters long			

3.2.727. rmsamp

Table(s)	PMCC_FEATURES (IDCX)

Description	Mean value of the root mean square amplitudes of the detection pixels in the family (Pa)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.728. rotation

Table(s)	MAPDISC (MAP)			
Description	Map rotation. This is the rotation of the projection from 0?, or due north. Rotation specifies the azimuth of the y-raster in degrees clockwise from north (for azimuthal equidistant projections only).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-1	f9.4	Y
Range	0.0 <= rotation < 360.0			

3.2.729. rscore

Table(s)	EVSC_PROD (IDCX)			
Description	Score for the regional seismic P/S event-screening criterion.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	rscore > -999.0			

3.2.730. rsptype

Table(s)	INSTRUMENT (STATIC)			
Description	Instrument response type. The column value denotes the style in which detailed calibration data are stored. The neighboring column dfile tells where the calibration data are saved. Rsptype = paz indicates the data are the poles and zeros of the Laplace transform. Rsptype = fap indicates the data are amplitude/phase values at a range of frequencies. Rsptype = fir indicates that the response type is a finite impulse response table. Rsptype = pazfir indicates a combination of poles, zeros, and finite impulse response. Other codes may be defined.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	Y
Range	any lower-case string up to six characters			

3.2.731. running

Table(s)	DLMAN (IDCX)			
Description	Flag indicating whether or not DLMan is running.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y

Range	running IN {y, n}

3.2.732. samprate

Table(s)	INSTRUMENT (STATIC)			
Description	Sampling rate. This column is the sample rate in samples/second. The column value is specifically the nominal sample rate, not accounting for clock drift.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	samples/sec	NOT ALLOWED	f11.7	Y
Range	samprate > 0.0			

3.2.733. samprate

Table(s)	OLD_WFPROTO (IDCX), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Sampling rate. This column is the sample rate in samples/second. The value may vary slightly from the nominal to reflect clock drift.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	samples/sec	NOT ALLOWED	f11.7	Y
Range	samprate > 0.0			

3.2.734. scale

Table(s)	MAPDISC (MAP)			
Description	Map scale.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	radians per pixel for mercator projections; kilometers per pixel for azimuthal equidistant projections	-1	f9.4	Y
Range	scale > 0			

3.2.735. score

Table(s)	EVSC_PROD (IDCX)			
Description	Composite event-screening score. This score numerically indicates the degree to which a particular event meets or does not meet the combined screening criteria.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	score > -999.0			

3.2.736. sdepth

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORI-

	GERR_TEMP_GA (SEL3)			
Description	Depth error. This is the maximum error of a depth estimate for a level of confidence given by conf (see smajax, sminax, and sxx, syy, szz, stt, sxy, sxz, syz, stx, sty, stz).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-1.0	f9.4	Y
Range	sdepth > 0.0			

3.2.737. sdobs

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Standard error of one observation. This column is derived from the discrepancies in the arrival times of the phases used to locate an event. This column is defined as the square root of the sum of the squares of the time residuals divided by the number of degrees of freedom. The latter is the number of defining observations (ndef in Origin (Origin_temp_ga)) minus the dimension of the system solved (4 if depth is allowed to be a free variable, 3 if depth is constrained).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f9.4	Y
Range	sdobs > 0.0			

3.2.738. seaz

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)			
Description	Station-to-event azimuth calculated from the station and event locations and measured clockwise from north.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f7.2	Y
Range	0.0 <= seaz <= 360.0			

3.2.739. seazlr

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Azimuth of the eigenvector (e3) associated with the smallest eigenvalue (l3). This column is corrected by 180° to give an estimate of the station-to-event azimuth (with an 180° ambiguity). This column is an S-type column calculated at the time of the maximum 3-component amplitude. The only difference between seazs and seazlr is in the definition of the overlapping time windows.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f7.2	Y
Range	0.0 <= seazlr <= 360.0			

3.2.740. seazp

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Azimuth of the eigenvector (e1) associated with the largest eigenvalue (l1). The column value is corrected by 180° to give an estimate of the station-to-event azimuth. This P-type value is calculated at the time of maximum rectilinearity.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f7.2	Y
Range	0.0 <= seazp <= 360.0			

3.2.741. seazs

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	Azimuth of the eigenvector (e3) associated with the smallest eigenvalue (l3). This column is corrected by 180° to give an estimate of the station-to-event azimuth (with an 180° ambiguity). This column is an S-type column calculated at the time of the maximum 3-component amplitude. The only difference between seazs and seazlr is in the definition of the overlapping time windows.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees	-999.0	f7.2	Y
Range	0.0 <= seazs <= 360.0			

3.2.742. secondkey

Table(s)	ARCH_DATA_TYPE (IDCX)			
Description	Not used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any string up to 24 characters			

3.2.743. segtype

Table(s)	OLD_WFPROTO (IDCX)			
Description	Flag indicating whether the waveform segment is original (o), virtual (v), segmented (s) or duplicated (d).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)		- (hyphen)	a1	Y
Range	segtype IN {o, v, s, d}			

3.2.744. segtype

Table(s)	WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)
Description	Flag indicating whether the waveform segment is original (o), virtual (v), segmented (s) or duplicated (d).

Storage type	Unit	NA value	External	Nullable
varchar2(1)		- (hyphen)	a1	Y
Range	segtype IN {o, v, s, d}			

3.2.745. servicetime

Table(s)	DATAUSER (IDCX)			
Description	Last time a request from the user with the userid in the datauser table was serviced.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999999999999	f17.5	Y
Range	servicetime >= 0			

3.2.746. sigma_time

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Standard deviation of the probability weighted time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)		-999.0	f17.5	Y
Range	sigma_time >= 0			

3.2.747. sigmafreq

Table(s)	PMCC_FEATURES (IDCX)			
Description	Standard deviation of the frequency values of the detection pixels in the family (Hz)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f10.4	Y
Range				

3.2.748. sigtype

Table(s)	MSGDISC (IDCX)			
Description	Digital signature type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		- (hyphen)	a64	Y
Range	ASCII character string			

3.2.749. skewness

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)

Description	Skew of the estimated signal energy between onset_time and termination_time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-9999999999999	f9.4	Y
Range	-10000000.0 < skewness < 10000000.0			

3.2.750. slo_hi

Table(s)	BREGION_TAB (STATIC)			
Description	Upper bound of the slowness window.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	seconds/degree		f8.3	Y
Range	slo_hi > 0.0			

3.2.751. slo_lo

Table(s)	BREGION_TAB (STATIC)			
Description	Lower bound of the slowness window.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)	seconds/degree		f8.3	Y
Range	slo_lo > 0.0			

3.2.752. slo_rel

Table(s)	STATION_PHASE_RELY (STATIC)			
Description	Slowness reliability estimate			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)		-999.0	f8.3	Y
Range	0.0 <= slo_rel <= 1.0			

3.2.753. slodef

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Slowness defining code. This one-character flag indicates whether or not the slowness of a phase was used to constrain the event location. This column is defining (slodef = d) or non defining (slodef = n) for this arrival.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	slodef IN {d, n}			

3.2.754. slores

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC)			
Description	Slowness residual. This column gives the difference between an observed slowness and a theoretical prediction. The prediction is calculated for the related phase and event origin described in the record.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/kilometer	-999.0	f7.2	N
Range	slores > -999.0			

3.2.755. slores

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Slowness residual. This column gives the difference between an observed slowness and a theoretical prediction. The prediction is calculated for the related phase and event origin described in the record.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/kilometer	-999.0	f7.2	Y
Range	slores > -999.0			

3.2.756. slores

Table(s)	ALLOW_RESID (LEB)			
Description	The maximum allowable difference between an observed slowness and a theoretical prediction.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/degree	-999.0	f7.2	Y
Range	slores > -999.0			

3.2.757. slow

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Observed slowness of a detected arrival. Units are seconds/degree.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/degree	-1.0	f7.2	Y
Range	slow >= 0.0			

3.2.758. slow

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), HYDRO_ARR_GROUP (IDCX), HYDRO_ARR_GROUP (LEB), HYDRO_ARR_GROUP (REB)
Description	Observed slowness of an arrival. Units are seconds/kilometer.

Storage type	Unit	NA value	External	Nullable
float(24)	seconds/kilometer	-1.0	f7.2	Y
Range	slow >= 0.0			

3.2.759. slow

Table(s)	PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3)			
Description	Slowness of a predicted arrival. Units are seconds/degree.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds/degree	-1.0	f7.2	Y
Range	slow >= 0.0			

3.2.760. slow_rely

Table(s)	STA_RELY (STATIC)			
Description	Slowness reliability flag. The value is a single-character flag to indicate whether (y) or not (n) the slowness is considered to be reliable.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	Y
Range	slow_rely IN { 0,1}			

3.2.761. slow_weight

Table(s)	PHASE_WEIGHT_EVENT_DEFINE (STATIC)			
Description	Weight given to defining slowness for this phase			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	N
Range				

3.2.762. smaj_sc

Table(s)	EVSC_PROD (IDCX)			
Description	Semi-major axis of error ellipse, used for onshore/offshore determination, for a confidence level given by loc_conf of table PRODUCTTYPEEVSC.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f7.2	Y
Range	smaj_sc > 0.0			

3.2.763. smajax

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)

Description	Semi-major axis of error ellipse for a given confidence. The column value is the length of the semi-major axis of the location error ellipse. The value is found by projecting the covariance matrix onto the horizontal plane. The level of confidence is specified by conf (see sdepth, sminax, and sxx, syy, szz, stt, sxy, sxz, syz, stx, sty, stz).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-1.0	f9.4	Y
Range	smajax > 0.0			

3.2.764. smin_sc

Table(s)	EVSC_PROD (IDCX)			
Description	Semi-minor axis of error ellipse, used for onshore/offshore determination, for a confidence level given by loc_conf of table PRODUCTTYPEEVSC.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-999.0	f7.2	Y
Range	smin_sc > 0.0			

3.2.765. sminax

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Semi-minor axis of error ellipse. The column value is the length of the semi-minor axis of the location error ellipse. The value is found by projecting the covariance matrix onto the horizontal plane. The level of confidence is specified by conf (see sdepth, smajax, and sxx, syy, szz, stt, sxy, sxz, syz, stx, sty, stz).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	kilometers	-1.0	f9.4	Y
Range	sminax > 0.0			

3.2.766. sn_snr

Table(s)	EVSC_REGIONAL (IDCX)			
Description	Signal-to-noise ratio of the Sn measurement.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	sn_snr > 0.0			

3.2.767. snr

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB), APMA (IDCX), APMA (LEB), APMA (REB), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), COMPLEXITY (REB), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)

Description	Signal-to-noise ratio. This is an estimate of the ratio of the amplitude of the signal to amplitude of the noise immediately preceding it. For apma, this value is based on the maximum 3-component amplitudes (see amps). This column is the average signal-to-noise ratio for the frequency bands that contributed to the final polarization estimates.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f10.2	Y
Range	snr > 0.0			

3.2.768. snr_hi

Table(s)	BREGION_TAB (STATIC)			
Description	Upper bound of the snr window.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)		-1	f8.3	Y
Range				

3.2.769. snr_high_band

Table(s)	EVSC_HYDRO (IDCX)			
Description	Hydroacoustic signal-to-noise ratio in the high-frequency band (32-64 Hz) from table Amplitude for a given orid/sta pair.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	snr_high_band > 0.0			

3.2.770. snr_lo

Table(s)	BREGION_TAB (STATIC)			
Description	Lower bound of the snr window.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,3)		-1	f8.3	Y
Range				

3.2.771. snthrsh

Table(s)	SITEAUX (STATIC)			
Description	Nominal signal-to-noise ratio.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f5.2	Y
Range	snthrsh > 1.0			

3.2.772. sound_speed

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.773. sp_tol1

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.774. sp_tol2

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.775. speed_transition

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.776. spike

Table(s)	QCSTATS (IDCX)			
Description	Amount of data in the detection processing interval masked due to spikes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-999.0	f17.5	Y
Range	spike >= 0.0			

3.2.777. sproid

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)

Description	Uniquely identifies a set of parameters used in the signal processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	sproid > 0			

3.2.778. src_dpnt_corr

Table(s)	EVENT_CONTROL (LEB), EVENT_CONTROL (REB), IN_EVENT_CONTROL (LEB)			
Description	Identification of whether or not, and what type of, source-dependent corrections were applied to the location. 0 = No source-dependent corrections applied to the event location. 1 = Test-site travel-time corrections applied to the event location. 2 = Source-Region Station-Timing (SRST) corrections applied to the event location. 3 = Regional level source-specific station corrections (SSSC) applied to the event location. SRST correction is not applied, even if it exists. 4 = Local level SSSCs applied to the location. SRST correction is not applied, even if it exists.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(2,0)		0	i2	Y
Range	src_dpnt_corr IN {0, 1, 2, 3, 4}			

3.2.779. srn

Table(s)	SREGION (STATIC)			
Description	Seismic region number, as given in [Fli74] (see grn, grname, and srname).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	srn > 0			

3.2.780. srn

Table(s)	ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3)			
Description	Seismic region number, as given in [Fli74] (see grn, grname, and srname).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	srn > 0			

3.2.781. srname

Table(s)	SREGION (STATIC)			
Description	Seismic region name. This column is the common name of a seismic region, as given in [Fli74]. Names may have changed due to changing political circumstances (see srn and grname).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(40)		NOT ALLOWED	a40	Y
Range	any upper-case string up to 40 characters			

3.2.782. sta

Table(s)	AFFILIATION (STATIC), AFFILIATION_EXCL (STATIC), AFFILIATION_SUB (STATIC), ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), ATTENCOEF (STATIC), BEAMAUX (STATIC), BREGION_TAB (STATIC), CEPPKS (REB), CHAN_GROUPS (STATIC), COMPLEXITY (REB), DERVDISC (IDCX), EVSC_HYDRO (IDCX), EVSC_REGIONAL (IDCX), FORBEAMAUX (STATIC), OLD_WFPROTO (IDCX), OUTAGE (IDCX), PARTICIPATION (STATIC), PMCC_FEATURES (IDCX), PRODUCTTYPESTA (IDCX), QCSTATS (IDCX), REQUEST (IDCX), RULESET (STATIC), SEMAPHORE (IDCX), SENSOR (STATIC), SITE (STATIC), SITEAUX (STATIC), SITECHAN (STATIC), SITEPOLL (STATIC), SITEPOLLCA (STATIC), SITE_ADDRESS (STATIC), SPLP (REB), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3), STANET (STATIC), STATION_PHASE_RELY (STATIC), STA_RELY (STATIC), THIRDMOM (REB), TIMEFREQ (REB), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Station code. This column value is the code name of a seismic, hydroacoustic, or infrasonic observatory and identifies a geographic location recorded in the SITE table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any upper-case string up to the character field width			

3.2.783. sta

Table(s)	WFCONV (IDCX)			
Description	Station code. This column value is the code name of a seismic, hydroacoustic, or infrasonic observatory and identifies a geographic location recorded in the SITE table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED - (hyphen) for assoc, chan_groups, dlfile, fileproduct, fkdisc, fsdisc, qcstats, stassoc, thirdmom	a6	N
Range	any upper-case string up to the character field width			

3.2.784. sta

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), FILEPRODUCT (IDCX), STASSOC (IDCX)

Description	Station code. This column value is the code name of a seismic, hydroacoustic, or infrasonic observatory and identifies a geographic location recorded in the SITE table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen)	a6	Y
Range	any upper-case string up to the character field width			

3.2.785. sta

Table(s)	AFFILIATION_LP (STATIC), CHAN_GROUPS_AUXWF (STATIC), DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), FS_STAGE-PRODUCT (IDCX), PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3)			
Description	Station code. This column value is the code name of a seismic, hydroacoustic, or infrasonic observatory and identifies a geographic location recorded in the SITE table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	Y
Range	any upper-case string up to the character field width			

3.2.786. sta

Table(s)	FKDISC (LEB)			
Description	Station code. This column value is the code name of a seismic, hydroacoustic, or infrasonic observatory and identifies a geographic location recorded in the SITE table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED - (hyphen) for assoc, chan_groups, dlfile, fileproduct, fkdisc, fsdisc, qcstats, stassoc, thirdmom	a6	Y
Range	any upper-case string up to the character field width			

3.2.787. sta

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)				Y
Range				

3.2.788. staname

Table(s)	SITE (STATIC)			
Description	Station name/description. This column value is the full name of the station whose codename is in sta. As an example, one record in the site table connects sta = ANMO to staname = ALBUQUERQUE, NEW MEXICO (SRO).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)		- (hyphen)	a50	Y
Range	any upper-case string up to 50 characters			

3.2.789. staper

Table(s)	SITEAUX (STATIC)			
Description	Standard period at which noise estimates are made.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f5.2	Y
Range	staper > 0.0			

3.2.790. start_time

Table(s)	REQUEST (IDCX)			
Description	Epoch start time of the data interval. Epochal time is given as seconds since hour 0 January 1, 1970.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	N
Range	any valid epoch time			

3.2.791. start_time

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Epoch start time of the data interval. Epochal time is given as seconds since hour 0 January 1, 1970.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	NOT ALLOWED	f17.5	Y
Range	any valid epoch time			

3.2.792. stassid

Table(s)	STASSOC (IDCX)
Description	Station association identification. The wavetrain from a single event may be made up of a number of arrivals. A unique stassid joins those arrivals believed to have come from a common event as measured at a single station. Stassid is also the key to the stassoc table, which contains additional signal measurements not contained within the arrival table, such as station magnitude estimates and computed signal characteristics.

Storage type	Unit	NA value	External	Nullable
number(10,0)		NOT ALLOWED	i10	N
Range	stassid > 0			

3.2.793. stassid

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Station association identification. The wavetrain from a single event may be made up of a number of arrivals. A unique stassid joins those arrivals believed to have come from a common event as measured at a single station. Stassid is also the key to the stassoc table, which contains additional signal measurements not contained within the arrival table, such as station magnitude estimates and computed signal characteristics.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	stassid > 0			

3.2.794. stat_prov

Table(s)	SITE_ADDRESS (STATIC)			
Description	State or province of a country.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)		- (hyphen)	a40	Y
Range	any string up to 40 characters			

3.2.795. state

Table(s)	REBDONE_DATADAY_FLAG (LEB)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(12)		- (hyphen)	a12	Y
Range	Not used			

3.2.796. state

Table(s)	INTERVAL (IDCX), INTERVAL (SEGMENT), REQUEST (IDCX)			
Description	Processing state. The processing state of the interval within the automated processing system.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y
Range	a set of strings defined at each installation for each automated processing system. For example, done, ready, queued, requested, timeout, running.			

3.2.797. state

Table(s)	GA_TAG (SEL1)			
Description	The state in the ga_tag table is used by GAassoc and GAconflict to recognize arrivals from auxiliary stations and handle them appropriately.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	Y
Range	any state currently recognized by GA applications			

3.2.798. state_count

Table(s)	MSGAUX (IDCX)			
Description	Number of failures of the message system to process a request.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(4,0)		NOT ALLOWED	i4	N
Range	state_count >= 0			

3.2.799. statecount

Table(s)	REQUEST (IDCX)			
Description	Count of failures. When state = failed, the statecount column records the number of failures to acquire this data.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	Y
Range	statecount >= 0			

3.2.800. station

Table(s)	CHANNAME (STATIC)			
Description	Station name.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any upper-case character string up to the field width			

3.2.801. station

Table(s)	NOOPS_CHANNAME (STATIC)			
Description	Station name. Provides the link between the channame and waveinterval tables.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		NOT ALLOWED	a6	N
Range	any upper-case character string up to the field width			

3.2.802. status

Table(s)	PRODTRACK (IDCX)			
Description	Status of the product, as related to the operation of Subscription system. Rows for which there are currently no active subscription have status NOACTIVE. Successfully delivered subscriptions have status DONE. NEW indicates new entries, with QUEUED indicating that there is some backlog of subscriptions to be processed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(12)		- (hyphen)	a12	Y
Range	status IN {NEW, DONE, NOACTIVE, QUEUED}			

3.2.803. status

Table(s)	DATAUSER (IDCX)			
Description	Flag indicating whether the user is currently active or inactive for requests and subscriptions. For subsuser possible values are a (active) or I (inactive). For datauser, possible values are ACTIVE or INACTIVE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		NOT ALLOWED	a24	N
Range	datauser status IN { ACTIVE, INACTIVE }. subsuser IN { a, I }			

3.2.804. status

Table(s)	SUBSUSER (IDCX)			
Description	Flag indicating whether the user is currently active or inactive for requests and subscriptions. For subsuser possible values are a (active) or I (inactive). For datauser, possible values are ACTIVE or INACTIVE.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	datauser status IN { ACTIVE, INACTIVE }. subsuser IN { a, I }			

3.2.805. status

Table(s)	MSGDATATYPE (IDCX), MSGDISC (IDCX)			
Description	Status of the current message. The NA value for all these status fields is - (hyphen), except in the MSGDATATYPE table where it is limited to DONE or FAILED.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	msgdatatype status IN { DONE, FAILED }, otherwise any character string up to the caracter field width.			

3.2.806. status

Table(s)	OUTAGE (IDCX)			
Description	Status of the current message. The NA value for all these status fields is - (hyphen), except in the msgdatatype table where it is limited to DONE or FAILED.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	msgdatatype status IN { DONE, FAILED }, otherwise any character string up to the caracter field width.			

3.2.807. status

Table(s)	MSGDEST (IDCX)			
Description	Status of the current message. The NA value for all these status fields is - (hyphen), except in the MSGDATATYPE table where it is limited to DONE or FAILED.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	msgdatatype status IN { DONE, FAILED }, otherwise any character string up to the caracter field width.			

3.2.808. status

Table(s)	PROBLEM (IDCX)			
Description	The status field indicates whether the problem is currently open or closed.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(33)		- (hyphen)	a33	Y
Range	status IN {open, closed}			

3.2.809. status

Table(s)	SUBS (IDCX)			
Description	(Subs): Status of the subscription, either a (active) or I (inactive). (Waveinterval): status of the time interval specified by time and endtime), currently always set to NEW.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)		- (hyphen)	a6	Y
Range	subs status in { a, i }, waveinterval status IN { NEW }			

3.2.810. status

Table(s)	DATAREADY (IDCX)
Description	Possible values for dataready are: n=new, i=insert, c=change, d=delete or -(hyphen). For ftpfailed, possible values are RETRY or FAILED.

Storage type	Unit	NA value	External	Nullable
varchar2(8)		- (hyphen)	a8	Y
Range	any valid string up to the character limit			

3.2.811. status

Table(s)	FTPFAILED (IDCX)			
Description	Status of FTP attempt (retry or failed)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		- (hyphen)	a8	Y
Range	any valid string up to the character limit			

3.2.812. statype

Table(s)	STATION_PHASE_RELY (STATIC)			
Description	Type of station			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any valid string up to 24 characters			

3.2.813. statype

Table(s)	ALLOW_RESID (LEB), DEFINE_PHASE_RESIDUAL (STATIC), WEIGHTS (STATIC)			
Description	Station type. This character string specifies the station type. Recommended entries are ss (single station) or ar (array).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	N
Range	statype IN {ss, ar}			

3.2.814. statype

Table(s)	SITE (STATIC)			
Description	Station type. This character string specifies the station type. Recommended entries are ss (single station) or ar (array).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	Y
Range	statype IN {ss, ar}			

3.2.815. stav

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB)
Description	Short-term average used to describe the amplitude of a signal. The amplitude is

	averaged over a small time interval, typically 1-2 seconds. This time window is defined in Sigpro processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	nanometers	-1.0	a18	Y
Range	stav > 0.0			

3.2.816. stdconstval

Table(s)	QCSTATS (IDCX)			
Description	Standard deviation of data in masked constant segments.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)		-999.0	f17.5	Y
Range	stdconstval >= 0.0			

3.2.817. stime

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Origin time error in seconds.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f8.2	Y
Range	stime >= 0.0			

3.2.818. stime

Table(s)	APMA (IDCX), APMA (LEB), APMA (REB)			
Description	This column is the epoch time at which S-type polarization columns are estimated. The value is the centre of the time window with the maximum 3-component amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds since 01-Jan-1970 00:00:00	9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.819. stime

Table(s)	MSGDEST (IDCX)			
Description	This column is the epochal time at which the corresponding message was sent.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds since 01-Jan-1970 00:00:00	9999999999.999	f17.5	Y
Range	any valid epochal time			

3.2.820. stmcor

Table(s)	SITEAUX (STATIC)			
Description	S-wave arrival time correction.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f6.3	Y
Range	stmcor > -999.0			

3.2.821. stream

Table(s)	CHANNAME (STATIC)			
Description	The connection name used in the data transmission. This is the name used in the CD Receiver output files (e.g. ARCES.20080124.20-24.w) In most cases, the stream name is the same as the station name. But for some stations the connection name differs from the station name expected by the processing applications that read CD2WNG output. For example, SONM has a connection name SONMS. This column also links to the sta in the FRAMEPRODUCT table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)			a6	N
Range				

3.2.822. stream

Table(s)	NOOPS_CHANNAME (STATIC)			
Description	the connection name used in the data transmission. This is the name used in the CD Receiver output files (e.g. ARCES.20080124.20-24.w) In most cases, the stream name is the same as the station name. But for some stations the connection name differs from the station name expected by the processing applications that read cd2w's output. For example, SONM has a connection name SONMS. This column also links to the sta in the frameproduct table.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)			a6	N
Range				

3.2.823. strike

Table(s)	EVSC_PROD (IDCX), ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Strike of semi-major axis of error ellipse. This column is the strike of the semi-major axis of the location error ellipse, measured in degrees clockwise from the North (see smajax).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	degrees (clockwise from North)	-1.0	f6.2	Y

Range	0.0 <= strike <= 360.0

3.2.824. strip

Table(s)	WFCONV (IDCX)			
Description	Flag showing whether or not the data are stripped of headers (y/n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	strip IN {n, y}			

3.2.825. stt

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	stt - seconds squared (sec2)	-1.0	f15.4	Y
Range	stt > 0.0			

3.2.826. stx

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	stx - kilometers per second (km/sec)	-1.0	f15.4	Y
Range				

3.2.827. sty

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			

Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	sty - kilometers per second (km/sec)	-1.0	f15.4	Y
Range				

3.2.828. stype

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB)			
Description	Signal type. This single-character flag indicates the event or signal type. The following definitions hold: l = local event r = regional event t = teleseismic event m = mixed or multiple event g = glitch (for example, non-seismic detection) e = calibration activity obfuscated the data l, r, and t are supplied by the reporting station or as an output of post-detection processing. g and c come from analyst comment or from status bits from GDSN and RSTN data.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	stype IN {l, r, t, m, g, c}			

3.2.829. stz

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	stz - kilometers per second (km/sec)	-1.0	f15.4	Y
Range				

3.2.830. sub_status

Table(s)	MSGAUX (IDCX)			
Description	Cause of failure.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		NOT ALLOWED	a24	Y

Range	any string up to 24 characters

3.2.831. subject

Table(s)	MSGDISC (IDCX)			
Description	Subject header from an email message.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)		- (hyphen)	a64	Y
Range	any string up to 64 characters			

3.2.832. subsid

Table(s)	SUBS (IDCX)			
Description	Subscription identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	subsid > 0			

3.2.833. subsname

Table(s)	SUBS (IDCX)			
Description	Subscription name, which is defined by subscriber.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		- (hyphen)	a24	Y
Range	any string up to 24 characters			

3.2.834. subtype

Table(s)	MSGDISC (IDCX)			
Description	Specification of whether or not the request includes waveforms. In the future, this column may contain indications of other message subtypes.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)		- (hyphen)	a2	Y
Range	subtype IN {V, R, L}			

3.2.835. supressempty

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Flag indicating whether (y) or not (n) VDMS should send out empty subscription messages when no data are available (Not currently used at the IDC).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(2)		- (hyphen)	a2	Y
Range	suppressempty IN { y, n }			

3.2.836. svar

Table(s)	SPVAR (REB)			
Description	Variance of the detrended log spectrum between fmin and fmax. The spectrum is measured in nm-sec.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f12.6	Y
Range	any floating point value			

3.2.837. sweight

Table(s)	WEIGHTS (STATIC)			
Description	Slowness weight.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		0.0	f5.2	Y
Range	sweight >= 0.0			

3.2.838. sxx

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	sxx - kilometers squared (km2)	-1.0	f15.4	Y
Range	sxx > 0.0			

3.2.839. sxy

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	sxy - kilometers squared	-1.0	f15.4	Y

	(km2)			
Range				

3.2.840. sxz

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	sxz - kilometers squared (km2)	-1.0	f15.4	Y
Range				

3.2.841. syy

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	syy - kilometers squared (km2)	-1.0	f15.4	Y
Range	syy > 0.0			

3.2.842. syz

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(24)	syz - kilometers squared (km2)	-1.0	f15.4	Y
Range				

3.2.843. szz

Table(s)	ORIGERR (IDCX), ORIGERR (LEB), ORIGERR (REB), ORIGERR (SEL1), ORIGERR (SEL2), ORIGERR (SEL3), ORIGERR_TEMP_GA (SEL2), ORIGERR_TEMP_GA (SEL3)			
Description	Elements of the covariance matrix for the location identified by orid. The covariance matrix is symmetric (and positive definite) so that sxy = syx, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with sdobs, ndef, and dtype) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	szz - kilometers squared (km2)	-1.0	f15.4	Y
Range	szz > 0.0			

3.2.844. table_name

Table(s)	ARCH_DATA_TYPE (IDCX)			
Description	Name of table encapsulating the data type.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	any string up to the character field width that is a valid table name			

3.2.845. tablename

Table(s)	DATAREADY (IDCX)			
Description	Database table name containing data ready to be distributed by the Subscription Subsystem.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		NOT ALLOWED	a24	Y
Range	any string up to the character field width that is a valid table name			

3.2.846. tagid

Table(s)	WFTAG (IDCX), WFTAG (SEL1), WFTAG (SEL2), WFTAG (SEL3)			
Description	Tagname value. This column contains the value of a foreign key identified in tagname. For example, if tagname is arid, then wftag may be joined to Arrival where arrival.arid = wftag.tagid. If tagname is orid, then wftag and Origin (Origin_temp_ga) may be joined where origin.orid = wftag.tagid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number(10,0)		NOT ALLOWED	i10	N
Range	tagid > 0			

3.2.847. tagid

Table(s)	DATAREADY (IDCX)			
Description	Tagname value. This column contains the value of a foreign key identified in tagname. For example, if tagname is arid, then wftag may be joined to Arrival where ARRIVAL.arid = WFTAG.tagid. If tagname is orid, then wftag and Origin (Origin_temp_ga) may be joined where ORIGIN.orid = WFTAG.tagid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	tagid > 0			

3.2.848. tagid

Table(s)	WFTAG (SEGMENT)			
Description	Tagname value. This column contains the value of a foreign key identified in tagname. For example, if tagname is arid, then wftag may be joined to Arrival where arrival.arid = wftag.tagid. If tagname is orid, then wftag and Origin (Origin_temp_ga) may be joined where origin.orid = wftag.tagid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i10	N
Range	tagid > 0			

3.2.849. tagid2

Table(s)	DATAREADY (IDCX)			
Description	Secondary tag value used by the Subscription Subsystem for data types that require more than one identifier (for example, time range for origin).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	Y
Range	tagid2 > 0			

3.2.850. tagname

Table(s)	DATAREADY (IDCX)			
Description	Tagname type. This column value is the name of the foreign key whose value is in tagid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(12)		NOT ALLOWED	a12	Y
Range	any character string up to the field width			

3.2.851. tagname

Table(s)	WFTAG (IDCX), WFTAG (SEGMENT), WFTAG (SEL1), WFTAG (SEL2), WFTAG (SEL3)			
Description	Tagname type. This column value is the name of the foreign key whose value is in tagid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	N
Range	tagname IN {arid, evid, orid, stassid, msgid}			

3.2.852. termination_time

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Estimated termination time of signal.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-9999999999999	f17.5	Y
Range	any valid epoch time			

3.2.853. thatdb

Table(s)	XTAG (IDCX)			
Description	Database account for the records specified by thatname and thatid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	any character string up to 32 characters that is a valid account name			

3.2.854. thatid

Table(s)	XTAG (IDCX)			
Description	Identifier for thatname.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	thatid > 0			

3.2.855. thatname

Table(s)	XTAG (IDCX)			
Description	Key for thatid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	Y
Range	thatname IN {arid, orid, wfid .}			

3.2.856. thisdb

Table(s)	XTAG (IDCX)			
Description	Database account for the records specified by thisname and thisid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)		- (hyphen)	a32	Y
Range	any character string up to 32 characters that is a valid account name			

3.2.857. thisid

Table(s)	XTAG (IDCX)			
Description	Identifier for thisname.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	Y
Range	thisid > 0			

3.2.858. thisname

Table(s)	XTAG (IDCX)			
Description	Key for thisid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)		NOT ALLOWED	a8	Y
Range	thisname IN {arid, orid, wfid .}			

3.2.859. threshcons

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.860. threshfamlen

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.861. threshfammax

Table(s)	PMCC_RECIPE (IDCX)

Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	Y
Range				

3.2.862. threshfammin

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	Y
Range				

3.2.863. threshnsens

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	Y
Range				

3.2.864. time

Table(s)	ARRIVAL (IDCX), ARRIVAL (LEB), ARRIVAL (REB), BEAMAUX (STATIC), DERVDISC (IDCX), FKDISC (LEB), OLD_WFPROTO (IDCX), ORIGIN (IDCX), ORIGIN (LEB), ORIGIN (REB), ORIGIN (SEL1), ORIGIN (SEL2), ORIGIN (SEL3), ORIGIN_TEMP_GA (SEL2), ORIGIN_TEMP_GA (SEL3), QCSTATS (IDCX), SENSOR (STATIC), SITEAUX (STATIC), WFDISC (IDCX), WFDISC (SEGMENT), WFDISC_NOMIG (SEGMENT)			
Description	Epoch time, given as seconds since midnight, January 1, 1970, and stored in a double-precision floating number. time refers to the table in which it is found; for example, in arrival it is the arrival time, in origin it is the origin time, in wfdisc it is the start time of data, and in sensor it is the start time for which measurements are valid. Where the date of historical events is known, time is set to the start time of that date. Where the date of contemporary arrival measurements is known but no time is given, then time is set to the NA Value. Times are given to, at most, one millisecond accuracy. Where the date is unknown, or prior to February 10, 1653, time is set to the NA Value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-9999999999.999	f17.5	N
Range	any valid epoch time			

3.2.865. time

Table(s)	DETECTION (IDCX), DETECTION (LEB), DETECTION (REB), FILE-

	PRODUCT (IDCX), INTERVAL (IDCX), INTERVAL (SEGMENT), OUTAGE (IDCX), PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3), STASSOC (IDCX), TIMESTAMP (IDCX)			
Description	Epoch time, given as seconds since midnight, January 1, 1970, and stored in a double-precision floating number. time refers to the table in which it is found; for example, in arrival it is the arrival time, in origin it is the origin time, in wfdisc it is the start time of data, and in sensor it is the start time for which measurements are valid. Where the date of historical events is known, time is set to the start time of that date. Where the date of contemporary arrival measurements is known but no time is given, then time is set to the NA Value. Times are given to, at most, one millisecond accuracy. Where the date is unknown, or prior to February 10, 1653, time is set to the NA Value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.866. time

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	Epoch time, given as seconds since midnight, January 1, 1970, and stored in a double-precision floating number. time refers to the table in which it is found; for example, in arrival it is the arrival time, in origin it is the origin time, in wfdisc it is the start time of data, and in sensor it is the start time for which measurements are valid. Where the date of historical events is known, time is set to the start time of that date. Where the date of contemporary arrival measurements is known but no time is given, then time is set to the NA Value. Times are given to, at most, one millisecond accuracy. Where the date is unknown, or prior to February 10, 1653, time is set to the NA Value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds		f17.5	Y
Range	any valid epoch time			

3.2.867. time

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)				Y
Range				

3.2.868. time

Table(s)	PROBLEM (IDCX), PROBLEMLOG (IDCX)
Description	Epoch time, given as seconds since midnight, January 1, 1970, and stored in a double-precision floating number. time refers to the table in which it is found; for example, in arrival it is the arrival time, in origin it is the origin time, in wf-

	disc it is the start time of data, and in sensor it is the start time for which measurements are valid. Where the date of historical events is known, time is set to the start time of that date. Where the date of contemporary arrival measurements is known but no time is given, then time is set to the NA Value. Times are given to, at most, one millisecond accuracy. Where the date is unknown, or prior to February 10, 1653, time is set to the NA Value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(17,5)	seconds	-9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.869. time_rely

Table(s)	STA_RELY (STATIC)			
Description	Time reliability flag. The value is a single-character flag to indicate whether (y) or not (n) the time is considered to be reliable.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	Y
Range	time_rely IN { 0,1}			

3.2.870. time_spread

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Root-mean-square time spread of the estimated signal energy between onset_time and termination_time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f9.4	Y
Range	timespread > 0.0			

3.2.871. time_tol

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.872. time_weight

Table(s)	PHASE_WEIGHT_EVENT_DEFINE (STATIC)			
Description	Weight given to defining time for this phase			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	N
Range				

3.2.873. timedef

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Time-defining code. This one-character flag indicates whether or not the time of a phase was used to constrain the event location. This column is defining (timedef = d) or non defining (timedef = n).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)		- (hyphen)	a1	Y
Range	timedef IN {n, d}			

3.2.874. timelastsend

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Last epoch time that the product was delivered by the Subscription system.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(54)	seconds	-9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.875. timenextsend

Table(s)	PRODUCTCRITERIA (IDCX)			
Description	Next epoch time that the product will be delivered by the Subscription system.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(54)	seconds	-9999999999.999	f17.5	Y
Range	any valid epoch time			

3.2.876. timeres

Table(s)	ALLOW_RESID (LEB), ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Time residual. This column is a travel-time residual measured in seconds. The residual is found by taking the observed arrival time (saved in the ARRIVAL table) of a seismic phase and subtracting the expected arrival time. The expected arrival time is calculated by a formula based on an earth velocity model (column vmodel), an event location and origin time (saved in table ORIGIN (Origin_temp_ga)), the distance to the station (column dist in table ASSOC), and the particular seismic phase (column phase in table ASSOC).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f8.3	Y
Range	timeres > -999.0			

3.2.877. timres

Table(s)	DEFINE_PHASE_RESIDUAL (STATIC)			
Description	Time residual (seconds)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	NOT ALLOWED	f8.3	N
Range	timres >= 0.0			

3.2.878. tlen

Table(s)	DERVDISC (IDCX)			
Description	Duration of the data interval in seconds.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds		f8.3	Y
Range	tlen > 0			

3.2.879. tlen

Table(s)	AMPDESCRIPT (IDCX), FKDISC (LEB)			
Description	Time window length. Tlen should be NA in ampdescript if a velocity window is used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f10.3	Y
Range	tlen > 0.0			

3.2.880. tmf

Table(s)	THIRDMOM (REB)			
Description	Third moment of frequency.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	tmf >= 0.0			

3.2.881. tmfpct

Table(s)	THIRDMOM (REB)			
Description	Percentage of signal spectrum amplitudes greater than their corresponding noise spectrum amplitudes in the frequency range [tmf-fmin, tmf-fmax]. tmf-fmin and tmf-fmax are par-file parameters.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	0 <= tmfpct <= 100			

3.2.882. toff

Table(s)	AMPDESCRIPT (IDCX)			
Description	Offset from theoretical or observed arrival time. This column is used to define the start time of the amplitude measurement window and may be used in conjunction with either tlen to define a static window or with gvlo to define a dynamic window. If toff is set to -999, then gvhi must be used to define the start time of the window.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-999.0	f6.2	Y
Range	toff >= 0.0			

3.2.883. total_energy

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Total energy in the signal between onset_time and termination_time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	dB re mPa	-1.0	f9.4	Y
Range	total_energy > 0.0			

3.2.884. total_energy

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Total energy in the signal between onset_time and termination_time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	dB re µPa	-1.0	f9.4	Y
Range	total_energy > 0.0			

3.2.885. total_time

Table(s)	HYDRO_FEATURES (IDCX), HYDRO_FEATURES (LEB), HYDRO_FEATURES (REB)			
Description	Total time that the estimated signal pressure squared exceeds noise_onset_thresh between onset_time and termination_time.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	seconds	-1.0	f17.5	Y
Range	total_time > 0.0			

3.2.886. transmeth

Table(s)	MSGDEST (IDCX)
Description	Method by which the response is to be delivered to the requestor.

Storage type	Unit	NA value	External	Nullable
varchar2(16)		- (hyphen)	a16	Y
Range	any string up to 16 characters			

3.2.887. travel_time_error

Table(s)	PHASOR_DIST_DEPTH_RANGES (STATIC)			
Description	Travel time error in seconds, used for building error window for Phasor processing.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	-1.0	f8.3	Y
Range	travel_time_error >= 0.0			

3.2.888. tshift

Table(s)	SENSOR (STATIC)			
Description	Correction for clock errors. This column is designed to accommodate discrepancies between actual time and the numerical time written by data recording systems. Actual time is the sum of the reported time plus tshift.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	seconds	NOT ALLOWED	f6.2	Y
Range	any valid real number			

3.2.889. tweight

Table(s)	WEIGHTS (STATIC)			
Description	Time weight.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		0.0	f5.2	Y
Range	tweight >= 0.0			

3.2.890. typeid

Table(s)	FPDESCRIPTION (IDCX)			
Description	Identifier for the product type description.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i10	N
Range	typeid > 0			

3.2.891. typeid

Table(s)	FILEPRODUCT (IDCX)
Description	Identifier for the product type description.

Storage type	Unit	NA value	External	Nullable
number(10,0)		NOT ALLOWED	i10	Y
Range	typeid > 0			

3.2.892. typeid

Table(s)	FS_STAGEPRODUCT (IDCX)			
Description	obsolete			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	Y
Range				

3.2.893. typeid

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)				Y
Range				

3.2.894. uncertainty

Table(s)	NETMAG (LEB), NETMAG (REB), NETMAG (SEL1), NETMAG (SEL2), NETMAG (SEL3), STAMAG (LEB), STAMAG (REB), STAMAG (SEL1), STAMAG (SEL2), STAMAG (SEL3)			
Description	Magnitude uncertainty. This column value is the standard deviation of the accompanying magnitude measurement.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f7.2	Y
Range	uncertainty > 0.0			

3.2.895. units

Table(s)	AMPLITUDE (IDCX), AMPLITUDE (LEB), AMPLITUDE (REB)			
Description	Units of amplitude measure.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	units IN {nm}			

3.2.896. userid

Table(s)	SUBS (IDCX)
Description	User identifier for Subscription and Message Subsystems.

Storage type	Unit	NA value	External	Nullable
number(8,0)		-1	i8	N
Range	userid > 0			

3.2.897. userid

Table(s)	DATAUSER (IDCX), SUBSUSER (IDCX)			
Description	User identifier for Subscription and Message Subsystems.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	userid > 0			

3.2.898. userid

Table(s)	MSGDISC (IDCX)			
Description	User identifier for Subscription and Message Subsystems.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		-1	i8	Y
Range	userid > 0			

3.2.899. username

Table(s)	FTPLOGIN (IDCX)			
Description	UNIX login name or email address for FTP logins.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)		- (hyphen)	a16	Y
Range	any string up to the character field width that is a valid UNIX user name			

3.2.900. username

Table(s)	DATAUSER (IDCX)			
Description	User name component of the email address.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)		NOT ALLOWED	a24	N
Range	any string up to the character field width that is a valid UNIX user name			

3.2.901. username

Table(s)	SUBSUSER (IDCX)			
Description	(Datauser/Subsuser): user name component of the email address. (Ftplogin): UNIX login name or email address for FTP logins.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(24)		NOT ALLOWED	a24	Y
Range	any string up to the character field width that is a valid UNIX user name			

3.2.902. vamp

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB)			
Description	Vertical amplitude.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f11.2	Y
Range	vamp >= 0.0			

3.2.903. vang

Table(s)	SITECHAN (STATIC)			
Description	Vertical orientation of seismometer. This column measures the angle between the sensitive axis of a seismometer and the outward-pointing vertical direction. For a vertically oriented seismometer, vang = 0. For a horizontally oriented seismometer, vang = 90 (see hang).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees	NOT ALLOWED	f6.1	Y
Range	0.0 <= vang <= 90.0			

3.2.904. verifstatus

Table(s)	MSGDISC (IDCX)			
Description	Authentication status of the message, either PASS, FAIL or (hyphen) for not tested/applicable.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4)		- (hyphen)	a4	Y
Range	ASCII character string			

3.2.905. version

Table(s)	FILEPRODUCT (IDCX)			
Description	Version of the software that generated the file product.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)		-1.0	f6.2	Y
Range	version > 0.0			

3.2.906. version

Table(s)	FS_STAGEPRODUCT (IDCX)
Description	obsolete

Storage type	Unit	NA value	External	Nullable
float(53)			f17.5	Y
Range				

3.2.907. version

Table(s)	FILEPRODUCT_BU_20200610 (IDCX), FILEPRODUCT_TEMP (IDCX)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)				Y
Range				

3.2.908. vmodel

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3), PARRIVAL (LEB), PARRIVAL (REB), PARRIVAL (SEL1), PARRIVAL (SEL2), PARRIVAL (SEL3)			
Description	Velocity model. This character string identifies the velocity model of the earth used to compute the travel times of seismic phases. A velocity model is required for event location (if phase is defining) or for computing travel-time residuals.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)		- (hyphen)	a15	Y
Range	any string up to 15 characters			

3.2.909. vsnr

Table(s)	AMP3C (IDCX), AMP3C (LEB), AMP3C (REB)			
Description	Vertical signal-to-noise ratio. Ratio of vamp to root-mean-square amplitude of a vertically-oriented component filtered in a frequency band centred at cfreq Hz.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f10.2	Y
Range	vsnr >= 0.0			

3.2.910. wfid

Table(s)	OLD_WFPROTO (IDCX)			
Description	Unique waveform identifier (integer)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		NOT ALLOWED	i8	Y
Range	wfid > 0			

3.2.911. wfid

Table(s)	WFDISC (IDCX), WFDISC (SEGMENT)			
Description	Unique waveform identifier (integer)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		NOT ALLOWED	i8	N
Range	wfid > 0			

3.2.912. wfid

Table(s)	WFAUX (IDCX), WFTAG (IDCX), WFTAG (SEGMENT), WFTAG (SEL1), WFTAG (SEL2), WFTAG (SEL3)			
Description	Unique waveform identifier for a wfdisc record.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)		-1	i10	N
Range	wfid > 0			

3.2.913. wfid

Table(s)	WFDISC_NOMIG (SEGMENT)			
Description	Unique waveform identifier (integer)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)		NOT ALLOWED	i8	N
Range	wfid > 0			

3.2.914. wgt

Table(s)	ASSOC (IDCX), ASSOC (LEB), ASSOC (REB), ASSOC (SEL1), ASSOC (SEL2), ASSOC (SEL3), ASSOC_TEMP_GA (SEL2), ASSOC_TEMP_GA (SEL3)			
Description	Location weight. This column gives the final weight assigned to the allied arrival by the location program. This column is used primarily for location programs that adaptively weight data by their residuals.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-1.0	f6.3	Y
Range	0.0 <= wgt < 1.0			

3.2.915. wingap

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y

Range	

3.2.916. winlen

Table(s)	PMCC_RECIPE (IDCX)			
Description	Not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)			f8.3	Y
Range				

3.2.917. xavpct

Table(s)	TIMEFREQ (REB)			
Description	Average ratio of bad points to total of the cross-correlation trace.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	0 <= xavpct <= 1			

3.2.918. xcoef

Table(s)	ATTENCOEF (STATIC)			
Description	Constant coefficient in the distance correction formula.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	any valid floating point value			

3.2.919. xcor

Table(s)	TIMEFREQ (REB)			
Description	For timefreq, xcor is the zero lag cross-correlation between the three pairs of time-frequency matrices.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f9.4	Y
Range	0 <= xcor <= 1			

3.2.920. ycoef

Table(s)	ATTENCOEF (STATIC)			
Description	Geometrical spreading coefficient in the distance correction formula.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	any valid floating point value			

3.2.921. zavcep

Table(s)	TIMEFREQ (REB)			
Description	Average maximum value in the two-dimensional cepstrum of the vertical component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	zavcep >= 0			

3.2.922. zavcor

Table(s)	TIMEFREQ (REB)			
Description	Average autocorrelation along the time axis across all frequencies excluding randomized points of the vertical component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	zavcor >= 0			

3.2.923. zavpct

Table(s)	TIMEFREQ (REB)			
Description	Average ratio of bad points to total of the vertical component traces.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		NOT ALLOWED	f7.2	Y
Range	0 <= zavpct <= 1			

3.2.924. zcoef

Table(s)	ATTENCOEF (STATIC)			
Description	Attenuation coefficient in the distance correction formula.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)		-999.0	f7.2	Y
Range	any valid floating point value			

3.2.925. zrcr_delfreq

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	The estimated standard deviation in zrcr_freq.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	–1.0	f7.2	Y
Range	zcr_delfreq > 0.0			

3.2.926. zrcr_freq

Table(s)	INFRA_FEATURES (IDCX), INFRA_FEATURES (LEB), INFRA_FEATURES (REB)			
Description	Frequency of the arrival as estimated from the zero-crossing rate of the traditional beam-formed waveform data in the time interval defined by coinc_time and coinc_dur.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(24)	Hertz	–1.0	f7.2	Y
Range	zcr_freq > 0.0			

3.3. S/H/I Trigger Descriptions

3.3.1. DATAREADY_TO_PRODTRACK

Table	DATAREADY
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	DECLARE
	
	-- site_time: EST for CMR, GMT at the Vienna IDC
	  site_time VARCHAR(4) := 'GMT';
	
	-- select_string: Information always selected into table prodtrack
	-- at CMR, will resolve to: SELECT distinct pc.prodid, -1, :new.dataid,
	-- -1, 'NEW', NEW_TIME(SYSDATE,'EST','GMT')
	  select_string VARCHAR(1000) := 'defaultvalue';
	
	-- generic strings are modified in mkstring_generic.sql
	-- note that these are not fully formed sql statements!
	
	-- Note by CT, 16 July 2003:
	-- This trigger requires the GLOBAL TEMPORARY table
	-- idcx.arrival_sta_trigger.
	-- Dropping this table will invalidate the trigger.
	
	  GET_ARRIVAL_DAY_RANGE_string VARCHAR(10000) := 'default-value';
	  GET_ARRIVAL_DAY_STA_string VARCHAR(10000) := 'default-value';
	  GET_ARRIVAL_INST_RANGE_string VARCHAR(10000) := 'default-value';
	  GET_ARRIVAL_INST_STA_string VARCHAR(10000) := 'default-value';
	  GET_ARRIVAL_TIME_STA_string VARCHAR(10000) := 'default-value';
	  GET_ARRIVAL_TIME_RANGE_string VARCHAR(10000) := 'default-value';
	  JD_TO_TIME_RANGE_string VARCHAR(10000) := 'default-value';
	  ORID_JOIN_string VARCHAR(10000) := 'defaultvalue';
	  ORIG_LIMITS_string VARCHAR(10000) := 'default-value';
	  TIME_JOIN_CYCLE_string VARCHAR(10000) := 'default-value';
	  TIME_JOIN_DAILY_string VARCHAR(10000) := 'default-value';
	  TIME_JOIN_INSTANT_string VARCHAR(10000) := 'default-

```
value';
      DETECT_DAY_string       VARCHAR(10000) := 'defaultvalue';
      DETECTION_TIME_string         VARCHAR(10000) := 'default-
value';
      PC_DAILY_string               VARCHAR(10000) := 'default-
value';
      PC_INSTANT_string       VARCHAR(10000) := 'defaultvalue';
      PC_TIME_string                VARCHAR(10000) := 'default-
value';
      STA_JOIN_DET_string           VARCHAR(10000) := 'default-
value';
      STA_REG_JOIN_DET_string       VARCHAR(10000) := 'default-
value';
      NO_SURPRESS_EMPTY_string      VARCHAR(10000) := 'default-
value';
      UPDATE_TIME_CYCLE_string      VARCHAR(10000) := 'default-
value';
      UPDATE_TIME_INSTANT_string    VARCHAR(10000) := 'default-
value';

--    these hold new values in row, used to construct query
strings

      new_dataid  NUMBER(10) := 0;
      new_tablename     VARCHAR2(24) := 'defaultvalue';
      new_account VARCHAR2(24) := 'defaultvalue';
      new_tagname VARCHAR2(12) := 'defaultvalue';
      new_tagid   NUMBER(10) := 0;
      new_tagid2  NUMBER(10) := 0;
      new_data_info     VARCHAR2(24) := 'defaultvalue';

--    These strings hold the final sql queries, and are modi-
fied in
--    the files mkstring_origin.sql and mkstring_other.sql

      origin_jdate_string_1         VARCHAR(30000) := 'default-
value';
      origin_jdate_string_2         VARCHAR(30000) := 'default-
value';
      origin_orid_string_1          VARCHAR(30000) := 'default-
value';
      origin_time_inst_string_1     VARCHAR(30000) := 'default-
value';
      origin_time_inst_string_2     VARCHAR(30000) := 'default-
value';
      origin_time_string_1          VARCHAR(30000) := 'default-
value';
      origin_time_string_2          VARCHAR(30000) := 'default-
value';
      arrival_jdate_string          VARCHAR(30000) := 'default-
value';
      arrival_arid_string           VARCHAR(30000) := 'default-
value';
```

```
      arrival_time_string_1         VARCHAR(30000) := 'default-
value';
      arrival_time_string_2         VARCHAR(30000) := 'default-
value';
      arrival_time_inst_string_1    VARCHAR(30000) := 'default-
value';
      arrival_time_inst_string_2    VARCHAR(30000) := 'default-
value';
      wfdisc_wfid_string            VARCHAR(30000) := 'default-
value';
      wftag_jdate_string            VARCHAR(30000) := 'default-
value';
      wftag_orid_string       VARCHAR(30000) := 'defaultvalue';
      fileproduct_string            VARCHAR(30000) := 'default-
value';
      evchar_string                 VARCHAR(30000) := 'default-
value';

-- Lines added by CT on 16-JUL-2003
-- Start modification
        delete_temp_table               VARCHAR(30000) := 'de-
faultvalue';
        insert_day_string               VARCHAR(30000) := 'de-
faultvalue';
        insert_time_string              VARCHAR(30000) := 'de-
faultvalue';
-- End modification


BEGIN
      new_dataid  := :new.dataid;
      new_tablename     := :new.tablename;
      new_account := :new.account;
      new_tagname := :new.tagname;
      new_tagid   := :new.tagid;
      new_tagid2  := :new.tagid2;
      new_data_info     := :new.data_info;

--    modify select_string
      select_string:= 'SELECT distinct pc.prodid,...
```

3.3.2. FILEPRODUCT_NULL_LDDATE

Table	FILEPRODUCT
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`BEGIN`
	`  :new.lddate := sysdate;`
	`END;`

3.3.3. FILEPRODUCT_TO_DATAREADY

Table	FILEPRODUCT
Type	AFTER EACH ROW
Event	INSERT

Trigger body	DECLARE
	  Id NUMBER(10);
	BEGIN
	  SELECT keyvalue into Id from lastid
	  where keyname = 'dataid'
	  for update of lastid.keyvalue;
	  INSERT into dataready
	  SELECT
	  lastid.keyvalue+1,
	  'fileproduct',
	  f.prodtype,
	  'fpid',
	  :new.fpid,
	  -1,
	  :new.sta,
	  'n',
	  :new.lddate
	  FROM
	  FPDescription f, lastid lastid
	  WHERE
	  f.typeid=:new.typeid
	  AND
	  lastid.keyname = 'dataid';
	  UPDATE lastid set keyvalue = keyvalue + 1
	  where keyname = 'dataid';
	END;

3.3.4. LASTID_CHECKS

Table	LASTID
Type	BEFORE EACH ROW
Event	UPDATE

Trigger body	DECLARE msg VARCHAR2(4000);
	  BEGIN
	  :new.LDDATE := SYSDATE;
	  IF :new.keyvalue<=:old.keyvalue THEN
	  msg :='New keyvalue '||:new.keyvalue||' for '||:new.keyname||' is less than previous one ' || :old.key-value;
	  raise_application_error(-20003, msg);
	  END IF;
	  IF :new.keyname<>:old.keyname THEN
	  msg :='Keyname value ?|:old.keyvalue|? could not be changed to '||:new.keyname;
	  raise_application_error(-20004, msg);

	END IF; END;

3.3.5. WFDISC_ENDTIME

Table	WFDISC
Type	AFTER EACH ROW
Event	INSERT OR UPDATE

Trigger body	DECLARE
	PRAGMA AUTONOMOUS_TRANSACTION;
	Station VARCHAR2(16);
	MaxTime FLOAT(53);
	BEGIN
	--- the internal commits require the PRAGMA AUTONOMOUS_TRANSACTION
	BEGIN -- first block
	SELECT net into Station FROM "STANET" WHERE sta = :new.sta;
	EXCEPTION
	WHEN NO_DATA_FOUND THEN
	raise_application_error(-20001, 'Station '||:new.sta||' not in stanet!');
	END; --first block
	BEGIN -- second block, other no data exception
	SELECT time into MaxTime FROM "TIMESTAMP"
	WHERE procclass = 'DLTRIGGER' and procname = Station;
	IF :new.endtime > MaxTime THEN
	UPDATE "TIMESTAMP"
	SET time = :new.endtime
	WHERE procclass = 'DLTRIGGER' and procname = Station;
	COMMIT; -- *MUST* commit autonomous transaction here
	END IF;
	EXCEPTION
	WHEN NO_DATA_FOUND THEN
	INSERT INTO "TIMESTAMP"
	VALUES ('DLTRIGGER', Station, :new.endtime, sysdate);
	COMMIT; -- *MUST* commit autonomous transaction here too
	END; -- second block
	-- Do *NOT* catch the "OTHERS" exception here, or the exception raised by
	-- raise_application_error() is caught here and the trigger succeeds
	-- (the transaction in not rolled back), which is definitely not what
	-- you want!!
	END;

3.3.6. WFDISC_NVIAR_ENDTIME

Table	WFDISC
Type	AFTER EACH ROW
Event	INSERT OR UPDATE

Trigger body	DECLARE
	Station VARCHAR2(16);
	MaxTime FLOAT(53);
	BEGIN
	SELECT time into MaxTime FROM timestamp
	WHERE procclass = 'DLTRIGGER' and procname = 'NVIAR';
	IF :new.endtime > MaxTime THEN
	UPDATE timestamp
	SET time = :new.endtime
	WHERE procclass = 'DLTRIGGER' and procname = 'NVIAR';
	END IF;
	END;

3.4. S/H/I Index Descriptions

Table name	Index name	Unique	Index columns
AFFILIATION	AFFNETXX	No	net
	AFFSTAX	Yes	sta, net
	PK_AFFILIATION	Yes	net, sta
AFFILIATION_SUB	PK_AFFILIATION_SUB	Yes	offdate, net, sta
ALLOCATE_HOUR	ALLOCATE_HOURX	Yes	jdate, hour, action
	PK_ALLOCATE_HOUR	Yes	jdate, hour, action
ALLOW_RESID	ALLOWPHASEX	No	phase, statype
	PK_ALLOW_RESID	Yes	phase, mindelta, sta-type
AMP3C	AMP3CARIDX	No	arid
	AMP3C_LDDATEX	No	lddate
	PK_AMP3C	Yes	arid, rid
AMPDESCRIPT	PK_AMPDESCRIPT	Yes	amptype
AMPLITUDE	AMPARIDX	No	arid
	AMPLDDATEX	No	lddate
	AMPLITUDE_LDDATEX	No	lddate
	AMPPARIDX	No	parid
	AMP_PARID_ARID_X	No	parid, arid
	PK_AMPLITUDE	Yes	ampid
APMA	APMA_LDDATEX	No	lddate
	PK_APMA	No	arid
ARCH_DATA_TYPE	PK_ARCH_DATA_TYPE	Yes	archid
ARRIVAL	ARRIVAL_ARID_COMMID_X	Yes	arid, commid
	ARRIVAL_LDDATEX	No	lddate
	ARSTASSX	No	stassid
	ARSTTIMEX	No	sta, time
	ARTIMEX	No	time
	PK_ARRIVAL	Yes	arid
ASSOC	ASSARIDX	No	arid, arid
	ASSOCX	Yes	orid, arid
	ASSOC_LDDATEX	No	lddate
ASSOC_TEMP_GA	ASSOC_TEMP_GAX	Yes	arid, orid
	ATGORIDX	No	orid
ATTENCOEF	PK_ATTENCOEF	Yes	attenid, sta, ratiotype, chan

Table name	Index name	Unique	Index columns
CEPPKS	CEPPKS_LDDATEX	No	lddate
	PK_CEPPKS	Yes	orid, sta, pkqf
CHANNAME	PK_CHANNAME	Yes	stream, extern_sta, extern_chan, revision
CHAN_GROUPS	PK_CHAN_GROUPS	Yes	class, name, sta, chan, ondate, offdate
COLORDISC	COLORDISC_PK	Yes	colormapid
COMPLEXITY	COMPLEXITY_LDDATEX	No	lddate
	COMPLORIDX	No	orid
	PK_COMPLEXITY	Yes	orid, sta, phase
DATAREADY	DATAREADY_LDDATEX	No	lddate
	DRTAGIDX	No	tagid
	PK_DATAREADY	Yes	dataid
DATAUSER	PK_DATAUSER	Yes	userid
DERVDISC	DERVDISC_PK	Yes	dervid
	DERV_STAX	No	sta, time
DETECTION	DETLDDATEX	No	lddate
	DETTIMEX	No	time
	PK_DETECTION	Yes	arid
DISCARD	DISCARDEVIDX	No	evid
	DISCARD_LDDATEX	No	lddate
	PK_DISCARD	Yes	evid, reason, lddate
DLMAN	DLMANX	Yes	dlid
DSEISGRID	PK_DSEISGRID	Yes	grdname, icell
DSEISINDEX	PK_DSEISINDEX	Yes	grdname
EVCHAR_PROD	EVCHAR_PROD_LDDATEX	No	lddate
	PK_EVCHAR_PROD	Yes	prodid, orid
EVENT	EVENT_LDDATEX	No	lddate
	PK_EVENT	Yes	evid
EVENT_CONTROL	EVNTCTRLEVIDX	No	evid
	PK_EVENT_CONTROL	Yes	orid
EVSC_HYDRO	PK_EVSC_HYDRO	No	orid, sta
EVSC_PROD	EVSC_PROD_ORIDX	No	orid
	PK_EVSC_PROD	Yes	prodid, orid
EVSC_REGIONAL	PK_EVSC_REGIONAL	No	orid, sta, chan
FDSNCHAN	FDSNCHAN_PK	Yes	mapid
FILEPRODUCT	FILEPROD_CHANX	No	chan

Table name	Index name	Unique	Index columns
	FILEPROD_DFILEX	No	dfile
	FILEPROD_ENDTIMEX	No	endtime
	FILEPROD_LDDATEX	No	lddate
	FILEPROD_TID	No	typeid, sta
	FILEPROD_TIMEX	No	time
	PK_FILEPRODUCT	Yes	fpid
FORBEAMAUX	PK_FORBEAMAUX	Yes	sta, chan
FPDESCRIPTION	PK_FPDESCRIPTION	Yes	typeid
FTPFAILED	PK_FTPFAILED	No	msgid
FTPLOGIN	FTPLOGIN_PK	Yes	ftp_address
GA_TAG	GAIDX	No	id, state
GREGION	GREGGRNAMEX	No	grname
	PK_GREGION	Yes	grn
HYDRO_ARR_GROUP	PK_HYDRO_ARR_GROUP	Yes	hydro_id
HYDRO_ASSOC	HYDRO_ASSOC_ARIDX	No	arid, hydro_id
	HYDRO_ASSOC_HIDX	No	hydro_id, arid
HYDRO_FEATURES	HYDARIDX	No	arid
	HYDRO_FEATURES_LDDA-TEX	No	lddate
	PK_HYDRO_FEATURES	Yes	arid, high_cut, low_cut
INFRA_FEATURES	INFRAARIDX	No	arid
	INFRA_FEATURES_LDDATEX	No	lddate
	PK_INFRA_FEATURES	No	arid
INSTRUMENT	PK_INSTRUMENT	Yes	inid
INTERVAL	INCLANASTAMO	No	class, name, state, moddate
	INCLASSX	No	class, name, endtime
	INENDX	No	endtime
	INMODX	No	moddate
	PK_INTERVAL	No	intvlid
IN_EVENT_CONTROL	IN_EVENT_CONTROLX	Yes	orid
LASTID	LASTIDX	Yes	keyname
MAPCOLOR	MAPCOLOR_PK	Yes	mapid, colormapid
MAPDISC	MAPDISC_PK	Yes	mapid
MAPOVER	MAPOVER_PK	Yes	mapid, overlayid
MAPPOINT	MAPPOINT_PK	Yes	lat, lon, mptype
MSGAUX	MGSXMSGIDX	No	msgid

Table name	Index name	Unique	Index columns
	MSGAUX_LDDATEX	No	lddate
	PK_MSGAUX	Yes	msgid, msgrow, state_count
MSGDATATYPE	MSGDATATYPE_LDDATEX	No	lddate
	MSGTYPMSGIDX	No	msgid
	PK_MSGDATATYPE	Yes	msgid, foff
MSGDEST	MSGDEST_LDDATEX	No	lddate
	MSGDSTMSGIDZ	No	msgid
	MSGDSTSTATX	No	status
	PK_MSGDEST	Yes	msgdid
MSGDISC	MSGDISC_ITIMEX	No	itime
	MSGDISC_LDDATEX	No	lddate
	MSGIDATEX	No	idate
	MSGINTIDX	No	intid
	MSGSTATX	No	status
	PK_MSGDISC	Yes	msgid
NA_VALUE	PK_NA_VALUE	Yes	attribute
NETMAG	NETMAG_LDDATEX	No	lddate
	NMORIDX	No	orid
	PK_NETMAG	Yes	magid
NETWORK	PK_NETWORK	Yes	net
ORIGERR	ORIGERR_LDDATEX	No	lddate
	PK_ORIGERR	Yes	orid
ORIGERR_TEMP_GA	ORIGERR_TEMP_GAX	Yes	orid
ORIGIN	ANLOG_JDATEX	No	jdate
	EVIDX	Yes	evid
	ORIGIN_LDDATEX	No	lddate
	ORTIMEX	No	time
	PK_ORIGIN	Yes	orid
ORIGIN_TEMP_GA	ORIGIN_TEMP_GAX	Yes	orid
	OTGTIMEX	No	time
OUTAGE	OTGSTAX	No	sta, chan, time
	OTGTIMEX	No	time
	PK_OUTAGE	Yes	otgid
OVERLAYDISC	OVERLAYDISC_PK	Yes	overlayid
PARRIVAL	PAREVIDX	No	evid
	PARREVIDX	No	evid

Table name	Index name	Unique	Index columns
	PARRIVALX	Yes	parid
	PARRIVAL_LDDATEX	No	lddate
	PARRORIDX	No	orid
	PK_PARRIVAL	Yes	parid
PARTICIPATION	PK_PARTICIPATION	Yes	net, sta, arid, sta, begin_date
PMCC_FEATURES	PK_PARTICIPATION	Yes	net, sta, arid, sta, begin_date
	PMCC_FEAT_ARIX	No	arid
PMCC_RECIPE	PMCC_RECIPE_PK	Yes	pmccrecid
PROBLEM	PK_PROBLEM	Yes	prid
PROBLEMLOG	PK_PROBLEMLOG	Yes	plid
PROBLEMMAIL	PK_PROBLEMMAIL	Yes	procclass, login
PRODTRACK	INMSGIDX	No	msgid
	INPRODIDX	No	prodid
	INSTATUSX	No	status
	PRODTRACK_LDDATEX	No	lddate
	PRODTRACK_PK	Yes	prodid, dataid, msgid
PRODUCTCRITERIA	PRODCRITX	Yes	prodid
PRODUCTTYPEEVPARS	PK_PRODUCTTYPEEVPARS	No	prodid
PRODUCTTYPEEVSC	PK_PRODUCTTYPEEVSC	Yes	prodid
PRODUCTTYPEORIGIN	PK_PRODUCTTYPEORIGIN	Yes	prodid
PRODUCTTYPESTA	PK_PRODUCTTYPESTA	Yes	prodid, sta, chan
QCCHANINT	QCCHANINT_PK	Yes	qcchanintid
	QCCHANINT_QCNETINTID_IDX	No	qcnetintid
QCDEFPARAM	QCDEFPARAM_PK	Yes	qcdefparamid
QCDEFSET	QCDEFSET_PK	Yes	qcdefsetid
QCMASK	QCMASK_PK	Yes	qcmaskid
	QCMASK_QCCHANINTID_IDX	No	qcchanintid
QCNETINT	QCENDTIMEX	No	endtime
	QCNETINT_AN_T_ET_IDX	No	aff_net, time, endtime
	QCNETINT_PK	Yes	qcnetintid
QCRULEPARAM	QCRULEPARAM_PK	Yes	qcruleparamid
QCRULESET	QCRULESET_PK	Yes	qcrulesetid
QCSTATS	QCSTATSIDX	No	qcstatsid
	QCSTATS_ENDSTCHNX	No	endtime, sta, chan

Table name	Index name	Unique	Index columns
	QCSTATS_LDDATEX	No	lddate
	QCSTATS_TIMEX	No	time
REGCOEF	PK_REGCOEF	Yes	rcoefid, rcoeftype
REMARK	PK_REMARK	Yes	commid, lineno
	REMARKX	Yes	commid, lineno
	REMARK_LDDATEX	No	lddate
REQUEST	PK_REQUEST	Yes	reqid
	REQENDX	No	end_time
	REQMODX	No	modtime
	REQSTATEX	No	state
	REQSTENDX	No	start_time, end_time
	REQUEST_LDDATEX	No	lddate
REVAUDIT	PK_REVAUDIT	No	revid
	REVTAGNDX	No	revtagid1, revtagname1
RULESET	RULESETSN	Yes	sta, offdate, name
	RULESET_PK	Yes	sta, ondate, regexp
SEISGRID	PK_SEISGRID	Yes	grdname, icell
	SEISGRIDX	Yes	icell
SEISINDEX	PK_SEISINDEX	Yes	grdname
SENSOR	PK_SENSOR	Yes	sta, chan, time, end-time
	SENSSTAX	No	sta, chan
SITE	PK_SITE	Yes	sta, ondate
	SITESTAX	No	sta
SITEAUX	PK_SITEAUX	Yes	sta, chan, time
SITECHAN	PK_SITECHAN	Yes	sta, chan, ondate
	SITECHANX	Yes	chanid
	STCSTAX	No	sta, chan
SITEPOLL	PK_SITEPOLL	Yes	sta, net, chan
SITE_ADDRESS	PK_SITE_ADDRESS	Yes	sta
SPLP	PK_SPLP	Yes	orid, sta
	SPLPORIDX	No	orid
	SPLP_LDDATEX	No	lddate
SPVAR	PK_SPVAR	Yes	arid, fmin, fmax
	SPVAR_LDDATEX	No	lddate
SREGION	PK_SREGION	Yes	srn
STAMAG	PK_STAMAG	Yes	magid, ampid, sta

Table name	Index name	Unique	Index columns
	SMAMPIDX	No	ampid
	SMARIDX	No	arid
	SMMAGIDX	No	magid, sta
	SMORIDX	No	orid
	SM_ARID_COMMID_X	No	arid, commid
	STAMAG_LDDATEX	No	lddate
STANET	PK_STANET	Yes	net, sta
	STANETSTAX	Yes	sta
STASSOC	PK_STASSOC	No	stassid
	STASSOC_LDDATEX	No	lddate
	STASSTIMEX	No	time
SUBS	PK_SUBS	Yes	userid, prodid
SUBSUSER	PK_SUBSUSER	Yes	userid
THIRDMOM	PK_THIRDMOM	Yes	arid
TIMEFREQ	PK_TIMEFREQ	Yes	orid, sta
TIMESTAMP	TIMESTAMP_PK	Yes	procclass, procname
WEIGHTS	PK_WEIGHTS	Yes	phase, statype, net
WFAUX	WFAUXX	Yes	wfid
WFCONV	PK_WFCONV	Yes	sta, chan
WFDISC	PK_WFDISC	Yes	wfid
	WFDIRX	No	dir, dfile
	WFDISC_LDDATEX	No	lddate
	WFSTAX	No	sta, time, endtime, chan
	WFTIMEX	No	time
	WF_TIME_ENDTIME_IDX	No	time, endtime
WFTAG	PK_WFTAG	Yes	tagname, tagid, wfid
	WFTTAGIDX	No	tagid
	WFTWFIDX	No	wfid
XTAG	XTAGTHATIDNDX	Yes	thatid

3.5. S/H/I Synonym Descriptions

Schema	Synonym	Table	Table Owner
IDCX	AFFILIATION	AFFILIATION	STATIC
	ATTENCOEF	ATTENCOEF	STATIC
	CHANNAME	CHANNAME	STATIC
	CHAN_GROUPS	CHAN_GROUPS	STATIC
	CHAN_GROUPS_AUXWF	CHAN_GROUPS_AUXWF	STATIC
	FORBEAMAUX	FORBEAMAUX	STATIC
	INSTRUMENT	INSTRUMENT	STATIC
	NETWORK	NETWORK	STATIC
	PARTICIPATION	PARTICIPATION	STATIC
	REGCOEF	REGCOEF	STATIC
	RULESET	RULESET	STATIC
	SENSOR	SENSOR	STATIC
	SITE	SITE	STATIC
	SITEAUX	SITEAUX	STATIC
	SITECHAN	SITECHAN	STATIC
	STANET	STANET	STATIC
LEB	AFFILIATION	AFFILIATION	STATIC
	AMPDESCRIPT	AMPDESCRIPT	IDCX
	BEAMAUX	BEAMAUX	STATIC
	BREGION_DEF	BREGION_DEF	STATIC
	BREGION_TAB	BREGION_TAB	STATIC
	DERVDISC	DERVDISC	IDCX
	DSEISGRID	DSEISGRID	STATIC
	DSEISINDEX	DSEISINDEX	STATIC
	GREGION	GREGION	STATIC
	IADR_WFTAG	WFTAG	IDCX
	INSTRUMENT	INSTRUMENT	STATIC
	IN_ASSOC	ASSOC	SEL3
	IN_HYDRO_ARR_GROUP	HYDRO_ARR_GROUP	IDCX
	IN_HYDRO_ASSOC	HYDRO_ASSOC	IDCX
	IN_NETMAG	NETMAG	SEL3
	IN_ORIGERR	ORIGERR	SEL3
	IN_ORIGIN	ORIGIN	SEL3
	IN_PARRIVAL	PARRIVAL	SEL3

Schema	Synonym	Table	Table Owner
	IN_STAMAG	STAMAG	SEL3
	LASTID	LASTID	IDCX
	MSGDISC	MSGDISC	IDCX
	NA_VALUE	NA_VALUE	IDCX
	NETWORK	NETWORK	STATIC
	OUT_AMPLITUDE	AMPLITUDE	LEB
	OUT_ARRIVAL	ARRIVAL	LEB
	OUT_ASSOC	ASSOC	LEB
	OUT_EVENT_CONTROL	EVENT_CONTROL	LEB
	OUT_HYDRO_ARR_GROUP	HYDRO_ARR_GROUP	LEB
	OUT_HYDRO_ASSOC	HYDRO_ASSOC	LEB
	OUT_HYDRO_FEATURES	HYDRO_FEATURES	LEB
	OUT_INFRA_FEATURES	INFRA_FEATURES	LEB
	OUT_NETMAG	NETMAG	LEB
	OUT_ORIGERR	ORIGERR	LEB
	OUT_ORIGIN	ORIGIN	LEB
	OUT_PARRIVAL	PARRIVAL	LEB
	OUT_STAMAG	STAMAG	LEB
	PARTICIPATION	PARTICIPATION	STATIC
	PHASOR_DIST_DEPTH_RANGES	PHASOR_DIST_DEPTH_RANGES	STATIC
	PMCC_FEATURES	PMCC_FEATURES	IDCX
	REQUEST	REQUEST	IDCX
	SEISGRID	SEISGRID	STATIC
	SEISINDEX	SEISINDEX	STATIC
	SENSOR	SENSOR	STATIC
	SITE	SITE	STATIC
	SITEAUX	SITEAUX	STATIC
	SITECHAN	SITECHAN	STATIC
	SREGION	SREGION	STATIC
	STATION_PHASE_RELY	STATION_PHASE_RELY	STATIC
	WEIGHTS	WEIGHTS	STATIC
	WFDISC	WFDISC	IDCX
	WFTAG	WFTAG	SEL3
REB	AFFILIATION	AFFILIATION	STATIC
	AMPDESCRIPT	AMPDESCRIPT	IDCX
	BEAMAUX	BEAMAUX	STATIC

Schema	Synonym	Table	Table Owner
	DSEISGRID	DSEISGRID	STATIC
	DSEISINDEX	DSEISINDEX	STATIC
	GREGION	GREGION	STATIC
	INSTRUMENT	INSTRUMENT	STATIC
	LASTID	LASTID	IDCX
	NA_VALUE	NA_VALUE	IDCX
	NETWORK	NETWORK	STATIC
	PARTICIPATION	PARTICIPATION	STATIC
	SEISGRID	SEISGRID	STATIC
	SEISINDEX	SEISINDEX	STATIC
	SENSOR	SENSOR	STATIC
	SITE	SITE	STATIC
	SITEAUX	SITEAUX	STATIC
	SITECHAN	SITECHAN	STATIC
	SREGION	SREGION	STATIC
	WFDISC	WFDISC	IDCX
SEGMENT	AFFILIATION	AFFILIATION	STATIC
	ARRIVAL	ARRIVAL	REB
	ASSOC	ASSOC	REB
	INSTRUMENT	INSTRUMENT	STATIC
	NETWORK	NETWORK	STATIC
	ORIGERR	ORIGERR	REB
	ORIGIN	ORIGIN	REB
	PARTICIPATION	PARTICIPATION	STATIC
	SENSOR	SENSOR	STATIC
	SITE	SITE	STATIC
	SITEAUX	SITEAUX	STATIC
	SITECHAN	SITECHAN	STATIC
	STANET	STANET	STATIC
SEL1	AFFILIATION	AFFILIATION	STATIC
	AMP3C	AMP3C	IDCX
	AMPDESCRIPT	AMPDESCRIPT	IDCX
	AMPLITUDE	AMPLITUDE	IDCX
	APMA	APMA	IDCX
	ARRIVAL	ARRIVAL	IDCX
	ASSOC_STAPRO	ASSOC	IDCX
	BEAMAUX	BEAMAUX	STATIC

Schema	Synonym	Table	Table Owner
	DETECTION	DETECTION	IDCX
	GREGION	GREGION	STATIC
	HYDRO_ARR_GROUP	HYDRO_ARR_GROUP	IDCX
	HYDRO_ASSOC	HYDRO_ASSOC	IDCX
	HYDRO_FEATURES	HYDRO_FEATURES	IDCX
	INFRA_FEATURES	INFRA_FEATURES	IDCX
	INSTRUMENT	INSTRUMENT	STATIC
	INTERVAL	INTERVAL	IDCX
	LASTID	LASTID	IDCX
	NA_VALUE	NA_VALUE	IDCX
	NETWORK	NETWORK	STATIC
	ORIGERR_STAPRO	ORIGERR	IDCX
	ORIGIN_STAPRO	ORIGIN	IDCX
	PARTICIPATION	PARTICIPATION	STATIC
	REQUEST	REQUEST	IDCX
	SENSOR	SENSOR	STATIC
	SITE	SITE	STATIC
	SITEAUX	SITEAUX	STATIC
	SITECHAN	SITECHAN	STATIC
	SREGION	SREGION	STATIC
	STASSOC	STASSOC	IDCX
	WFDISC	WFDISC	IDCX
SEL2	AFFILIATION	AFFILIATION	STATIC
	AMP3C	AMP3C	IDCX
	AMPDESCRIPT	AMPDESCRIPT	IDCX
	AMPLITUDE	AMPLITUDE	IDCX
	APMA	APMA	IDCX
	ARRIVAL	ARRIVAL	IDCX
	ASSOC_STAPRO	ASSOC	IDCX
	BEAMAUX	BEAMAUX	STATIC
	DERVDISC	DERVDISC	IDCX
	DETECTION	DETECTION	IDCX
	GA_TAG	GA_TAG	SEL1
	GREGION	GREGION	STATIC
	HYDRO_ARR_GROUP	HYDRO_ARR_GROUP	IDCX
	HYDRO_ASSOC	HYDRO_ASSOC	IDCX
	HYDRO_FEATURES	HYDRO_FEATURES	IDCX

Schema	Synonym	Table	Table Owner
	INFRA_FEATURES	INFRA_FEATURES	IDCX
	INSTRUMENT	INSTRUMENT	STATIC
	INTERVAL	INTERVAL	IDCX
	LASTID	LASTID	IDCX
	NA_VALUE	NA_VALUE	IDCX
	NETWORK	NETWORK	STATIC
	ORIGERR_STAPRO	ORIGERR	IDCX
	ORIGIN_STAPRO	ORIGIN	IDCX
	PARTICIPATION	PARTICIPATION	STATIC
	PMCC_FEATURES	PMCC_FEATURES	IDCX
	PMCC_RECIPE	PMCC_RECIPE	IDCX
	REQUEST	REQUEST	IDCX
	SENSOR	SENSOR	STATIC
	SITE	SITE	STATIC
	SITEAUX	SITEAUX	STATIC
	SITECHAN	SITECHAN	STATIC
	SREGION	SREGION	STATIC
	STASSOC	STASSOC	IDCX
	WFDISC	WFDISC	IDCX
SEL3	AFFILIATION	AFFILIATION	STATIC
	AMP3C	AMP3C	IDCX
	AMPDESCRIPT	AMPDESCRIPT	IDCX
	AMPLITUDE	AMPLITUDE	IDCX
	APMA	APMA	IDCX
	ARRIVAL	ARRIVAL	IDCX
	ASSOC_STAPRO	ASSOC	IDCX
	BEAMAUX	BEAMAUX	STATIC
	DERVDISC	DERVDISC	IDCX
	DETECTION	DETECTION	IDCX
	GA_TAG	GA_TAG	SEL1
	GREGION	GREGION	STATIC
	HYDRO_ARR_GROUP	HYDRO_ARR_GROUP	IDCX
	HYDRO_ASSOC	HYDRO_ASSOC	IDCX
	HYDRO_FEATURES	HYDRO_FEATURES	IDCX
	INFRA_FEATURES	INFRA_FEATURES	IDCX
	INSTRUMENT	INSTRUMENT	STATIC
	INTERVAL	INTERVAL	IDCX

Schema	Synonym	Table	Table Owner
	LASTID	LASTID	IDCX
	NA_VALUE	NA_VALUE	IDCX
	NETWORK	NETWORK	STATIC
	ORIGERR_STAPRO	ORIGERR	IDCX
	ORIGIN_STAPRO	ORIGIN	IDCX
	PARTICIPATION	PARTICIPATION	STATIC
	PMCC_FEATURES	PMCC_FEATURES	IDCX
	PMCC_RECIPE	PMCC_RECIPE	IDCX
	REQUEST	REQUEST	IDCX
	SENSOR	SENSOR	STATIC
	SITE	SITE	STATIC
	SITEAUX	SITEAUX	STATIC
	SITECHAN	SITECHAN	STATIC
	SREGION	SREGION	STATIC
	STASSOC	STASSOC	IDCX
	WFDISC	WFDISC	IDCX
STATIC	LASTID	LASTID	IDCX
	NA_VALUE	NA_VALUE	IDCX

3.6. S/H/I Foreign Key Descriptions

No foreign keys found for SHI.

4. RADIONUCLIDE ENTITY RELATIONSHIPS

4.1. Overview

This chapter contains entity-relationship diagrams that illustrate the relationships between database tables for the radionuclide database schema according to conventions described in section 1.6.

4.2. Radionuclide Tables

Figures 24 through 38 illustrate the relationships among the various tables in the radionuclide (RN) database. In the following figures, a table is linked by either a single or double arrow. A single arrow indicates that a single record exists in the relationship to the table from where the record came. A double arrow indicates that one or more records exist in the relationship. For example, a single GARDS_SAMPLE_STATUS record exists for each GARDS_SAMPLE_DATA record. However, multiple GARDS_NUCL_IDED records are related to each GARDS_SAMPLE_DATA record.

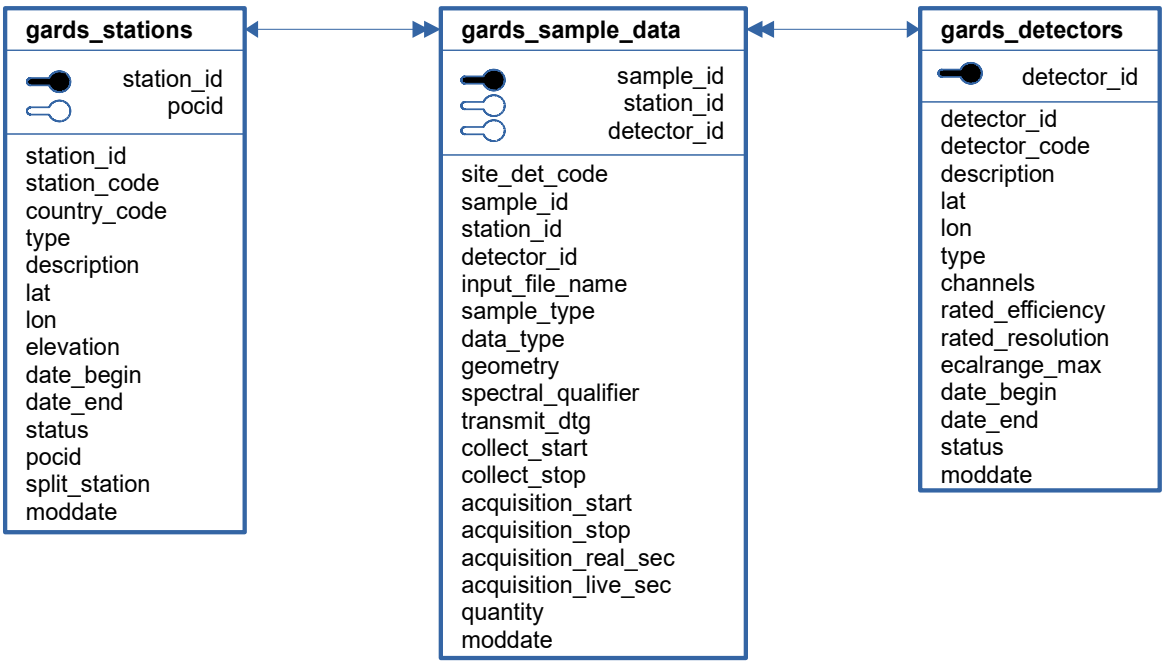


Figure 24: Radionuclide equipment

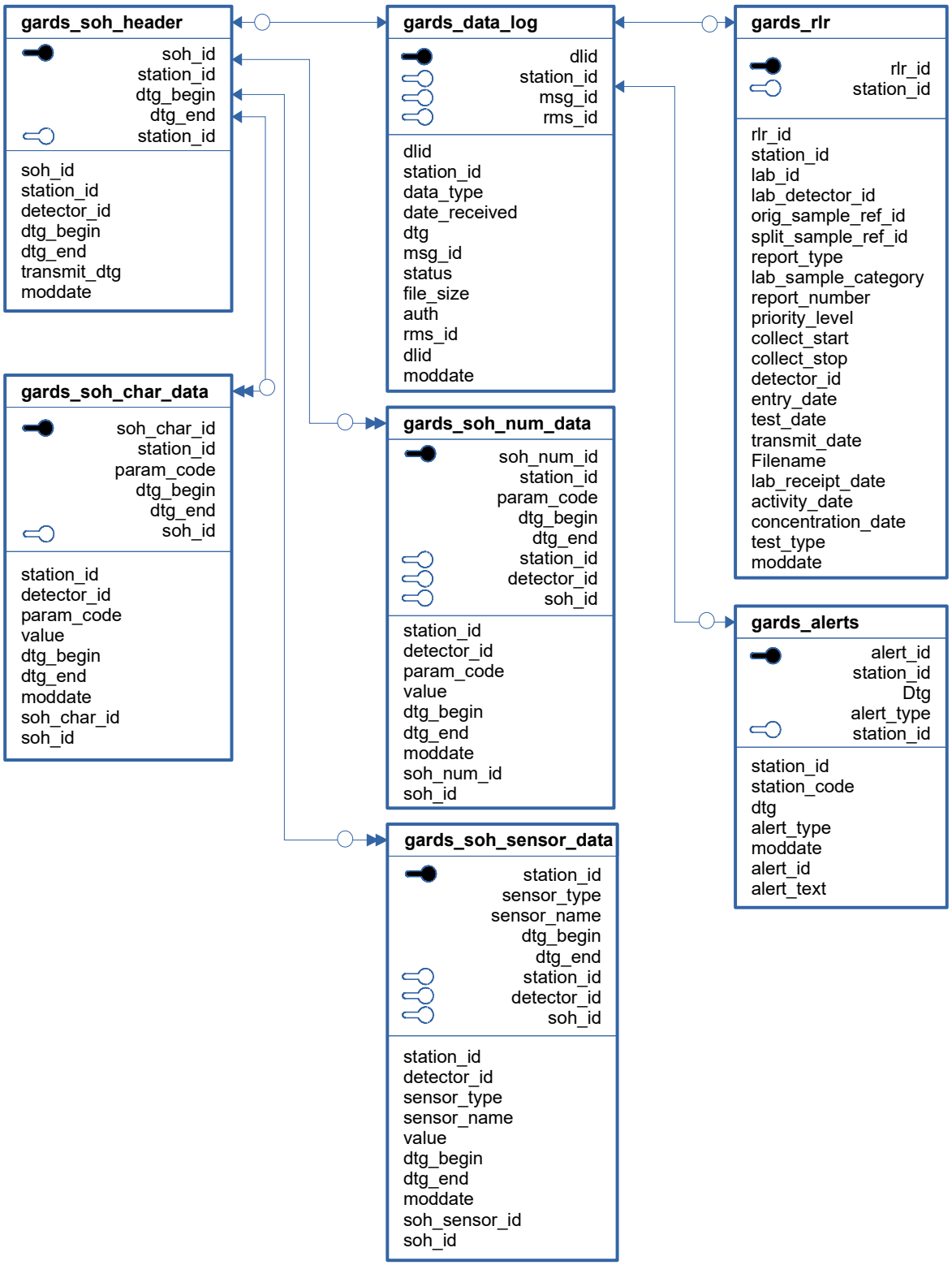


Figure 25: Raw sensor data

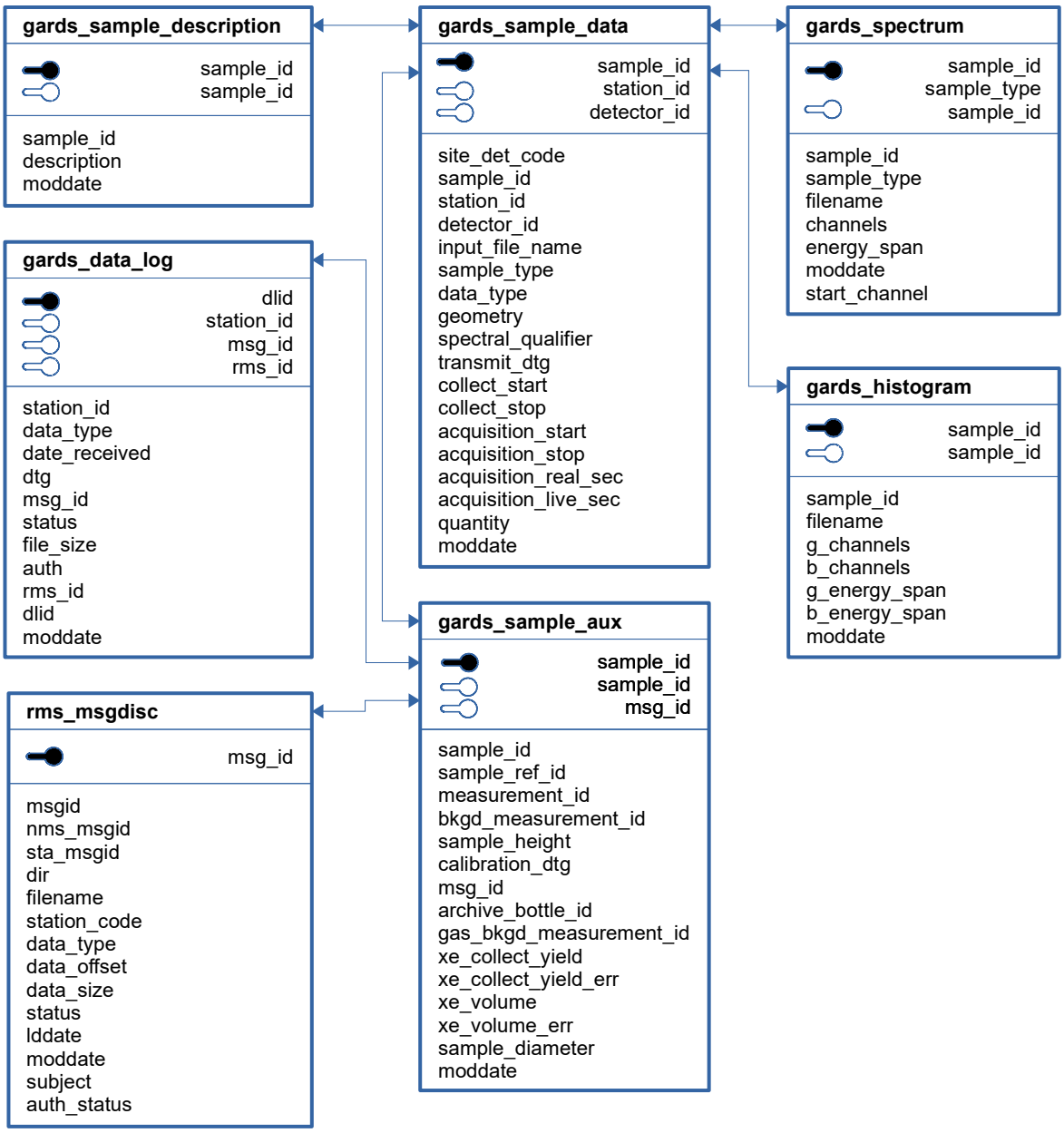


Figure 26: Sample information and pulse height data

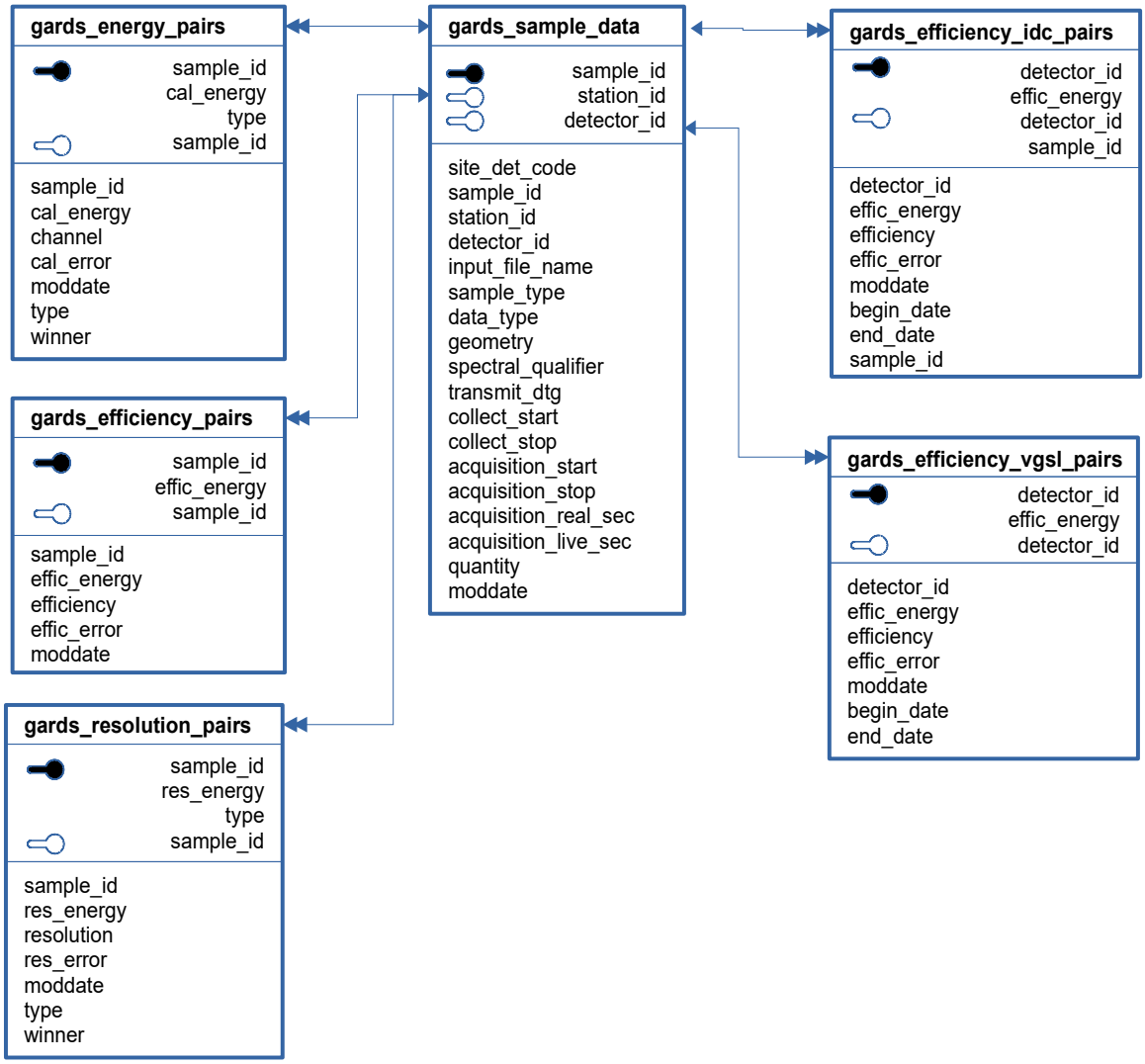


Figure 27: Gamma calibration pairs

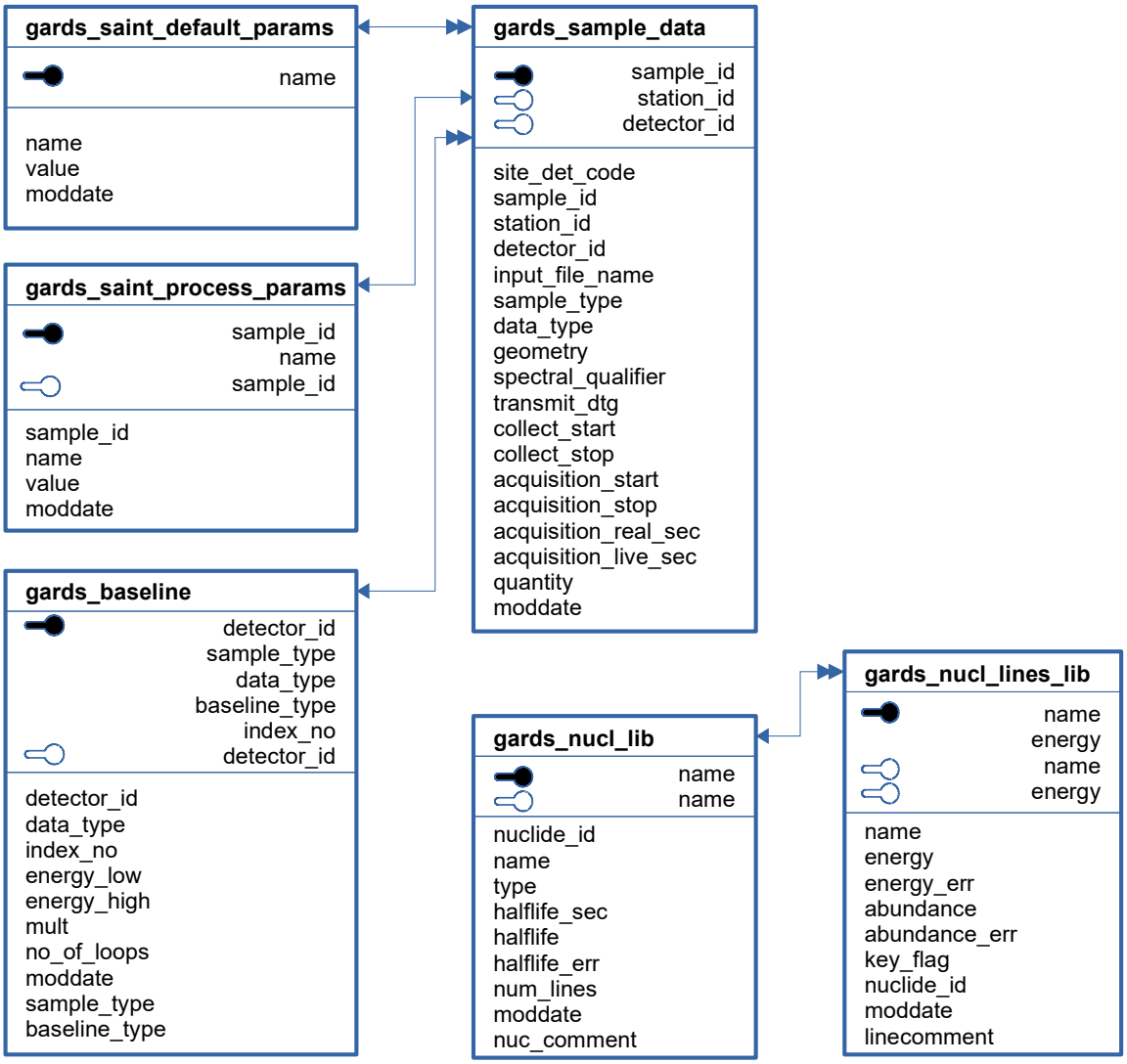


Figure 28: Processing parameters and nuclear data for gamma spectra

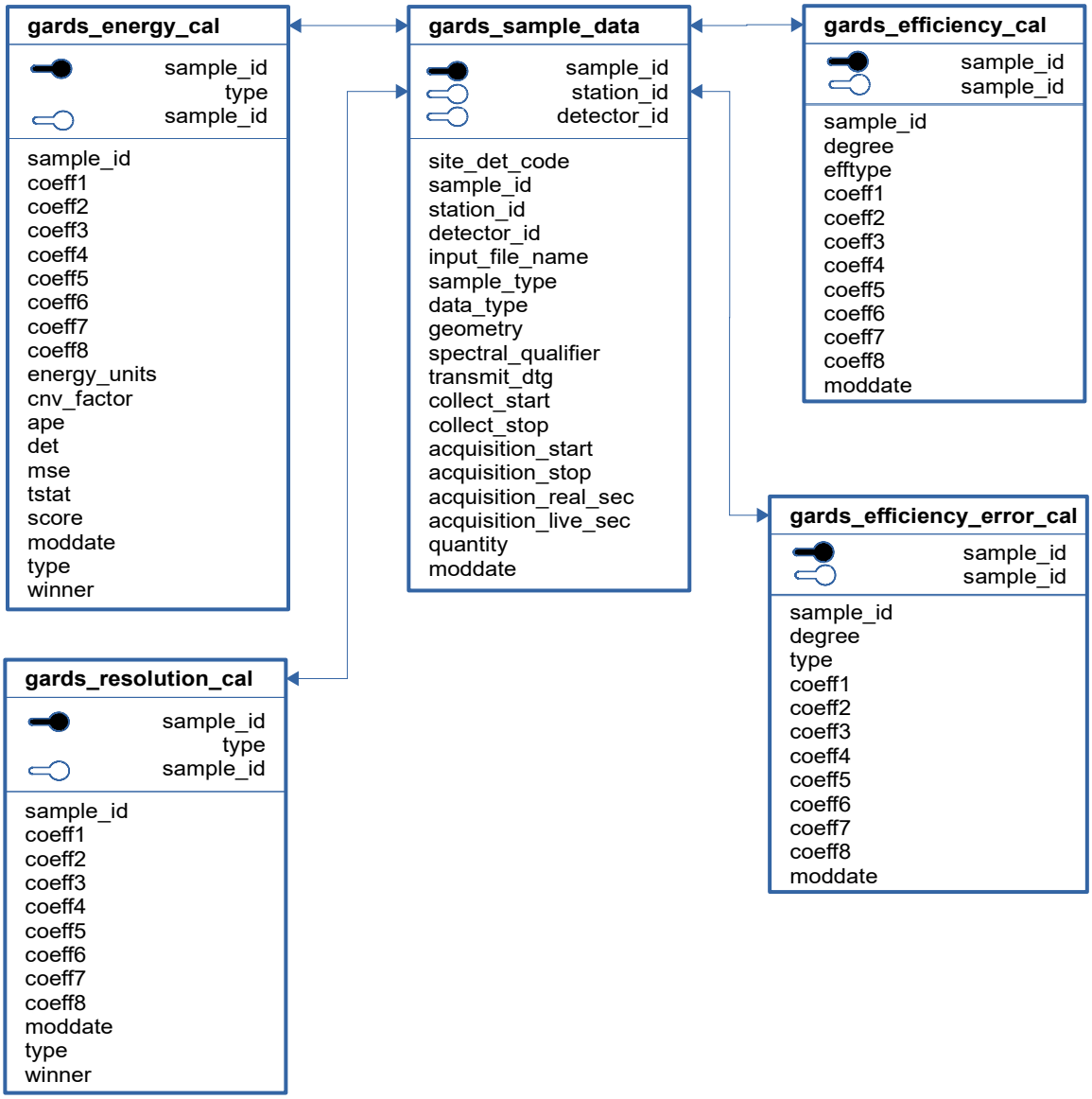


Figure 29: Calculated gamma calibration data

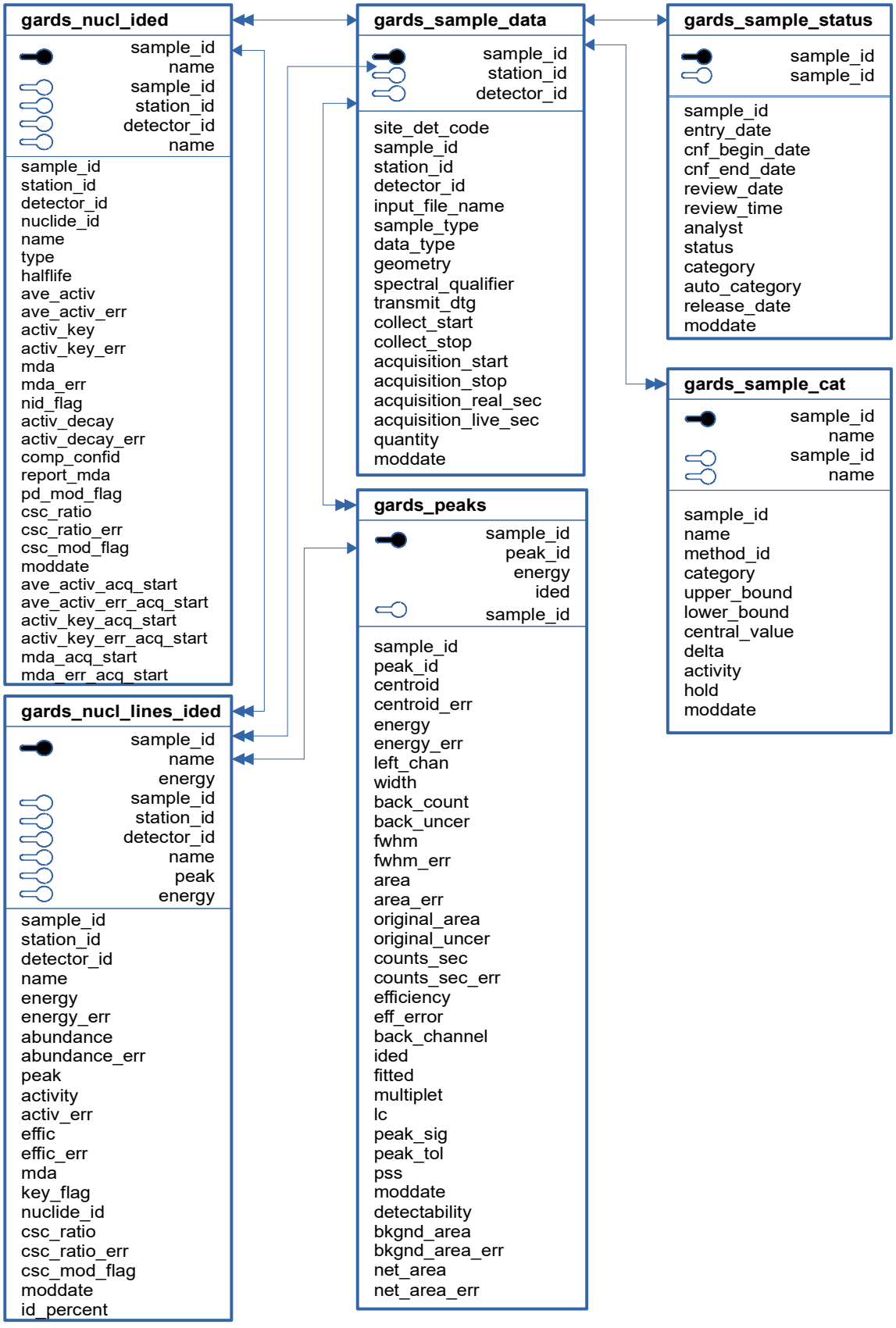


Figure 30: Results and categorization data for particulate gamma spectra

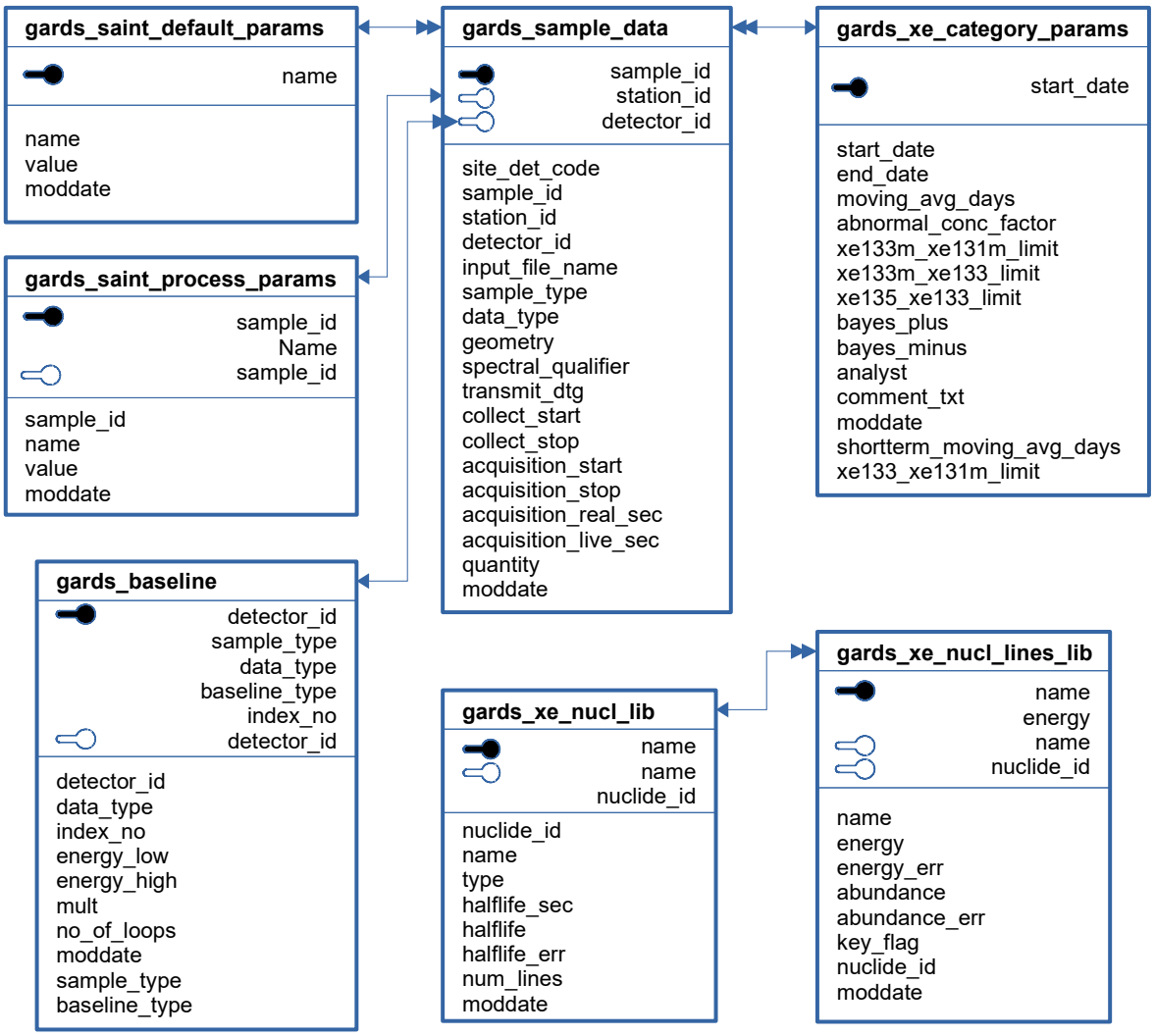


Figure 31: Processing parameters and nuclear data for SPALAX gamma spectra

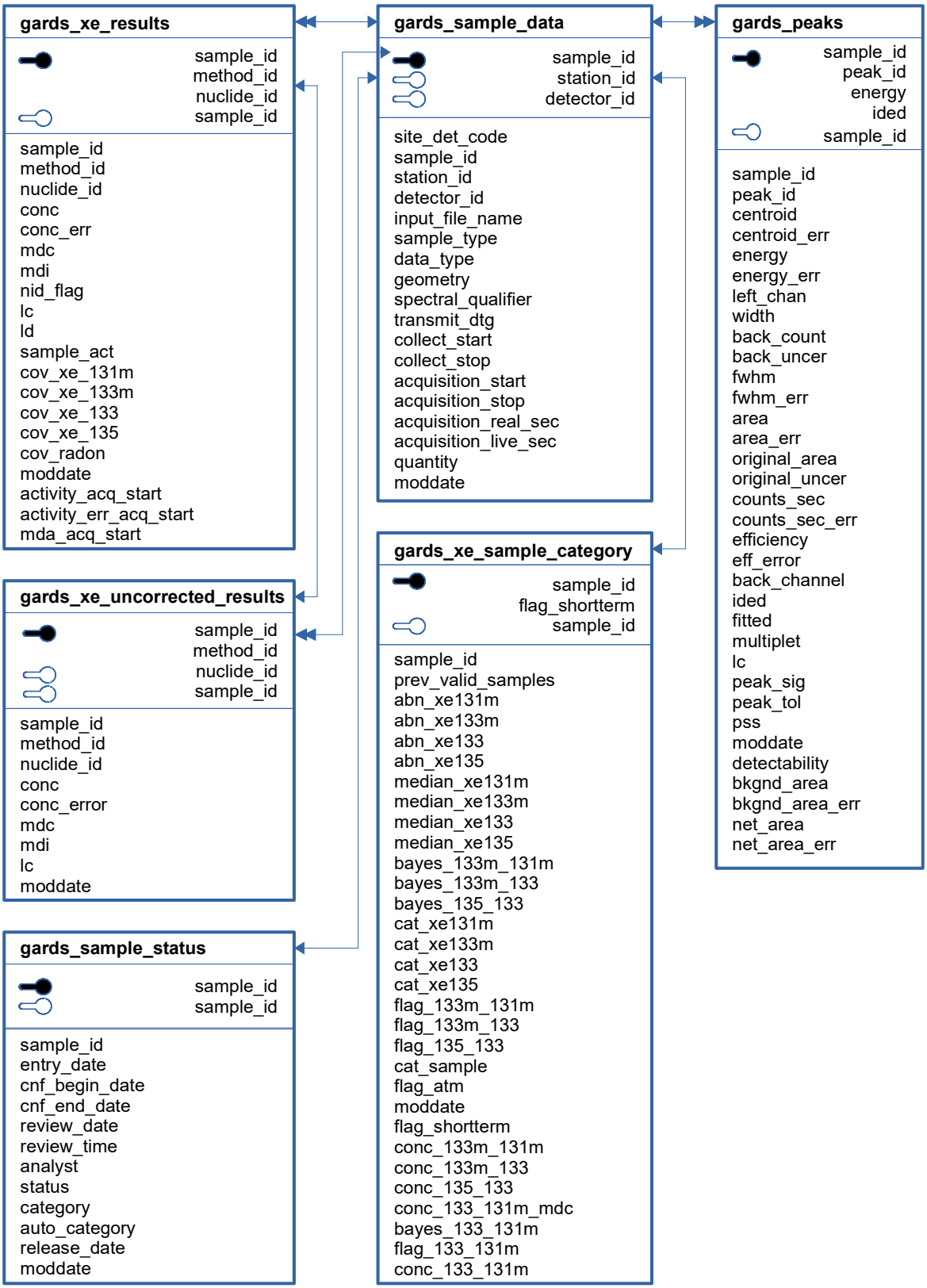


Figure 32: Results and categorization data for SPALAX gamma spectra

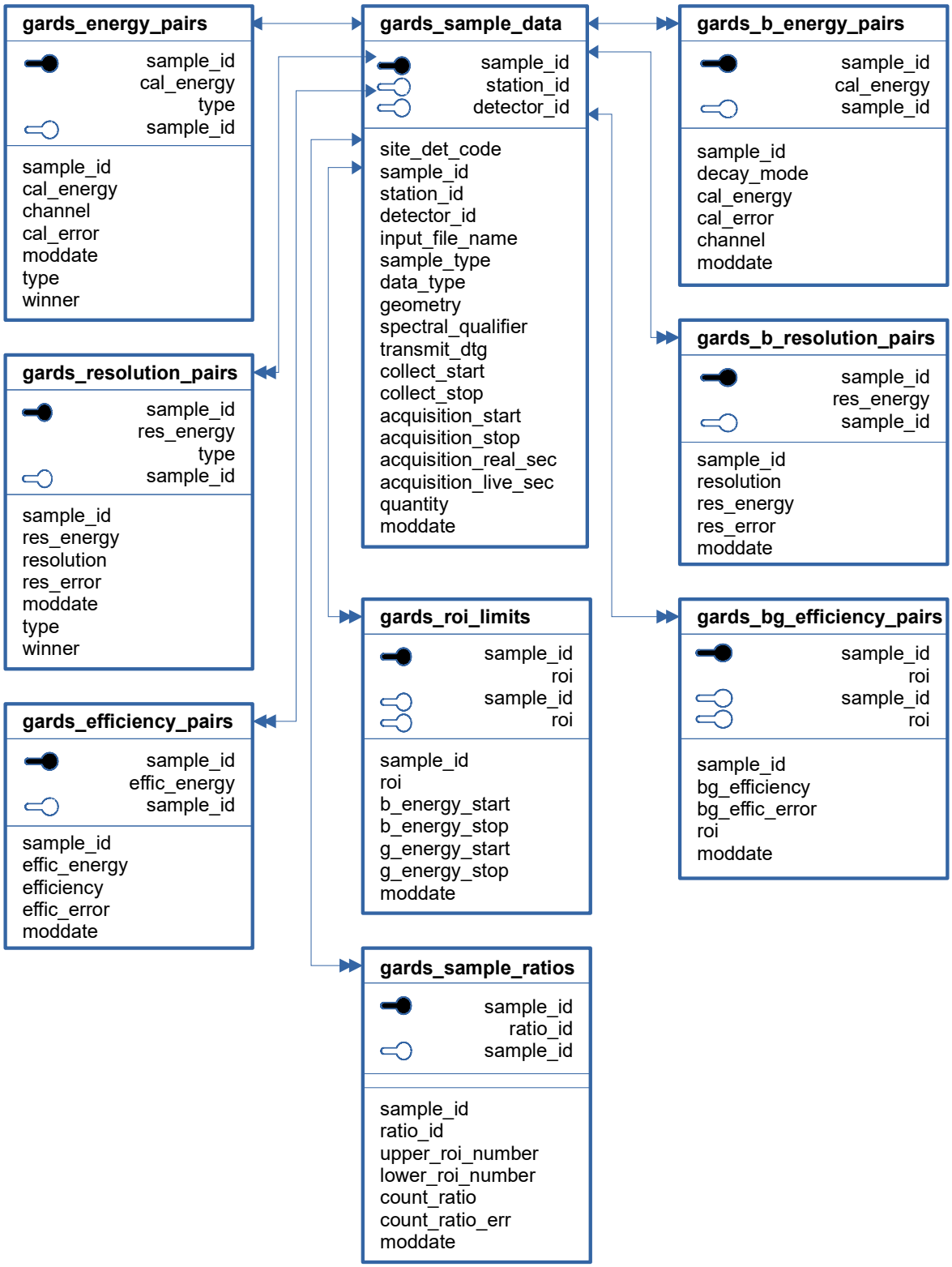


Figure 33: Beta-Gamma calibration pairs

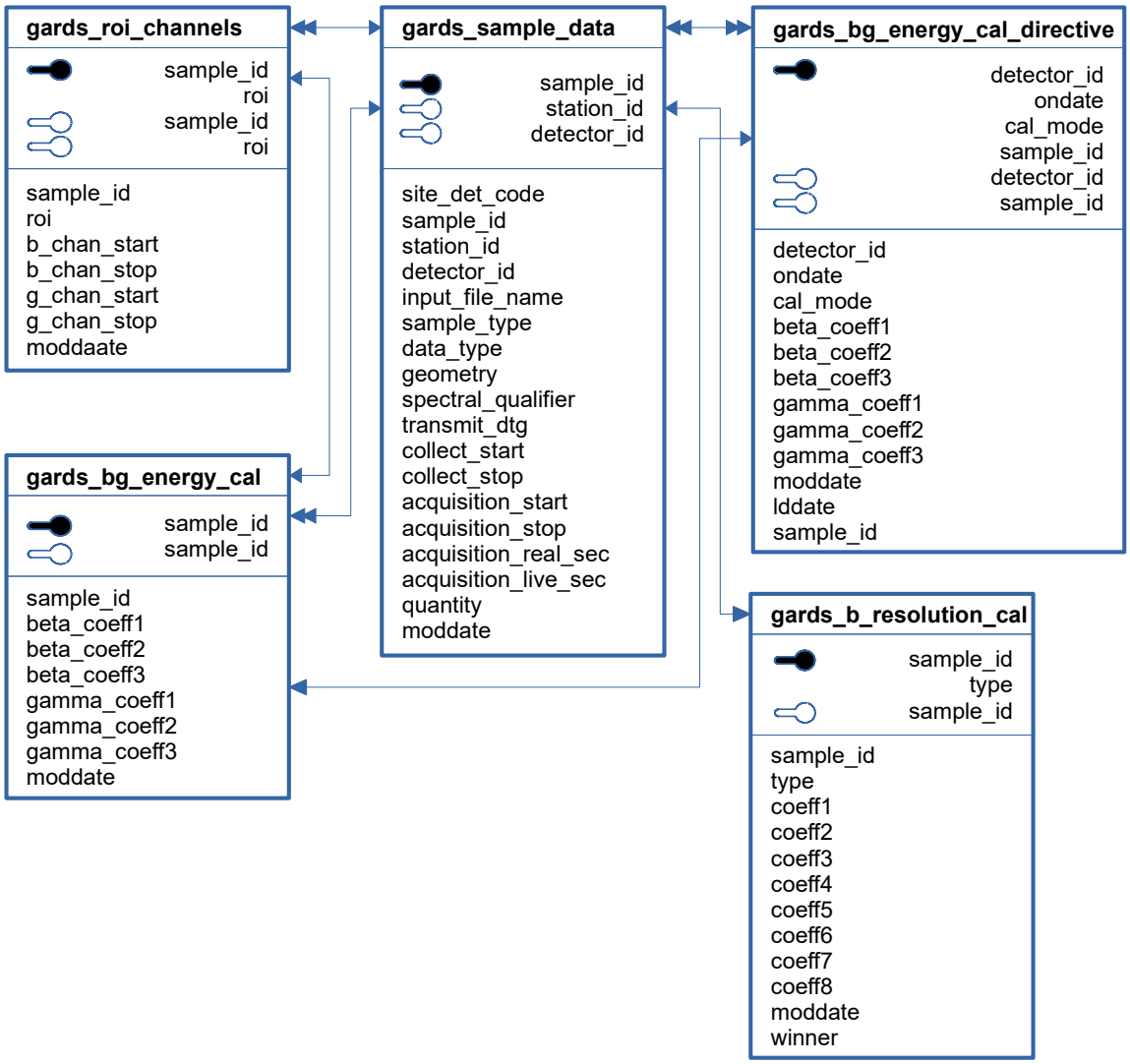


Figure 34: Calculated beta-gamma calibration data

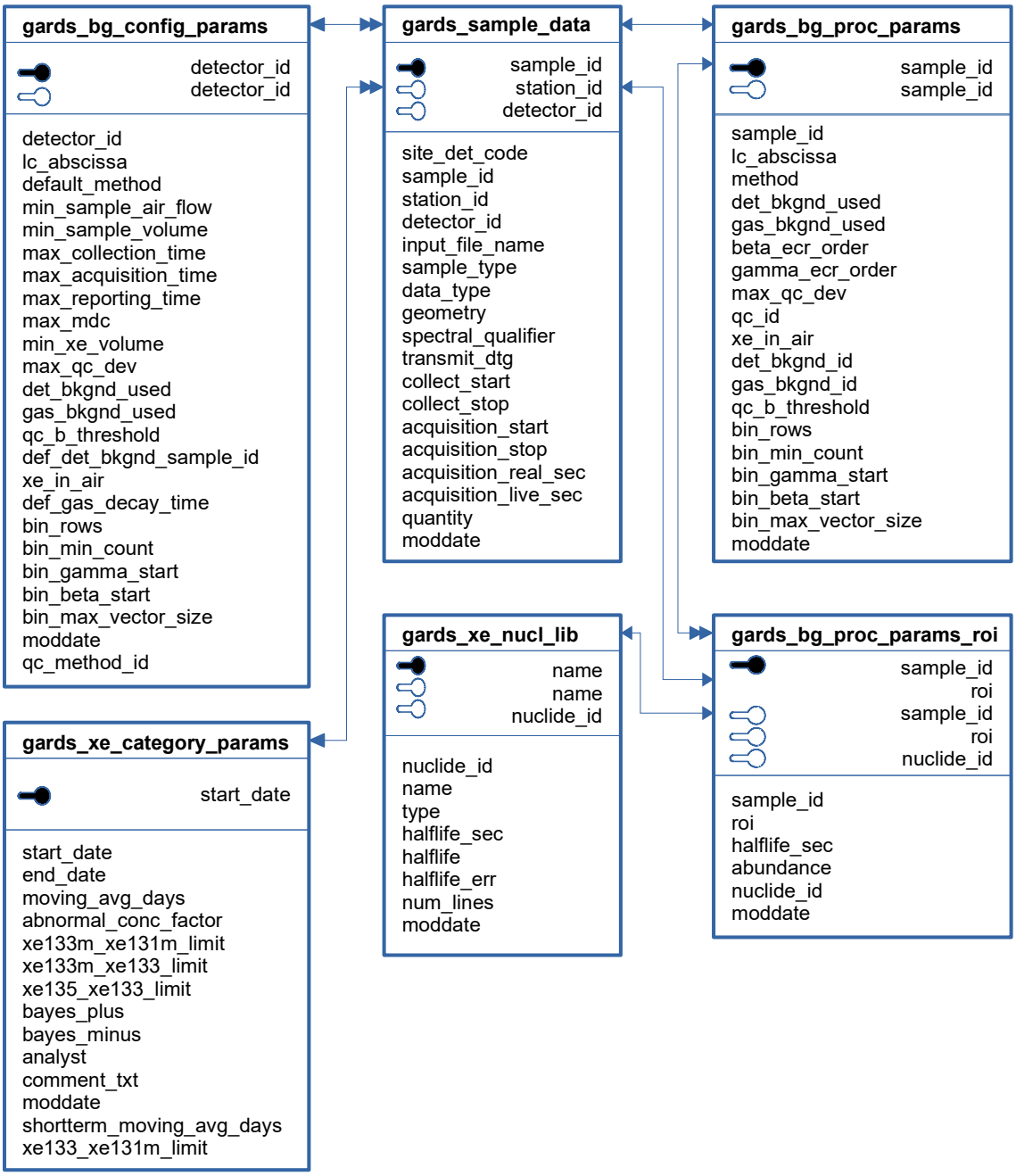


Figure 35: Processing parameters and nuclear data for SAUNA beta-gamma coincidence spectra

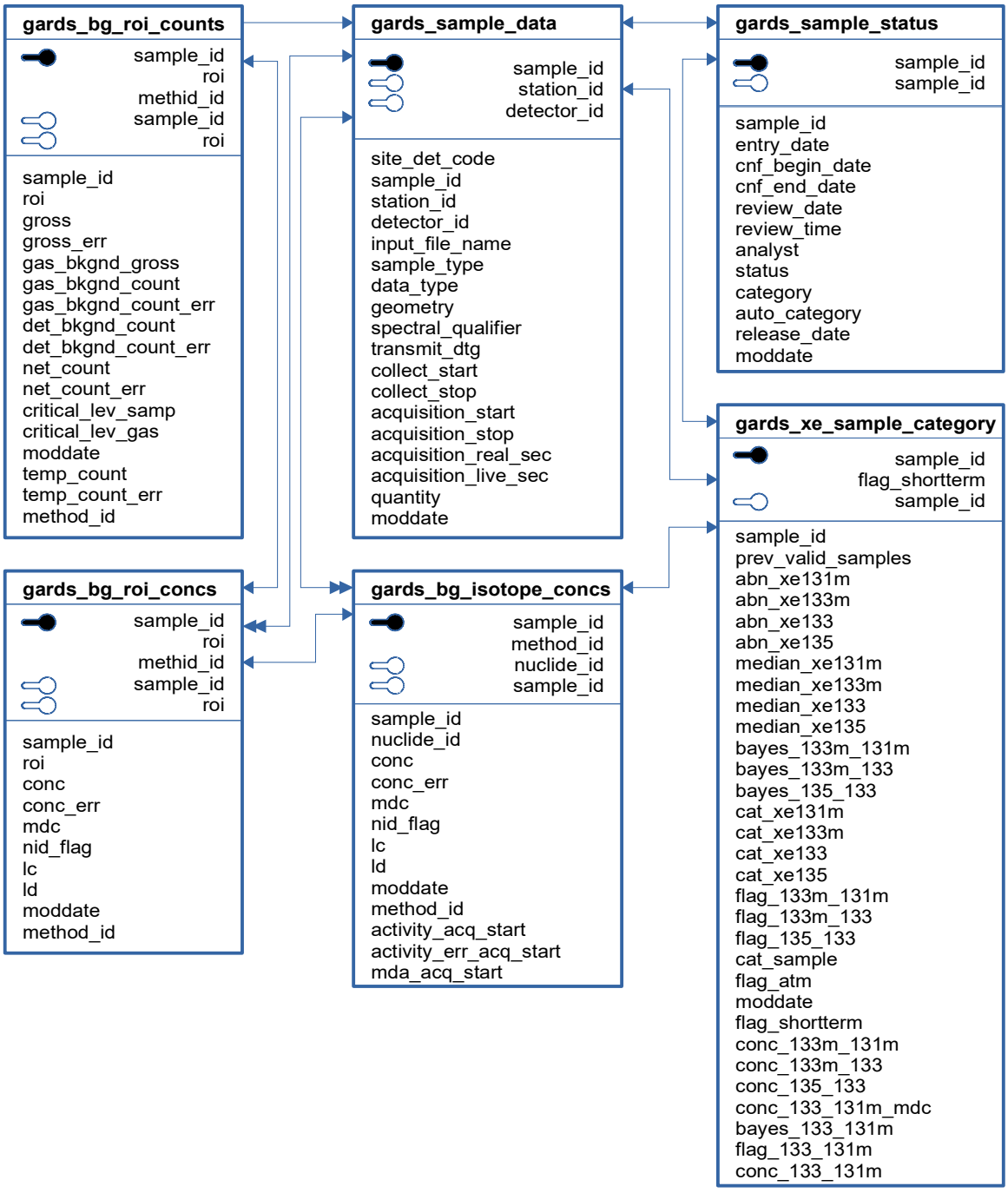


Figure 36: Results and categorization data of beta-gamma coincidence spectra

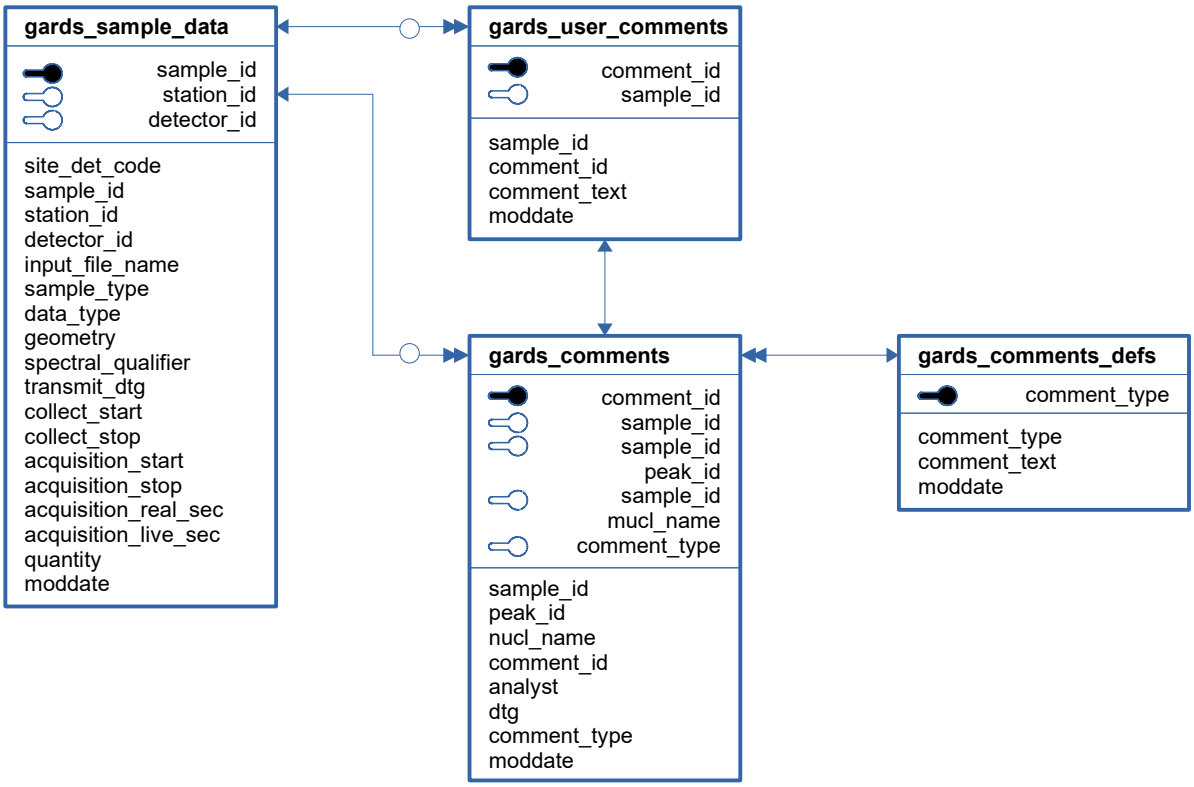


Figure 37: Analyst comment data

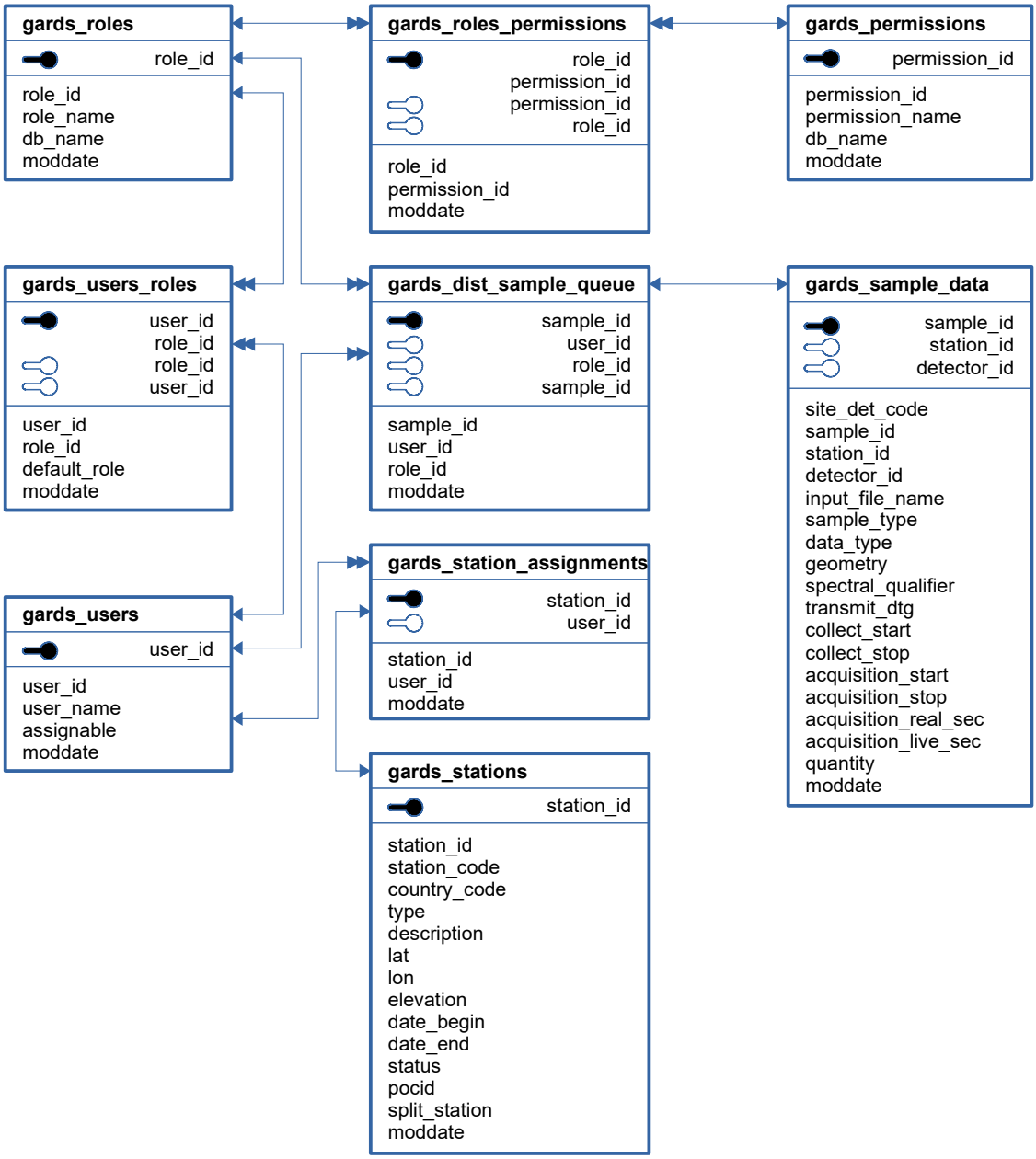


Figure 38: Multiple analyst review data

5. RADIONUCLIDE OBJECTS DESCRIPTION

5.1. Radionuclide Table Descriptions

5.1.1. GARDS_ALERTS

Owner(s)	RMSAUTO
Description	The gards_alerts table contains data describing radionuclide alert messages.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
I	station_id	*number*	Station identifier	N
	station_code	*char(5)*	Station code	Y
I	dtg	*date*	Transmit date and time	N
I	alert_type	*varchar2(20)*	Alert type	N
	moddate	*date*	Date/time at which the row was last modified	N
PI	alert_id	*number*	Unique alert identifier	N
	alert_text	*clob(4000)*	Alert message text	N

Primary key	PK_ALERTS(alert_id)
Triggers	ALERTS_S_I, ALERTS_B_I_U

5.1.2. GARDS_AUTO_SAMPLE_CAT

Owner(s)	RMSAUTO
Description	The gards_auto_sample_cat table contains original categorization values from automatic processing.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
PI	name	*varchar2(8)*	Nuclide name	N
	method_id	*number*	Method identifier	Y
	category	*number*	Categorization level	Y
	upper_bound	*number*	Upper limit of the amount of a nuclide that can be acceptably found	Y
	lower_bound	*number*	Lower limit of the amount of a nuclide that can be acceptably found	Y
	central_value	*number*	Current estimate of the level of the random process	Y
	delta	*number*	Value of a variable used in bounds estimation (EWMA algorithm)	Y

	activity	*number*	Activity concentration	Y
	hold	*number*	Flag for holding current sample from being used in the history for categorizing subsequent samples (0 = use; 1 = skip)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_AUTOSAMPCAT(sample_id, name)
Foreign keys	FK_SID_AUTOSAMPCAT(sample_id, GARDS_SAMPLE_DATA)
Triggers	AUTO_SAMPLE_CAT_B_I_U, GARDS_SAMPLE_CAT_I_U_D

5.1.3. GARDS_AUX_LIB

Owner(s)	RMSMAN
Description	The gards_aux_lib table provides additional radionuclide information that is used in the parent-daughter calculation.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
	branch_ratio	*number*	Percentage of decays to a given decay path	Y
	chain_id	*number*	Decay chain index	Y
	chain_pos	*number*	Position in the decay chain	Y
	halflife_act	*varchar2(23)*	Half-life	Y
	halflife_act_err	*varchar2(23)*	Uncertainty of half-life	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_AUX_LIB(name)
Triggers	AUX_LIB_B_I_U

5.1.4. GARDS_AUX_LINES_LIB

Owner(s)	RMSMAN
Description	The gards_aux_lines_lib table provides additional radionuclide energy line information that is used in the parent-daughter calculation.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
PI	energy	*number*	Energy of the decay line	N
PI	abundance_act	*number*	Emission intensity per disintegration	N
	abundance_act_err	*number*	Uncertainty of the abundance_act	Y
	line_comment	*varchar2(500)*	Comment associated with the decay line	Y

	line_type	*varchar2(8)*	Type of the decay line	Y
	reference	*varchar2(8)*	Reference for line properties	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_AUX_LINS_LIB(name, energy, abundance_act)
Triggers	AUX_LINES_LIB_B_I_U

5.1.5. GARDS_BASELINE

Owner(s)	RMSAUTO, RMSMAN
Description	The GARDS_BASELINE table contains default and detector specific parameters of the baseline algorithm (Smoothing-Lawn Mover)

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	detector_id	*number*	Detector identifier	N
PI	data_type	*char(1)*	Data type of the spectrum	N
PI	index_no	*number*	Index of the energy bin	N
	energy_low	*number*	Lower energy of the bin	Y
	energy_high	*number*	Higher energy of the bin	Y
	mult	*number*	Multiplicand in the baseline formula	N
	mult	*number*	Multiplicand in the baseline formula	Y
	no_of_loops	*number*	Number of loops for this bin of the baseline calculation	N
	moddate	*date*	Date/time at which the row was last modified	Y
PI	sample_type	*char(1)*	Sample type	N
PI	baseline_type	*char(1)*	Baseline type (G,B)	N

Primary key	PK_GARDS_BASELINE(detector_id, sample_type, data_type, baseline_type, index_no)
Triggers	BASELINE_B_I_U

5.1.6. GARDS_BG_CONFIG_PARAMS

Owner(s)	RMSMAN
Description	The gards_bg_config_params table holds detector specific configuration parameters for beta-gamma coincidence analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	detector_id	*number*	Detector identifier	N
	lc_abscissa	*float(126)*	Abscissas of the normal distribution corresponding to a predefined confidence level for	Y

			estimating the critical level	
	default_method	*number*	Default method identifier	Y
	min_sample_air_flow	*float(126)*	Minimum air flow accepted for sampling	Y
	min_sample_volume	*float(126)*	Minimum sample volume accepted for collection	Y
	max_collection_time	*number*	Maximum duration accepted for collection	Y
	max_acquisition_time	*number*	Maximum duration accepted for acquisition	Y
	max_reporting_time	*number*	Maximum duration accepted for reporting	Y
	max_mdc	*float(126)*	Maximum MDC value	Y
	min_xe_volume	*float(126)*	Minimum xenon volume	Y
	max_qc_dev	*number*	Maximum deviation of the quality check (integer)	Y
	det_bkgnd_used	*number*	Flag indicating whether the detector background measurement is used in the sample spectrum analysis	Y
	gas_bkgnd_used	*number*	Flag indicating whether the gas background measurement is used in the sample spectrum analysis	Y
	qc_b_threshold	*number*	Threshold for channels in the histogram used when calculating beta and gamma spectra	Y
	def_det_bkgnd_sample_id	*number*	Default detector background sample	Y
	xe_in_air	*float(126)*	Volume of stable xenon (ml) in the 1 m3 air	Y
	def_gas_decay_time	*number*	Default gas decay time	Y
	bin_rows	*number*	Number of rows to merge in binning	Y
	bin_min_count	*number*	Minimum counts in each bin	Y
	bin_gamma_start	*number*	The gamma start offset in histogram	Y
	bin_beta_start	*number*	The beta start offset in histogram	Y
	bin_max_vector_size	*number*	Maximum size for the bin vector	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	qc_method_id	*number*	QC analysis method ID	Y
	interference_used	*number*	Flag indicating whether the interference corrections are used in the sample spectrum analysis	Y

Primary key	PK_GARDS_BG_CONFIG_PARAMS(detector_id)
Foreign keys	FK_BG_CONFIG_PARAMS(detector_id, GARDS_DETECTORS)
Triggers	BG_CONFIG_PARAMS_B_I_U

5.1.7. GARDS_BG_DETECTOR_STD_SPECTRA

Owner(s)	RMSAUTO
Description	The gards_bg_detector_std_spectra tables holds the relation between a detector and standard spectra for the standard spectra method (SSM). (Note: the SSM is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	detector_id	*number*	Detector identifier	N
PI	sample_id	*number*	Sample identifier	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_DETECTOR_STD_SPEC(detector_id, sample_id)
Triggers	BG_DET_STD_SPECTRA_B_I_U

5.1.8. GARDS_BG_EFFICIENCY_PAIRS

Owner(s)	RMSAUTO
Description	The gards_bg_efficiency_pairs table contains detection efficiencies of beta gamma coincidences in the associated ROIs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	bg_efficiency	*number*	Detection efficiency of b-g coincidence event	Y
	bg_effic_error	*number*	Uncertainty of bg_efficiency	Y
PI	roi	*number*	ROI identifier	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_EFFICIENCY_PAIRS(sample_id, roi)
Foreign keys	FK_BG_EFFICIENCY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_EFFICIENCY_PAIRS_B_I_U, GARDS_BG_EFFIC_PRS_I_U_D

5.1.9. GARDS_BG_EFFICIENCY_PAIRS

Owner(s)	RMSMAN
Description	The gards_bg_efficiency_pairs table contains detection efficiencies of beta gamma coincidences in the associated ROIs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	bg_efficiency	*number*	Detection efficiency of b-g coincidence event	Y
	bg_effic_error	*number*	Uncertainty of bg_efficiency	Y
PI	roi	*number*	ROI identifier	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_EFFICIENCY_PAIRS(sample_id, roi)
Foreign keys	FK_BG_EFFICIENCY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_EFFICIENCY_PAIRS_B_I_U

5.1.10. GARDS_BG_ENERGY_CAL

Owner(s)	RMSAUTO
Description	The GARDS_BG_ENERGY_CAL table holds coefficients of linear or quadratic polynomials of beta and gamma energy calibrations that is used in the spectrum analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	beta_coeff1	*number*	Zero order coefficient of the beta energy calibration	Y
	beta_coeff2	*number*	First order coefficient of the beta energy calibration	Y
	beta_coeff3	*number*	Second order coefficient of the beta energy calibration	Y
	gamma_coeff1	*number*	Zero order coefficient of the gamma energy calibration	Y
	gamma_coeff2	*number*	First order coefficient of the gamma energy calibration	Y
	gamma_coeff3	*number*	Second order coefficient of the gamma energy calibration	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_BG_EN_CAL(sample_id)
Foreign keys	FK_BG_EN_CAL_SID(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ENERGY_CAL_B_I_U, GARDS_BG_ENERGY_CAL_I_U_D

5.1.11. GARDS_BG_ENERGY_CAL

Owner(s)	RMSMAN
Description	The GARDS_BG_ENERGY_CAL table holds coefficients of linear or quadratic polynomials of beta and gamma energy calibrations that is used in the spectrum analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	beta_coeff1	*number*	Zero order coefficient of the beta energy calibration	Y
	beta_coeff2	*number*	First order coefficient of the beta energy calibration	Y
	beta_coeff3	*number*	Second order coefficient of the beta energy calibration	Y
	gamma_coeff1	*number*	Zero order coefficient of the gamma energy calibration	Y
	gamma_coeff2	*number*	First order coefficient of the gamma energy calibration	Y
	gamma_coeff3	*number*	Second order coefficient of the gamma energy calibration	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_BG_EN_CAL(sample_id)
Foreign keys	FK_BG_EN_CAL_SID(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ENERGY_CAL_B_I_U

5.1.12. GARDS_BG_ENERGY_CAL_DIRECTIVE

Owner(s)	RMSMAN
Description	The GARDS_BG_ENERGY_CAL_DIRECTIVE table holds detector specific coefficients of linear or quadratic polynomials of beta and gamma energy calibrations, determined by calibration modes.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	detector_id	*number*	Detector identifier	N
PI	ondate	*number(8,0)*	Julian start date	N
PI	cal_mode	*char(1)*	Calibration mode	N
	beta_coeff1	*number*	Zero order coefficient of the beta energy calibration	Y
	beta_coeff2	*number*	First order coefficient of the beta energy calibration	Y
	beta_coeff3	*number*	Second order coefficient of the beta energy	Y

			calibration	
	gamma_coeff1	*number*	First order coefficient of the gamma energy calibration	Y
	gamma_coeff2	*number*	Second order coefficient of the gamma energy calibration	Y
	gamma_coeff3	*number*	Second order coefficient of the gamma energy calibration	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	lddate	*date*	Load date	Y
PI	sample_id	*number*	Sample identifier	N

Primary key	SYS_C008707817(detector_id, ondate, cal_mode, sample_id)
Foreign keys	FK_BG_EN_CAL_DIR_DETID(detector_id, GARDS_DETECTORS)
Triggers	BG_ENERGY_CAL_DIR_B_I_U

5.1.13. GARDS_BG_ENERGY_CAL_ORIG

Owner(s)	RMSAUTO
Description	The GARDS_BG_ENERGY_CAL_ORIG table holds coefficients of linear or quadratic polynomials of beta and gamma energy calibrations that are used in the original automatic analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	beta_coeff1	*number*	Zero order coefficient of the beta energy calibration	Y
	beta_coeff2	*number*	First order coefficient of the beta energy calibration	Y
	beta_coeff3	*number*	Second order coefficient of the beta energy calibration	Y
	gamma_coeff1	*number*	Zero order coefficient of the gamma energy calibration	Y
	gamma_coeff2	*number*	First order coefficient of the gamma energy calibration	Y
	gamma_coeff3	*number*	Second order coefficient of the gamma energy calibration	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_BG_EN_CAL_ORIG(sample_id)
Foreign keys	FK_BG_EN_CAL_ORIG_SID(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ENERGY_CAL_ORIG_B_I_U, GARDS_BG_ENERGY_CAL_ORIG_I_U_D

5.1.14. GARDS_BG_ENERGY_CAL_ORIG

Owner(s)	RMSMAN
Description	The GARDS_BG_ENERGY_CAL_ORIG table holds coefficients of linear or quadratic polynomials of beta and gamma energy calibrations that are used in the original automatic analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	beta_coeff1	*number*	Zero order coefficient of the beta energy calibration	Y
	beta_coeff2	*number*	First order coefficient of the beta energy calibration	Y
	beta_coeff3	*number*	Second order coefficient of the beta energy calibration	Y
	gamma_coeff1	*number*	Zero order coefficient of the gamma energy calibration	Y
	gamma_coeff2	*number*	First order coefficient of the gamma energy calibration	Y
	gamma_coeff3	*number*	Second order coefficient of the gamma energy calibration	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_BG_EN_CAL_ORIG(sample_id)
Foreign keys	FK_BG_EN_CAL_ORIG_SID(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ENERGY_CAL_ORIG_B_I_U

5.1.15. GARDS_BG_ISOTOPE_CONCS

Owner(s)	RMSAUTO
Description	The GARDS_BG_ISOTOPE_CONCS contains activity and activity concentration results of xenon isotopes by beta gamma coincidence analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	nuclide_id	*number*	Nuclide identifier	N
	conc	*float(126)*	Activity concentration	Y
	conc_err	*float(126)*	Uncertainty of the activity concentration	Y
	mdc	*float(126)*	Minimum detectable concentration	Y
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	lc	*float(126)*	Critical level in activity concentration	Y
	ld	*float(126)*	Detection limit in activity concentration	Y

	moddate	*date*	Date/time at which the row was last modified	Y
PI	method_id	*number*	Method identifier	N
	activity_acq_start	*float(126)*	Activity at the start of acquisition	Y
	activity_err_acq_start	*float(126)*	Uncertainty of the activity at the start of acquisition	Y
	mda_acq_start	*float(126)*	Minimum detectable activity at the start of acquisition	Y

Primary key	PK_GARDS_BG_ISOTOPE_CONCS(sample_id, nuclide_id, method_id)
Foreign keys	FK_BG_ISOTOPE_CONCS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ISOTOPE_CONCS_B_I_U, GARDS_BG_ISOTOPE_CONCS_I_U_D

5.1.16. GARDS_BG_ISOTOPE_CONCS

Owner(s)	RMSMAN
Description	The GARDS_BG_ISOTOPE_CONCS contains activity and activity concentration results of xenon isotopes by beta gamma coincidence analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	nuclide_id	*number*	Nuclide identifier	N
	conc	*float(126)*	Activity concentration	Y
	conc_err	*float(126)*	Uncertainty of the activity concentration	Y
	mdc	*float(126)*	Minimum detectable concentration	Y
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	lc	*float(126)*	Critical level in activity concentration	Y
	ld	*float(126)*	Detection limit in activity concentration	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	method_id	*number*	Method identifier	N
	activity_acq_start	*float(126)*	Activity at the start of acquisition	Y
	activity_err_acq_start	*float(126)*	Uncertainty of the activity at the start of acquisition	Y
	mda_acq_start	*float(126)*	Minimum detectable activity at the start of acquisition	Y

Primary key	PK_GARDS_BG_ISOTOPE_CONCS(sample_id, nuclide_id, method_id)
Foreign keys	FK_BG_ISOTOPE_CONCS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ISOTOPE_CONCS_B_I_U

5.1.17. GARDS_BG_PROC_PARAMS

Owner(s)	RMSMAN
Description	The gards_bg_proc_params table holds spectrum specific parameters of beta-gamma processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	lc_abscissa	*float(126)*	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level	Y
	method	*number*	Method identifier	Y
	det_bkgnd_used	*number*	Flag indicating whether the detector background measurement is used in the sample spectrum analysis	Y
	gas_bkgnd_used	*number*	Flag indicating whether the gas background measurement is used in the sample spectrum analysis	Y
	beta_ecr_order	*number*	Order of the beta energy calibration function	Y
	gamma_ecr_order	*number*	Order of the gamma energy calibration function	Y
	max_qc_dev	*number*	Maximum deviation of the quality check (integer)	Y
	qc_id	*number*	Unique quality control identifier	Y
	xe_in_air	*float(126)*	Volume of stable xenon (ml) in the 1 m3 air	Y
	det_bkgnd_id	*number*	Unique identifier for the gas background used in the calculation	Y
	gas_bkgnd_id	*number*	Unique gas background identifier	Y
	qc_b_threshold	*number*	The value that is used to mask the gamma spectrum in QC control of the spectrum	Y
	bin_rows	*number*	Number of rows to merge in binning	Y
	bin_min_count	*number*	Minimum counts in each bin	Y
	bin_gamma_start	*number*	The gamma start offset in histogram	Y
	bin_beta_start	*number*	The beta start offset in histogram	Y
	bin_max_vector_size	*number*	Maximum size for the bin vector	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	interference_used	*number*	Flag indicating whether the interference corrections are used in the sample spectrum analysis	Y

Primary key	PK_GARDS_BG_PROC_PARAMS(sample_id)
Foreign keys	FK_BG_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_PROC_PARAMS_B_I_U

5.1.18. GARDS_BG_PROC_PARAMS

Owner(s)	RMSAUTO
Description	The gards_bg_proc_params table holds spectrum specific parameters of beta-gamma processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	lc_abscissa	*float(126)*	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level	Y
	method	*number*	Method identifier	Y
	det_bkgnd_used	*number*	Flag indicating whether the detector background measurement is used in the sample spectrum analysis	Y
	gas_bkgnd_used	*number*	Flag indicating whether the gas background measurement is used in the sample spectrum analysis	Y
	beta_ecr_order	*number*	Order of the beta energy calibration function	Y
	gamma_ecr_order	*number*	Order of the gamma energy calibration function	Y
	max_qc_dev	*number*	Maximum deviation of the quality check (integer)	Y
	qc_id	*number*	Unique quality control identifier	Y
	xe_in_air	*float(126)*	Volume of stable xenon (ml) in the 1 m3 air	Y
	det_bkgnd_id	*number*	Unique identifier for the gas background used in the calculation	Y
	gas_bkgnd_id	*number*	Unique gas background identifier	Y
	qc_b_threshold	*number*	The value that is used to mask the gamma spectrum in QC control of the spectrum	Y
	bin_rows	*number*	Number of rows to merge in binning	Y
	bin_min_count	*number*	Minimum counts in each bin	Y
	bin_gamma_start	*number*	The gamma start offset in histogram	Y
	bin_beta_start	*number*	The beta start offset in histogram	Y
	bin_max_vector_size	*number*	Maximum size for the bin vector	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	interference_used	*number*	Flag indicating whether the interference cor-	Y

			rections are used in the sample spectrum analysis	

Primary key	PK_GARDS_BG_PROC_PARAMS(sample_id)
Foreign keys	FK_BG_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_PROC_PARAMS_B_I_U, GARDS_BG_PROC_PARAMS_I_U_D

5.1.19. GARDS_BG_PROC_PARAMS_ROI

Owner(s)	RMSMAN
Description	The GARDS_BG_PROC_PARAMS_ROI contains regions of interest (ROIs) of beta gamma processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	halflife_sec	*number*	Half-life in seconds	Y
	abundance	*float(126)*	Emission intensity per disintegration	Y
	nuclide_id	*number*	Nuclide identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_PROC_PARAMS_ROI(sample_id, roi)
Foreign keys	FK_BG_PROC_PARAMS_ROI(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_PROC_PARAMS_ROI_B_I_U

5.1.20. GARDS_BG_PROC_PARAMS_ROI

Owner(s)	RMSAUTO
Description	The GARDS_BG_PROC_PARAMS_ROI contains regions of interest (ROIs) of beta gamma processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	halflife_sec	*number*	Half-life in seconds	Y
	abundance	*float(126)*	Emission intensity per disintegration	Y
	nuclide_id	*number*	Nuclide identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_PROC_PARAMS_ROI(sample_id, roi)
Foreign keys	FK_BG_PROC_PARAMS_ROI(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_PROC_PARAMS_ROI_B_I_U, GARDS_BG_PROC_PARAMS_ROI_I_U_D

5.1.21. GARDS_BG_QC_RESULT

Owner(s)	RMSAUTO
Description	Holds results of beta gamma quality control (QC) processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	amplitude	*float(126)*	Amplitude of QC peak	Y
	fwhm	*float(126)*	Full width at half maximum	Y
	centroid	*float(126)*	Centroid energy of QC peak	Y
	offset	*float(126)*	Noise offset in QC spectrum	Y
	slope	*float(126)*	Noise slope in QC spectrum	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_QC_RESULT(sample_id)
Foreign keys	FK_BG_QC_RESULT(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_QC_RESULT_B_I_U, GARDS_BG_QC_RESULT_I_U_D

5.1.22. GARDS_BG_QC_RESULT

Owner(s)	RMSMAN
Description	Holds results of beta gamma quality control (QC) processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	amplitude	*float(126)*	Amplitude of QC peak	Y
	fwhm	*float(126)*	Full width at half maximum	Y
	centroid	*float(126)*	Centroid energy of QC peak	Y
	offset	*float(126)*	Noise offset in QC spectrum	Y
	slope	*float(126)*	Noise slope in QC spectrum	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_QC_RESULT(sample_id)
Foreign keys	FK_BG_QC_RESULT(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_QC_RESULT_B_I_U

5.1.23. GARDS_BG_ROI_CONCS

Owner(s)	RMSMAN
Description	The GARDS_BG_ROI_CONCS contains ROI specific activity and activity concentration results by beta gamma coincidence analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	conc	*float(126)*	Activity concentration	Y
	conc_err	*float(126)*	Uncertainty of the activity concentration	Y
	mdc	*float(126)*	Minimum detectable concentration	Y
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	lc	*float(126)*	Critical level in activity concentration	Y
	ld	*float(126)*	Detection limit in activity concentration	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	method_id	*number(38,0)*	Method identifier	N

Primary key	PK_GARDS_BG_ROI_CONCS(sample_id, roi, method_id)
Foreign keys	FK_BG_ROI_CONCS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ROI_CONCS_B_I_U

5.1.24. GARDS_BG_ROI_CONCS

Owner(s)	RMSAUTO
Description	The GARDS_BG_ROI_CONCS contains ROI specific activity and activity concentration results by beta gamma coincidence analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	conc	*float(126)*	Activity concentration	Y
	conc_err	*float(126)*	Uncertainty of the activity concentration	Y
	mdc	*float(126)*	Minimum detectable concentration	Y
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	lc	*float(126)*	Critical level in activity concentration	Y
	ld	*float(126)*	Detection limit in activity concentration	Y
	moddate	*date*	Date/time at which the row was last modified	Y

PI	method_id	*number(38,0)*	Method identifier	N

Primary key	PK_GARDS_BG_ROI_CONCS(sample_id, roi, method_id)
Foreign keys	FK_BG_ROI_CONCS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ROI_CONCS_B_I_U, GARDS_BG_ROI_CONCS_I_U_D

5.1.25. GARDS_BG_ROI_COUNTS

Owner(s)	RMSAUTO
Description	The GARDS_BG_ROI_COUNTS contains ROI specific net count results by beta gamma coincidence analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	gross	*number*	Gross number of counts in the sample spectrum	Y
	gross_err	*number*	Uncertainty of the gross number of counts in the sample spectrum	Y
	gas_bkgnd_gross	*number*	Gross number of counts in the gas background spectrum	Y
	gas_bkgnd_count	*number*	Net number of counts in the gas background spectrum	Y
	gas_bkgnd_count_err	*number*	Uncertainty of the net number of counts in the gas background spectrum	Y
	det_bkgnd_count	*number*	Gross number of counts in the detector background spectrum	Y
	det_bkgnd_count_err	*number*	Uncertainty of the gross number of counts in the detector background spectrum	Y
	net_count	*number*	Net number of counts in the ROI	Y
	net_count_err	*number*	Uncertainty of the net number of counts in the ROI	Y
	critical_lev_samp	*number*	Critical level in counts in the sample spectrum	Y
	critical_lev_gas	*number*	Critical level in counts in the gas background spectrum	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	temp_count	*number*	Net number of counts in the ROI by subtracting the detector background and interference corrections	Y
	temp_count_err	*number*	Uncertainty of the net number of counts in the ROI by subtracting the detector background and interference corrections	Y

PI	method_id	*number(38,0)*	Method identifier	N

Primary key	PK_GARDS_BG_ROI_COUNTS(sample_id, roi, method_id)
Foreign keys	FK_BG_ROI_COUNTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ROI_COUNTS_B_I_U, GARDS_BG_ROI_COUNTS_I_U_D

5.1.26. GARDS_BG_ROI_COUNTS

Owner(s)	RMSMAN
Description	The GARDS_BG_ROI_COUNTS contains ROI specific net count results by beta gamma coincidence analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	gross	*number*	Gross number of counts in the sample spectrum	Y
	gross_err	*number*	Uncertainty of the gross number of counts in the sample spectrum	Y
	gas_bkgnd_gross	*number*	Gross number of counts in the gas background spectrum	Y
	gas_bkgnd_count	*number*	Net number of counts in the gas background spectrum	Y
	gas_bkgnd_count_err	*number*	Uncertainty of the net number of counts in the gas background spectrum	Y
	det_bkgnd_count	*number*	Gross number of counts in the detector background spectrum	Y
	det_bkgnd_count_err	*number*	Uncertainty of the gross number of counts in the detector background spectrum	Y
	net_count	*number*	Net number of counts in the ROI	Y
	net_count_err	*number*	Uncertainty of the net number of counts in the ROI	Y
	critical_lev_samp	*number*	Critical level in counts in the sample spectrum	Y
	critical_lev_gas	*number*	Critical level in counts in the gas background spectrum	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	temp_count	*number*	Net number of counts in the ROI by subtracting the detector background and interference corrections	Y
	temp_count_err	*number*	Uncertainty of the net number of counts in the ROI by subtracting the detector background and interference corrections	Y

PI	method_id	*number(38,0)*	Method identifier	N

Primary key	PK_GARDS_BG_ROI_COUNTS(sample_id, roi, method_id)
Foreign keys	FK_BG_ROI_COUNTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_ROI_COUNTS_B_I_U

5.1.27. GARDS_BG_STD_SPECTRA

Owner(s)	RMSAUTO
Description	The gards_bg_std_spectra table holds standard spectra for the SSM. (Note: The SSM is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	number_of_decays	*number*	Number of decays for this standard spectra	Y
	nuclide_id	*number*	Nuclide identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_STD_SPECTRA(sample_id)
Foreign keys	FK_BG_STD_SPECTRA(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_STD_SPECTRA_B_I_U

5.1.28. GARDS_BG_STD_SPECTRA_RESULT

Owner(s)	RMSMAN
Description	The gards_bg_std_spectra_result table holds results by the SSM. (Note: The SSM is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	std_spectra_id	*number*	Identifier for the standard spectra	N
	gamma_coeff1	*float(126)*	Zero order coefficient of the gamma energy calibration	Y
	gamma_coeff2	*float(126)*	First order coefficient of the gamma energy calibration	Y
	gamma_coeff3	*float(126)*	Second order coefficient of the gamma energy calibration	Y
	beta_coeff1	*float(126)*	Zero order coefficient of the beta energy calibration	Y
	beta_coeff2	*float(126)*	First order coefficient of the beta energy calibration	Y
	beta_coeff3	*float(126)*	Second order coefficient of the beta energy	Y

			calibration	
	estimate	*float(126)*	Estimate	Y
	error	*float(126)*	Uncertainty of the estimate	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_STD_SPECTRA_RESULT(sample_id, std_spectra_id)
Foreign keys	FK_BG_STD_SPECTRA_RESULT(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_STD_SPECTRA_RESULT_B_I_U

5.1.29. GARDS_BG_STD_SPECTRA_RESULT

Owner(s)	RMSAUTO
Description	The gards_bg_std_spectra_result table holds results by the SSM. (Note: The SSM is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	std_spectra_id	*number*	Identifier for the standard spectra	N
	gamma_coeff1	*float(126)*	Zero order coefficient of the gamma energy calibration	Y
	gamma_coeff2	*float(126)*	First order coefficient of the gamma energy calibration	Y
	gamma_coeff3	*float(126)*	Second order coefficient of the gamma energy calibration	Y
	beta_coeff1	*float(126)*	Zero order coefficient of the beta energy calibration	Y
	beta_coeff2	*float(126)*	First order coefficient of the beta energy calibration	Y
	beta_coeff3	*float(126)*	Second order coefficient of the beta energy calibration	Y
	estimate	*float(126)*	Estimate	Y
	error	*float(126)*	Uncertainty of the estimate	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_BG_STD_SPECTRA_RESULT(sample_id, std_spectra_id)
Foreign keys	FK_BG_STD_SPECTRA_RESULT(sample_id, GARDS_SAMPLE_DATA)
Triggers	BG_STD_SPECTRA_RESULT_B_I_U, BG_STD_SPECTRA_RESULT_I_U_D

5.1.30. GARDS_B_ENERGY_PAIRS

Owner(s)	RMSAUTO

Description	The gards_b_energy_pairs table contains beta energy calibration pairs.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
	decay_mode	*char(1)*	Decay type	Y
PI	cal_energy	*number*	Beta energy	N
	cal_error	*number*	Uncertainty of the centroid channel	Y
	channel	*number*	Peak centroid channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_ENERGY_PAIRS(sample_id, cal_energy, type)
Foreign keys	FK_B_ENERGY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_B_ENRGY_PRS_INSERT, B_ENERGY_PAIRS_B_I_U

5.1.31. GARDS_B_ENERGY_PAIRS

Owner(s)	RMSMAN
Description	The gards_b_energy_pairs table contains beta energy calibration pairs.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
	decay_mode	*char(1)*	Decay type	Y
PI	cal_energy	*number*	Beta energy	N
	cal_error	*number*	Uncertainty of the centroid channel	Y
	channel	*number*	Peak centroid channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_ENERGY_PAIRS(sample_id, cal_energy, type)
Foreign keys	FK_B_ENERGY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	B_ENERGY_PAIRS_B_I_U

5.1.32. GARDS_B_ENERGY_PAIRS_ORIG

Owner(s)	RMSAUTO

Description	The gards_b_energy_pairs_orig table contains original beta energy calibration pairs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	decay_mode	*char(1)*	Decay type	Y
PI	cal_energy	*number*	Beta energy	N
	cal_error	*number*	Uncertainty of the centroid channel	Y
	channel	*number*	Peak centroid channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_ENERGY_PAIRS_ORIG(sample_id, cal_energy, type)
Foreign keys	FK_B_ENERGY_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_B_ENRGY_PRS_ORG_I_U_D, B_ENERGY_PAIRS_ORIG_B_I_U

5.1.33. GARDS_B_ENERGY_PAIRS_ORIG

Owner(s)	RMSMAN
Description	The gards_b_energy_pairs_orig table contains original beta energy calibration pairs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	decay_mode	*char(1)*	Decay type	Y
PI	cal_energy	*number*	Beta energy	N
	cal_error	*number*	Uncertainty of the centroid channel	Y
	channel	*number*	Peak centroid channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_ENERGY_PAIRS_ORIG(sample_id, cal_energy, type)
Foreign keys	FK_B_ENERGY_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	B_ENERGY_PAIRS_ORIG_B_I_U

5.1.34. GARDS_B_RESOLUTION_CAL

Owner(s)	RMSAUTO
Description	The gards_b_resolution_cal table contains beta resolution calibration coefficients calculated during analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	type	*char(8)*	Calibration type	N
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_RESOLUTION_CAL(sample_id, type)
Foreign keys	FK_GARDS_B_RESOLUTION_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_B_RESOLUTION_CAL_B_I_U, GARDS_B_RESOLUTION_CAL_I_U_D

5.1.35. GARDS_B_RESOLUTION_CAL

Owner(s)	RMSMAN
Description	The gards_b_resolution_cal table contains beta resolution calibration coefficients calculated during analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	type	*char(8)*	Calibration type	N
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y

	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_RESOLUTION_CAL(sample_id, type)
Foreign keys	FK_GARDS_B_RESOLUTION_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_B_RESOLUTION_CAL_B_I_U

5.1.36. GARDS_B_RESOLUTION_PAIRS

Owner(s)	RMSMAN
Description	The gards_b_resolution_pairs table contains beta resolution calibration pairs.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	resolution	*number*	Beta resolution (FWHM)	Y
PI	res_energy	*number*	Beta energy	N
	res_error	*number*	Uncertainty of the resolution	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_RESOLUTION_PAIRS(sample_id, res_energy, type)
Foreign keys	FK_B_RESOLUTION_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	B_RESOLUTION_PAIRS_B_I_U

5.1.37. GARDS_B_RESOLUTION_PAIRS

Owner(s)	RMSAUTO
Description	The gards_b_resolution_pairs table contains beta resolution calibration pairs.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	resolution	*number*	Beta resolution (FWHM)	Y
PI	res_energy	*number*	Beta energy	N
	res_error	*number*	Uncertainty of the resolution	Y
	moddate	*date*	Date/time at which the row was last modi-	Y

			fied	
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_RESOLUTION_PAIRS(sample_id, res_energy, type)
Foreign keys	FK_B_RESOLUTION_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	B_RESOLUTION_PAIRS_B_I_U, GARDS_B_RES_PAIRS_I_U_D

5.1.38. GARDS_B_RESOLUTION_PAIRS_ORIG

Owner(s)	RMSMAN
Description	The gards_b_resolution_pairs_orig table contains original beta resolution calibration pairs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	resolution	*number*	Beta resolution (FWHM)	Y
PI	res_energy	*number*	Beta energy	N
	res_error	*number*	Uncertainty of the resolution	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_RESOLUTION_PAIRS_O(sample_id, res_energy, type)
Foreign keys	FK_B_RESOLUTION_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	B_RESOLUTION_PAIRS_ORIG_B_I_U

5.1.39. GARDS_B_RESOLUTION_PAIRS_ORIG

Owner(s)	RMSAUTO
Description	The gards_b_resolution_pairs_orig table contains original beta resolution calibration pairs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	resolution	*number*	Beta resolution (FWHM)	Y
PI	res_energy	*number*	Beta energy	N
	res_error	*number*	Uncertainty of the resolution	Y
	moddate	*date*	Date/time at which the row was last modified	Y

PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_B_RESOLUTION_PAIRS_O(sample_id, res_energy, type)
Foreign keys	FK_B_RESOLUTION_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_B_RES_PRS_ORIG_I_U_D, B_RESOLUTION_PAIRS_ORIG_B_I_U

5.1.40. GARDS_CAT_CRITERIA_TESTS

Owner(s)	RMSMAN
Description	The gards_cat_criteria_tests table contains an optional list of category criteria tests. The column active_flag is set to 0 if the test should not be run. A trigger updates the moddate column when any modification is made to another column.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	test_id	*number*	Unique identifier	N
I	test_code	*varchar2(24)*	Unique test code identifier string	Y
	test_name	*varchar2(48)*	Test name label for display	Y
	lower_limit	*number*	Lower limit bound of the test	Y
	upper_limit	*number*	Upper limit bound of the test	Y
	units	*varchar2(16)*	Units of test bounds	Y
	active_flag	*number*	0 if test is inactive	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_CAT_CRIT_TEST_ID(test_id)
Triggers	CAT_CRITERIA_B_I_U

5.1.41. GARDS_CAT_TEMPLATE

Owner(s)	RMSMAN
Description	The gards_cat_template table contains categorization initialization information for typical nuclides. (Note: It is obsoleted.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	station_id	*number*	Station identifier	N
PFI	detector_id	*number*	Detector identifier	N
PI	name	*varchar2(8)*	Nuclide name	N
	method_id	*number*	Method identifier	Y
	method_type	*number*	Method type	Y
	init_begin_date	*date*	Initialization date	Y

	init_end_date	*date*	Decommissioning date	Y
PI	begin_date	*date*	Initialization date	N
	end_date	*date*	Decommissioning date	Y
	analyst	*varchar2(30)*	Name of the analyst who generated the statistics	Y
	comment_text	*varchar2(256)*	Comment on the filter	Y
	upper_bound	*number*	Upper limit of the amount of a nuclide that can be acceptably found	Y
	lower_bound	*number*	Lower limit of the amount of a nuclide that can be acceptably found	Y
	central_value	*number*	Current estimate of the level of the random process	Y
	delta	*number*	Value of a variable used in bounds estimation (EWMA algorithm)	Y
	abscissa	*number*	Acceptable number of standard deviations	Y
	dev_tolerance	*number*	Tolerance	Y
	num_samples	*number*	Number of samples used to collect the average and define the standard deviation (RDC algorithm)	Y
	alpha	*number*	Value of a variable used in bounds estimation	Y
	gamma	*number*	Value of a variable used in bounds estimation	Y
	tstat	*number*	Value of a variable used in bounds estimation	Y
	xform	*number*	Code for transform function applied to measurement data	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_CAT_TEMPLATE(station_id, detector_id, name, begin_date)
Foreign keys	FK_DETECTOR(detector_id, GARDS_DETECTORS), FK_STATION(station_id, GARDS_STATIONS)
Triggers	CAT_TEMPLATE_B_I_U

5.1.42. GARDS_CODES

Owner(s)	RMSMAN
Description	The gards_codes table contains codes for items used in radionuclide monitoring system.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	type	*varchar2(15)*	Category of code	N

PI	code	*char(5)*	Code symbol	N
	description	*varchar2(40)*	Code explanation	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_CODES(type, code)
Indexes	PK_GARDS_CODES(type, code)
Triggers	CODES_B_I_U

5.1.43. GARDS_COMMENTS

Owner(s)	RMSAUTO
Description	The gards_comments table contains automated analysis and interactive review comments related to peaks in the gards_peaks table and nuclides in tables of gards_nucl_ided and/or gards_nucl_lines_ided.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	sample_id	*number*	Sample identifier	Y
	peak_id	*number*	Peak identifier	Y
	nucl_name	*varchar2(8)*	Nuclide name	Y
PI	comment_id	*number*	Comment identifier	N
I	analyst	*varchar2(30)*	Name of the analyst who reviewed the sample	Y
	dtg	*date*	Entry date and time	Y
	comment_type	*number*	Comment code	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_COMMENTS(comment_id)
Foreign keys	FK_COMMENTS(sample_id, GARDS_SAMPLE_DATA)
Slaves	GARDS_USER_COMMENTS
Triggers	GARDS_COMMENTS_I_U_D, COMMENTS_B_I_U

5.1.44. GARDS_COMMENTS

Owner(s)	RMSMAN
Description	The gards_comments table contains automated analysis and interactive review comments related to peaks in the gards_peaks table and nuclides in tables of gards_nucl_ided and/or gards_nucl_lines_ided.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	sample_id	*number*	Sample identifier	Y
	peak_id	*number*	Peak identifier	Y

	nucl_name	*varchar2(8)*	Nuclide name	Y
PI	comment_id	*number*	Comment identifier	N
I	analyst	*varchar2(30)*	Name of the analyst who reviewed the sample	Y
	dtg	*date*	Entry date and time	Y
	comment_type	*number*	Comment code	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_COMMENTS(comment_id)
Foreign keys	FK_COMMENTS(sample_id, GARDS_SAMPLE_DATA)
Slaves	GARDS_USER_COMMENTS
Triggers	COMMENTS_B_I_U

5.1.45. GARDS_COMMENTS_DEFS

Owner(s)	RMSMAN
Description	The gards_comments_defs table contains predefined comments for automated analysis and interactive review.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	comment_type	*number*	Comment code	N
	comment_text	*varchar2(256)*	Comment text	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	comment_category	*varchar2(10)*	Comment category	Y

Primary key	PK_COMMENTS_DEF(comment_type)
Triggers	COMMENTS_DEFS_B_I_U

5.1.46. GARDS_CSC_MODCOEFF_LIB

Owner(s)	RMSMAN
Description	The gards_csc_modcoeff_lib table contains the cascade summing ratios. (Note: It is obsoleted.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	detector_id	*number*	Detector identifier	N
PI	name	*varchar2(8)*	Nuclide name	N
	name_d	*varchar2(8)*	Daughter nuclide name	Y
PI	energy	*number*	Energy of the decay line	N
	abundance	*number*	Emission intensity per disintegration	Y

	app_abundance	*number*	Apparent abundance	Y
	app_abundance_err	*number*	Absolute error in apparent abundance	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_CSC_MODCOEFF(detector_id, name, energy)
Foreign keys	FK_CSC_MODCFF_LIB(detector_id, GARDS_DETECTORS)
Triggers	CSC_MODCOEFF_LIB_B_I_U

5.1.47. GARDS_DATA_LOG

Owner(s)	RMSAUTO
Description	The gards_data_log table contains data describing radionuclide data messages received.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	station_id	*number*	Station identifier	Y
I	data_type	*varchar2(10)*	Type of the data message	Y
	date_received	*date*	Date that sample was received	Y
I	dtg	*date*	Transmit date and time	Y
I	msg_id	*number*	Message identifier	Y
I	status	*char(1)*	Status	Y
	file_size	*number*	Size of data file in bytes	Y
	auth	*number*	Authentication status of the associated message	Y
I	rms_id	*number*	Internal identifier for message	Y
PI	dlid	*number*	Unique data log identifier	N
I	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_DATA_LOG(dlid)
Triggers	DATA_LOG_B_I_U

5.1.48. GARDS_DBROLE_OWNER

Owner(s)	RMSMAN
Description	The gards_dbrole_owner table contains information to determine the database user that owns the RMS roles.

Pk/Fk/I	Column	Storage type	Description	Nullable
	owner	*varchar2(12)*	Name of database user who owns the RMS roles for the MAR tool	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Triggers	DBROLE_OWNER_B_I_U

5.1.49. GARDS_DETECTORS

Owner(s)	RMSMAN
Description	The gards_detectors table contains detector characteristics.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	detector_id	*number*	Detector identifier	N
I	detector_code	*varchar2(9)*	Detector code	Y
	description	*varchar2(80)*	Detector description	Y
	lat	*number*	Latitude (degrees)	Y
	lon	*number*	Longitude (degrees)	Y
	type	*varchar2(6)*	Detector type	Y
	channels	*number*	Number of channels in spectrum	Y
	rated_efficiency	*number*	Rated efficiency of detector	Y
	rated_resolution	*number*	Rated resolution of detector	Y
	ecal_range_max	*number*	Maximum detector calibration energy (keV)	Y
	date_begin	*date*	Initialization date	Y
	date_end	*date*	Decommissioning date	Y
	status	*varchar2(2)*	Status of detector, if Y then data from this detector are automatically processed if N then they are not automatically processed	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_DETECTORS(detector_id)
Slaves	GARDS_PROC_PARAMS_TEMPLATE, GARDS_BG_CONFIG_PARAMS, GARDS_UPDATE_REFLINES, GARDS_UPDATE_PARAMS_TEMPLATE, GARDS_CAT_TEMPLATE, GARDS_EFFICIENCY_VGSL_PAIRS, GARDS_QC-PARAMS, GARDS_QCTARGETS, GARDS_STADET, GARDS_IRF, GARDS_BG_ENERGY_CAL_DIRECTIVE, GARDS_CSC_MODCOEFF_LIB, GARDS_XE_PROC_PARAMS_TEMPLATE
Triggers	DETECTORS_B_I_U

5.1.50. GARDS_DIST_SAMPLE_QUEUE

Owner(s)	RMSMAN

Description	The gards_dist_sample_queue table contains a list of sample id that are assigned to users.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	user_id	*number*	User identifier	Y
	role_id	*number*	Role identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_DIST_SAMPLE_QUEUE(sample_id)
Foreign keys	FK_DIST_SAMPLE_QUEUE(sample_id, GARDS_SAMPLE_DATA)
Triggers	DIST_SAMPLE_QUEUE_B_I_U

5.1.51. GARDS_EFFICIENCY_CAL

Owner(s)	RMSMAN
Description	The gards_efficiency_cal table contains fitting coefficients of spectrum specific gamma efficiency calibration in log-log scale.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	degree	*number*	Polynomial degree of the efficiency equation	Y
	efftype	*char(8)*	Efficiency type	Y
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_EFFICIENCY_CAL(sample_id)
Foreign keys	FK_EFFICIENCY_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	EFFICIENCY_CAL_B_I_U

5.1.52. GARDS_EFFICIENCY_CAL

Owner(s)	RMSAUTO

Description	The gards_efficiency_cal table contains fitting coefficients of spectrum specific gamma efficiency calibration in log-log scale.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
	degree	*number*	Polynomial degree of the efficiency equation	Y
	efftype	*char(8)*	Efficiency type	Y
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_EFFICIENCY_CAL(sample_id)
Foreign keys	FK_EFFICIENCY_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_EFFICIENCY_CAL_I_U_D, EFFICIENCY_CAL_B_I_U

5.1.53. GARDS_EFFICIENCY_ERROR_CAL

Owner(s)	RMSAUTO
Description	The gards_efficiency_error_cal table contains gamma efficiency errors that are computed for the spectrum in case of the EMP configuration.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
	degree	*number*	Polynomial degree of the efficiency equation	Y
PI	type	*char(8)*	Uncertainty type	N
	coeff1	*number*	Uncertainty of the zero order coefficient	Y
	coeff2	*number*	Uncertainty of the first order coefficient	Y
	coeff3	*number*	Uncertainty of the second order coefficient	Y
	coeff4	*number*	Uncertainty of the third order coefficient	Y
	coeff5	*number*	Uncertainty of the fourth order coefficient	Y
	coeff6	*number*	Uncertainty of the fifth order coefficient	Y
	coeff7	*number*	Uncertainty of the sixth order coefficient	Y
	coeff8	*number*	Uncertainty of the seventh order coefficient	Y

	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_EFFICIENCY_ERROR_CAL(sample_id, type)
Foreign keys	FK_EFFICIENCY_ERROR_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_EFFICIENCY_ERR_CAL_I_U_D, EFFICIENCY_ERR_CAL_B_I_U

5.1.54. GARDS_EFFICIENCY_ERROR_CAL

Owner(s)	RMSMAN
Description	The gards_efficiency_error_cal table contains gamma efficiency errors that are computed for the spectrum in case of the EMP configuration.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	degree	*number*	Polynomial degree of the efficiency equation	Y
PI	type	*char(8)*	Uncertainty type	N
	coeff1	*number*	Uncertainty of the zero order coefficient	Y
	coeff2	*number*	Uncertainty of the first order coefficient	Y
	coeff3	*number*	Uncertainty of the second order coefficient	Y
	coeff4	*number*	Uncertainty of the third order coefficient	Y
	coeff5	*number*	Uncertainty of the fourth order coefficient	Y
	coeff6	*number*	Uncertainty of the fifth order coefficient	Y
	coeff7	*number*	Uncertainty of the sixth order coefficient	Y
	coeff8	*number*	Uncertainty of the seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_EFFICIENCY_ERROR_CAL(sample_id, type)
Foreign keys	FK_EFFICIENCY_ERROR_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	EFFICIENCY_ERROR_CAL_B_I_U

5.1.55. GARDS_EFFICIENCY_IDC_PAIRS

Owner(s)	RMSAUTO
Description	The gards_efficiency_idc_pairs table contains gamma efficiency pairs calculated by the IDC analysis software.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	detector_id	*number*	Detector identifier	N
PI	station_id	*number*	Station identifier	N
PI	effic_energy	*number*	Gamma energy	N

	efficiency	*number*	Efficiency	N
	effic_error	*number*	Uncertainty of the efficiency	N
PI	begin_date	*date*	Initialization date	N
	end_date	*date*	Decommissioning date	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	sample_id	*number*	Sample identifier	Y

Primary key	PK_GARDS_EFF_IDC_PAIRS(detector_id, station_id, begin_date, effic_energy)
Triggers	EFFICIENCY_IDC_PAIRS_B_I_U, EFFICIENCY_IDC_PAIRS_I_U_D

5.1.56. GARDS_EFFICIENCY_IDC_PAIRS

Owner(s)	RMSMAN
Description	The gards_efficiency_idc_pairs table contains gamma efficiency pairs calculated by the IDC analysis software.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	detector_id	*number*	Detector identifier	N
PI	station_id	*number*	Station identifier	N
PI	effic_energy	*number*	Gamma energy	N
	efficiency	*number*	Efficiency	N
	effic_error	*number*	Uncertainty of the efficiency	N
PI	begin_date	*date*	Initialization date	N
	end_date	*date*	Decommissioning date	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	sample_id	*number*	Sample identifier	Y

Primary key	PK_GARDS_EFF_IDC_PAIRS(detector_id, station_id, begin_date, effic_energy)
Triggers	EFFICIENCY_IDC_PAIRS_B_I_U

5.1.57. GARDS_EFFICIENCY_PAIRS

Owner(s)	RMSAUTO
Description	The gards_efficiency_pairs table contains gamma efficiency pairs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	effic_energy	*number*	Gamma energy	N
	efficiency	*number*	Efficiency	Y

	effic_error	*number*	Uncertainty of the efficiency	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_EFFICIENCY_PAIRS(sample_id, effic_energy)
Foreign keys	FK_EFFICIENCY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	EFFICIENCY_PAIRS_B_I_U, GARDS_EFFICIENCY_PAIRS_I_U_D

5.1.58. GARDS_EFFICIENCY_PAIRS

Owner(s)	RMSMAN
Description	The gards_efficiency_pairs table contains gamma efficiency pairs as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	effic_energy	*number*	Gamma energy	N
I	efficiency	*number*	Efficiency	Y
	effic_error	*number*	Uncertainty of the efficiency	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_EFFICIENCY_PAIRS(sample_id, effic_energy)
Foreign keys	FK_EFFICIENCY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	EFFICIENCY_PAIRS_B_I_U

5.1.59. GARDS_EFFICIENCY_VGSL_PAIRS

Owner(s)	RMSMAN
Description	The GARDS_EFFICIENCY_VGSL_PAIRS table contains detector specific efficiency pairs simulated by Monte-Carlo method

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	detector_id	*number*	Detector identifier	N
PI	effic_energy	*number*	Gamma energy	N
	efficiency	*number*	Efficiency	Y
	effic_error	*number*	Uncertainty of the efficiency	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	begin_date	*date*	Initialization date	N
PI	end_date	*date*	Decommissioning date	N

Primary key	PK_GARDS_EFF_VGSL_PAIRS(detector_id, begin_date, end_date, effic_energy)
Foreign keys	FK_EFFICIENCY_VGSL_PAIRS(detector_id, GARDS_DETECTORS)
Triggers	EFF_VGSL_PAIRS_B_I_U

5.1.60. GARDS_ENERGY_CAL

Owner(s)	RMSAUTO
Description	The gards_energy_cal table contains energy calibration coefficients associated with the spectrum analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	energy_units	*char(3)*	Energy units	Y
	cnv_factor	*number*	Conversion to keV	Y
	ape	*number*	Average prediction error	Y
	det	*number*	Determinant of the weighted design matrix, a measure of the spread of fitted data points and the error in the fitted points	Y
	mse	*number*	Mean-squared error of the fit, a measure of how large the residuals are	Y
	tstat	*number*	Percentage point of the t-statistic for Type I error with a probability set to 99%	Y
	score	*number*	Score used when comparing this ECR to other ECRs, a measure of both the ape and determinant	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_ENERGY_CAL(sample_id, type)
Foreign keys	FK_ENERGY_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_CAL_B_I_U, GARDS_ENERGY_CAL_I_U_D

5.1.61. GARDS_ENERGY_CAL

Owner(s)	RMSMAN
Description	The gards_energy_cal table contains energy calibration coefficients associated with the spectrum analysis.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	energy_units	*char(3)*	Energy units	Y
	cnv_factor	*number*	Conversion to keV	Y
	ape	*number*	Average prediction error	Y
	det	*number*	Determinant of the weighted design matrix, a measure of the spread of fitted data points and the error in the fitted points	Y
	mse	*number*	Mean-squared error of the fit, a measure of how large the residuals are	Y
	tstat	*number*	Percentage point of the t-statistic for Type I error with a probability set to 99%	Y
	score	*number*	Score used when comparing this ECR to other ECRs, a measure of both the ape and determinant	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_ENERGY_CAL(sample_id, type)
Foreign keys	FK_ENERGY_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_CAL_B_I_U

5.1.62. GARDS_ENERGY_CAL_COV

Owner(s)	RMSMAN
Description	The GARDS_ENERGY_CAL_COV table contains the covariance table for each

	coefficient of the energy calibration for a given sample_id

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(22,0)*	Sample identifier	N
PI	row_index	*number(10,0)*	Row index	N
PI	col_index	*number(10,0)*	Column index	N
	moddate	*date*	Date/time at which the row was last modified	Y
	coeff	*float(126)*	Calibration coefficient	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_ENERGY_CAL_COV(sample_id, type, row_index, col_index)
Foreign keys	FK_ENERGY_CAL_COV(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_CAL_COV_B_I_U

5.1.63. GARDS_ENERGY_CAL_COV

Owner(s)	RMSAUTO
Description	The GARDS_ENERGY_CAL_COV table contains the covariance table for each coefficient of the energy calibration for a given sample_id

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(22,0)*	Sample identifier	N
PI	row_index	*number(10,0)*	Row index	N
PI	col_index	*number(10,0)*	Column index	N
	moddate	*date*	Date/time at which the row was last modified	Y
	coeff	*float(126)*	Calibration coefficient	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_ENERGY_CAL_COV(sample_id, type, row_index, col_index)
Foreign keys	FK_ENERGY_CAL_COV(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_CAL_COV_B_I_U, GARDS_ENERGY_CAL_COV_I_U_D

5.1.64. GARDS_ENERGY_CAL_ORIG

Owner(s)	RMSAUTO
Description	The gards_energy_cal_orig table contains original energy calibration associated with the spectrum (calculated using the channel-energy pair data).

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	energy_units	*char(3)*	Energy units	Y
	cnv_factor	*number*	Conversion to keV	Y
	ape	*number*	Average prediction error	Y
	det	*number*	Determinant of the weighted design matrix, a measure of the spread of fitted data points and the error in the fitted points	Y
	mse	*number*	Mean-squared error of the fit, a measure of how large the residuals are	Y
	tstat	*number*	Percentage point of the t-statistic for Type I error with a probability set to 99%	Y
	score	*number*	Score used when comparing this ECR to other ECRs, a measure of both the ape and determinant	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_ENERGY_CAL_ORIG(sample_id, type)
Foreign keys	FK_ENERGY_CAL_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_ENERGY_CAL_ORIG_I_U_D, ENERGY_CAL_ORIG_B_I_U

5.1.65. GARDS_ENERGY_CAL_ORIG

Owner(s)	RMSMAN
Description	The gards_energy_cal_orig table contains original energy calibration associated with the spectrum (calculated using the channel-energy pair data).

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N

	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	energy_units	*char(3)*	Energy units	Y
	cnv_factor	*number*	Conversion to keV	Y
	ape	*number*	Average prediction error	Y
	det	*number*	Determinant of the weighted design matrix, a measure of the spread of fitted data points and the error in the fitted points	Y
	mse	*number*	Mean-squared error of the fit, a measure of how large the residuals are	Y
	tstat	*number*	Percentage point of the t-statistic for Type I error with a probability set to 99%	Y
	score	*number*	Score used when comparing this ECR to other ECRs, a measure of both the ape and determinant	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_ENERGY_CAL_ORIG(sample_id, type)
Foreign keys	FK_ENERGY_CAL_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_CAL_ORIG_B_I_U

5.1.66. GARDS_ENERGY_PAIRS

Owner(s)	RMSMAN
Description	The gards_energy_pairs table contains gamma energy calibration pairs.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	cal_energy	*number*	Gamma energy	N
	channel	*number*	Peak centroid channel	Y
	cal_error	*number*	Uncertainty of the centroid channel	Y
	moddate	*date*	Date/time at which the row was last modi-	Y

			fied	
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_ENERGY_PAIRS(sample_id, cal_energy, type)
Foreign keys	FK_ENERGY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_PAIRS_B_I_U

5.1.67. GARDS_ENERGY_PAIRS

Owner(s)	RMSAUTO
Description	The gards_energy_pairs table contains gamma energy calibration pairs.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	cal_energy	*number*	Gamma energy	N
	channel	*number*	Peak centroid channel	Y
	cal_error	*number*	Uncertainty of the centroid channel	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_ENERGY_PAIRS(sample_id, cal_energy, type)
Foreign keys	FK_ENERGY_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_ENERGY_PAIRS_I_U_D, ENERGY_PAIRS_B_I_U

5.1.68. GARDS_ENERGY_PAIRS_ORIG

Owner(s)	RMSAUTO
Description	The gards_energy_pairs_orig table contains original gamma energy calibration pairs specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	cal_energy	*number*	Gamma energy	N
	channel	*number*	Peak centroid channel	Y
	cal_error	*number*	Uncertainty of the centroid channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y

	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_ENERGY_PAIRS_ORIG(sample_id, cal_energy, type)
Foreign keys	FK_ENERGY_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_PAIRS_ORIG_B_I_U, GARDS_ENERGY_PAIRS_ORIG_I_U_D

5.1.69. GARDS_ENERGY_PAIRS_ORIG

Owner(s)	RMSMAN
Description	The gards_energy_pairs_orig table contains original gamma energy calibration pairs specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	cal_energy	*number*	Gamma energy	N
	channel	*number*	Peak centroid channel	Y
	cal_error	*number*	Uncertainty of the centroid channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_ENERGY_PAIRS_ORIG(sample_id, cal_energy, type)
Foreign keys	FK_ENERGY_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENERGY_PAIRS_ORIG_B_I_U

5.1.70. GARDS_ENVIRONMENT

Owner(s)	RMSAUTO
Description	The gards_environment table contains atmospheric conditions and related sample information. This table is applicable to PHDs sent in IMS1.0 format or earlier. It does not apply to messages in IMS2.0 format or later.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	flow_rate	*number*	Flow rate through filter (m3/h)	Y
	pressure	*number*	Outside pressure (mbar)	Y
	temperature	*number*	Outside temperature (degrees Celsius)	Y
	end_env_time	*date*	End date of environment data	Y
	end_env_time_se	*number*	End time of environment data (seconds)	Y

	c			
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ENVIRONMENT(sample_id)
Foreign keys	FK_ENVIRONMENT(sample_id, GARDS_SAMPLE_DATA)
Triggers	ENVIRONMENT_B_I_U

5.1.71. GARDS_FLAGS

Owner(s)	RMSAUTO
Description	The gards_flags table contains the name and thresholds for each of the tests run during rms_DBflags.

Pk/Fk/I	Column	Storage type	Description	Nullable
	flag_id	*number(8,0)*	Flag identifier	Y
	name	*varchar2(32)*	Name of the event screening test	Y
	threshold	*number*	Value against which the results in the test column are compared	Y
	units	*varchar2(16)*	Units of test bounds	Y
	test	*varchar2(8)*	Calculated result of the test	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Triggers	FLAGS_B_I_U

5.1.72. GARDS_FLAGS

Owner(s)	RMSMAN
Description	The gards_flags table contains the name and thresholds for each of the tests run during rms_DBflags.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	flag_id	*number(8,0)*	Flag identifier	N
	name	*varchar2(32)*	Name of the event screening test	Y
	threshold	*number*	Value against which the results in the test column are compared	Y
	units	*varchar2(16)*	Units of test bounds	Y
	test	*varchar2(8)*	Calculated result of the test	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	SYS_C0028368(flag_id)
Indexes	SYS_C0028368(flag_id)
Triggers	FLAGS_B_I_U

5.1.73. GARDS_FPE

Owner(s)	RMSAUTO
Description	The gards_fpe table contains one record for each sample associated with a fission product event.

Pk/Fk/I	Column	Storage type	Description	Nullable
	fpid	*number*	Fission product identifier	Y
	revid	*number*	Revision number	Y
	dtg	*date*	Entry date and time	Y
FI	sample_id	*number*	Sample identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Foreign keys	FK_FPE(sample_id, GARDS_SAMPLE_DATA)
Indexes	FPE_NDX(sample_id)
Triggers	FPE_B_I_U

5.1.74. GARDS_FPE

Owner(s)	RMSMAN
Description	The gards_fpe table contains one record for each sample associated with a fission product event.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	fpid	*number*	Fission product identifier	N
	revid	*number*	Revision number	Y
	dtg	*date*	Entry date and time	Y
FI	sample_id	*number*	Sample identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	GARDS_FPE_PK(fpid)
Foreign keys	FK_FPE(sample_id, GARDS_SAMPLE_DATA)
Indexes	PK_GARDS_FPE(fpid), FPE_NDX(sample_id)
Triggers	FPE_B_I_U

5.1.75. GARDS_HISTOGRAM

Owner(s)	RMSAUTO
Description	The gards_histogram table contains the pulse height data of the beta-gamma coincidence spectrum, e. g. the counts in each beta-gamma energy channel within a 2-D matrix format.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
	filename	*varchar2(256)*	Name of the file containing the histogram data	Y
	g_channels	*number*	Number of gamma channels in histogram	Y
	b_channels	*number*	Number of beta channels in histogram	Y
	g_energy_span	*number*	Gamma energy span of detector calibration (keV)	Y
	b_energy_span	*number*	Beta energy span of detector calibration (keV)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_HISTOGRAM(sample_id)
Foreign keys	FK_HISTOGRAM(sample_id, GARDS_SAMPLE_DATA)
Triggers	HISTOGRAM_B_I_U

5.1.76. GARDS_INTERVAL

Owner(s)	RMSAUTO
Description	The gards_interval table contains data used to update RMS data workflow.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
	class	*varchar2(16)*	Station name	Y
	name	*varchar2(16)*	Detector/system name	Y
	start_time	*float(53)*	Start time of the period	Y
	end_time	*float(53)*	End time of the period	Y
	state	*varchar2(16)*	Specific activity of detector/system	Y
I	moddate	*date*	Date/time at which the row was last modified	Y
I	lddate	*date*	Load date	Y
I	intvlid	*number*	Unique interval identifier	Y

Triggers	INTERVAL_B_I_U, INTERVAL_S_I

5.1.77. GARDS_IRF

Owner(s)	RMSMAN
Description	The gards_irf table contains the isotope response function, summing corrections as a function of the energy for a given Nuclide and a given Detector, which is determined by Monte-Carlo method normally.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	detector_id	*number*	Detector identifier	N
	energy	*number*	Gamma energy	Y
	irf	*number*	Isotope response function	Y
	irf_error	*number*	Uncertainty of the isotope response function	Y
	nuclide_name	*varchar2(8)*	Nuclide name	Y
	sum_corr	*number*	Summing cascade correction factor	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	begin_date	*date*	Initialization date	Y
	end_date	*date*	Decommissioning date	Y

Foreign keys	FK_IRF(detector_id, GARDS_DETECTORS)
Indexes	IRF_NDX(detector_id)
Triggers	IRF_B_I_U

5.1.78. GARDS_LAB_CATEGORY_DESCRIPTION

Owner(s)	RMSMAN
Description	The gards_lab_category_description table describes the different sample categories that can be sent to the IDC.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	lab_sample_category	*varchar2(2)*	Report sample category	N
	description	*varchar2(256)*	Description of the lab_sample_category	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_LAB_CATEGORY_DESCRIPTION(lab_sample_category)
Triggers	LAB_CAT_DESC_B_I_U

5.1.79. GARDS_MDAS2REPORT

Owner(s)	RMSMAN
Description	The GARDS_MDAS2REPORT table contains the list of nuclides which have their minimum detectable activities reported in radionuclide reports

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
P	type	*varchar2(16)*	Nuclide type	Y
	mda_min	*number*	Minimum of the minimum detectable activities	Y
	mda_max	*number*	Maximum of the minimum detectable activities	Y
PI	sample_type	*varchar2(2)*	Sample type	N
	dtg_begin	*date*	Begin date and time	Y
	dtg_end	*date*	End date and time	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_MDAS2REPORT(name, sample_type)

5.1.80. GARDS_MET_DATA

Owner(s)	RMSAUTO
Description	The gards_met_data table contains station local meteorological data.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	station_id	*number*	Station identifier	Y
	ave_humidity	*number*	Average humidity (percent relative humidity)	Y
	ave_out_temp	*number*	Average outside temperature (deg. C)	Y
	ave_pressure	*number*	Average atmospheric pressure.	Y
	ave_wind_dir	*number*	Average wind direction (degrees from North)	Y
	ave_wind_speed	*number*	Average wind speed (km/h)	Y
	total_rainfall	*number*	Total rainfall (mm)	Y
I	start_time	*date*	Start time of the period	Y
I	end_time	*date*	End time of the period	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	met_id	*number*	Unique meteorological data identifier	N
F	met_group_id	*number*	MET group identifier	Y

Primary key	PK_GARDS_MET_DATA(met_id)
Foreign keys	FK_MET_GROUP_ID_MET_HEADER(met_group_id, GARDS_MET_HEADER), FK_MET_DATA(station_id, GARDS_STATIONS)
Triggers	MET_DATA_B_I_U, MET_DATA_S_I

5.1.81. GARDS_MET_HEADER

Owner(s)	RMSAUTO
Description	The gards_met_header table contains the header information for MET messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	met_group_id	*number*	MET group identifier	N
FI	station_id	*number*	Station identifier	Y
I	start_time	*date*	Start time of the period	Y
I	end_time	*date*	End time of the period	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_MET_HEADER(met_group_id)
Foreign keys	FK_MET_HEADER(station_id, GARDS_STATIONS)
Slaves	GARDS_MET_DATA
Triggers	MET_HEADER_B_I_U, MET_HEADER_S_I

5.1.82. GARDS_NIC

Owner(s)	RMSAUTO
Description	Obsolete according to /ops/software/rn/doc/documentation/rnpipeline/DevLAN_OG-RN.3.3.doc

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sample_id	*number*		N
	station_id	*number*		Y
PI	name	*varchar2(8)*		N
	dtg	*date*		Y
	category	*number*		Y
	central_value	*number*		Y
	del	*number*		Y
	upper_lim	*number*		Y
	lower_lim	*number*		Y
PI	sample_status	*varchar2(1)*		N
	activity	*number*		Y
	hold	*number*		Y
	moddate	*date*		Y

Primary key	PK_GARDS_NIC(sample_id, name, sample_status)
Indexes	PK_GARDS_NIC(sample_id, name, sample_status)

5.1.83. GARDS_NIC

Owner(s)	RMSMAN
Description	Obsolete according to /ops/software/rn/doc/documentation/rnpipeline/DevLAN_OG-RN.3.3.doc

Pk/Fk/I	Column	Storage type	Description	Nullable
	sample_id	*number*		Y
	station_id	*number*		Y
	name	*varchar2(8)*		Y
	dtg	*date*		Y
	category	*number*		Y
	central_value	*number*		Y
	del	*number*		Y
	upper_lim	*number*		Y
	lower_lim	*number*		Y
	sample_status	*varchar2(1)*		Y
	activity	*number*		Y
	hold	*number*		Y
	moddate	*date*		Y

5.1.84. GARDS_NIC_INIT

Owner(s)	RMSMAN
Description	Obsolete according to /ops/software/rn/doc/documentation/rnpipeline/DevLAN_OG-RN.3.3.doc

Pk/Fk/I	Column	Storage type	Description	Nullable
	station_id	*number*		Y
	name	*varchar2(8)*		Y
	analyst	*varchar2(30)*		Y
	rec_begin	*date*		Y
	rec_end	*date*		Y
	init_begin	*date*		Y
	init_end	*date*		Y
	alpha	*number*		Y
	gamma	*number*		Y
	xform	*number*		Y
	tstat	*number*		Y
	central_value	*number*		Y

	del	*number*		Y
	upper_lim	*number*		Y
	lower_lim	*number*		Y
	comments	*varchar2(256)*		Y
	hold	*number*		Y
	init_detector_id	*number*		Y

5.1.85. GARDS_NOTIFY

Owner(s)	RMSAUTO
Description	The gards_notify table contains contact information for specific events. When the event occurs a message is automatically sent to the email_addr of the specified recipient(s).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	event	*varchar2(10)*	Type of occurrence	N
PI	email_addr	*varchar2(80)*	Recipient email addresses	N
	desciption	*varchar2(80)*	General description of occurrence	Y
PI	dtg_begin	*date*	Begin date and time	N
	dtg_end	*date*	End date and time	Y
	poc_id	*number*	Point of contact identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	description	*varchar2(80)*	General description of occurrence	Y

Primary key	PK_GARDS_NOTIFY(event, email_addr, dtg_begin)
Triggers	NOTIFY_B_I_U

5.1.86. GARDS_NOTIFY

Owner(s)	RMSMAN
Description	The gards_notify table contains contact information for specific events. When the event occurs a message is automatically sent to the email_addr of the specified recipient(s).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	event	*varchar2(10)*	Type of occurrence	N
PI	email_addr	*varchar2(80)*	Recipient email addresses	N
	desciption	*varchar2(80)*	General description of occurrence	Y
PI	dtg_begin	*date*	Begin date and time	N
	dtg_end	*date*	End date and time	Y

FI	poc_id	*number*	Point of contact identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_NOTIFY(event, email_addr, dtg_begin)
Foreign keys	FK_NOTIFY_POC(poc_id, GARDS_POC)
Indexes	NOTIFY_NDX(poc_id), PK_GARDS_NOTIFY(event, email_addr, dtg_begin)
Triggers	NOTIFY_B_I_U

5.1.87. GARDS_NUCL2QUANTIFY

Owner(s)	RMSMAN
Description	The gards_nucl2quantify table contains natural radionuclides to be quantified in radionuclide reports for particulates.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
PI	dtg_begin	*date*	Begin date and time	N
	dtg_end	*date*	End date and time	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_NUCL2QUANTIFY(name, dtg_begin)
Triggers	NUCL2QUANTIFY_B_I_U

5.1.88. GARDS_NUCL_IDED

Owner(s)	RMSMAN
Description	The gards_nucl_ided table contains information regarding nuclides identified.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
I	nuclide_id	*number*	Nuclide identifier	Y
PI	name	*varchar2(8)*	Nuclide name	N
	type	*varchar2(16)*	Nuclide type	Y
	halflife	*varchar2(23)*	Half-life	Y
	ave_activ	*number*	Average activity concentration	Y
	ave_activ_err	*number*	Uncertainty of the average activity concentration	Y
	activ_key	*number*	Activity concentration of the key line	Y

	activ_key_err	*number*	Uncertainty of the activity concentration	Y
	mda	*number*	Minimum detectable activity concentration	Y
	mda_err	*number*	Uncertainty of the minimum detectable activity concentration	Y
	nid_flag	*number*	Flag of whether the nuclide was detected or not	Y
	activ_decay	*number*	Activity decay factor	Y
	activ_decay_err	*number*	Uncertainty of activity decay factor	Y
	comp_confid	*number*	Computed confidence index	Y
	report_mda	*number*	Indicator of whether MDA is to be reported	Y
	pd_mod_flag	*number*	Flag indicating whether or not the nuclide's activity was modified by the parent/daughter calculation.	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	ave_activ_acq_start	*number*	Average activity at the start of acquisition	Y
	ave_activ_err_acq_start	*number*	Uncertainty of the average activity at the start of acquisition	Y
	activ_key_acq_start	*number*	Activity of the key line at the start of acquisition	Y
	activ_key_err_acq_start	*number*	Uncertainty of the activity of the key line at the start of acquisition	Y
	mda_acq_start	*number*	Minimum detectable activity at the start of acquisition	Y
	mda_err_acq_start	*number*	Uncertainty of the minimum detectable activity at the start of acquisition	Y

Primary key	PK_NUCL_IDED(sample_id, name)
Foreign keys	FK_NUCL_IDED(sample_id, GARDS_SAMPLE_DATA)
Slaves	GARDS_NUCL_IDED_ORIG
Triggers	NUCL_IDED_B_I_U

5.1.89. GARDS_NUCL_IDED

Owner(s)	RMSAUTO
Description	The gards_nucl_ided table contains information regarding nuclides identified.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
I	nuclide_id	*number*	Nuclide identifier	Y
PI	name	*varchar2(8)*	Nuclide name	N
	type	*varchar2(16)*	Nuclide type	Y
	halflife	*varchar2(23)*	Half-life	Y
	ave_activ	*number*	Average activity concentration	Y
	ave_activ_err	*number*	Uncertainty of the average activity concentration	Y
	activ_key	*number*	Activity concentration of the key line	Y
	activ_key_err	*number*	Uncertainty of the activity concentration	Y
	mda	*number*	Minimum detectable activity concentration	Y
	mda_err	*number*	Uncertainty of the minimum detectable activity concentration	Y
	nid_flag	*number*	Flag of whether the nuclide was detected or not	Y
	activ_decay	*number*	Activity decay factor	Y
	activ_decay_err	*number*	Uncertainty of activity decay factor	Y
	comp_confid	*number*	Computed confidence index	Y
	report_mda	*number*	Indicator of whether MDA is to be reported	Y
	pd_mod_flag	*number*	Flag indicating whether or not the nuclide's activity was modified by the parent/daughter calculation.	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	ave_activ_acq_start	*number*	Average activity at the start of acquisition	Y
	ave_activ_err_acq_start	*number*	Uncertainty of the average activity at the start of acquisition	Y
	activ_key_acq_start	*number*	Activity of the key line at the start of acquisition	Y
	activ_key_err_acq_start	*number*	Uncertainty of the activity of the key line at the start of acquisition	Y
	mda_acq_start	*number*	Minimum detectable activity at the start of acquisition	Y

	mda_err_acq_start	*number*	Uncertainty of the minimum detectable activity at the start of acquisition	Y

Primary key	PK_NUCL_IDED(sample_id, name)
Foreign keys	FK_NUCL_IDED(sample_id, GARDS_SAMPLE_DATA)
Slaves	GARDS_NUCL_IDED_ORIG
Triggers	NUCL_IDED_B_I_U, UPDATE_NUCL_ORIG_I_U_D, GARDS_NUCL_IDED_I_U_D

5.1.90. GARDS_NUCL_IDED_ORIG

Owner(s)	RMSAUTO
Description	The gards_nucl_ided_orig table contains information regarding nuclides identified during automated analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
I	nuclide_id	*number*	Nuclide identifier	Y
PFI	name	*varchar2(8)*	Nuclide name	N
	type	*varchar2(16)*	Nuclide type	Y
	halflife	*varchar2(23)*	Half-life	Y
	ave_activ	*number*	Average activity concentration	Y
	ave_activ_err	*number*	Uncertainty of the average activity concentration	Y
	activ_key	*number*	Activity concentration of the key line	Y
	activ_key_err	*number*	Uncertainty of the activity concentration	Y
	mda	*number*	Minimum detectable activity concentration	Y
	mda_err	*number*	Uncertainty of the minimum detectable activity concentration	Y
	nid_flag	*number*	Flag of whether the nuclide was detected or not	Y
	activ_decay	*number*	Activity decay factor	Y
	activ_decay_err	*number*	Uncertainty of activity decay factor	Y
	comp_confid	*number*	Computed confidence index	Y
	report_mda	*number*	Indicator of whether MDA is to be reported	Y
	pd_mod_flag	*number*	Flag indicating whether or not the nuclide's activity was modified by the parent/daughter calculation.	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y

	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	ave_activ_acq_start	*number*	Average activity at the start of acquisition	Y
	ave_activ_err_acq_start	*number*	Uncertainty of the average activity at the start of acquisition	Y
	activ_key_acq_start	*number*	Activity of the key line at the start of acquisition	Y
	activ_key_err_acq_start	*number*	Uncertainty of the activity of the key line at the start of acquisition	Y
	mda_acq_start	*number*	Minimum detectable activity at the start of acquisition	Y
	mda_err_acq_start	*number*	Uncertainty of the minimum detectable activity at the start of acquisition	Y

Primary key	PK_NUCL_IDED_ORIG(sample_id, name)
Foreign keys	FK_SID_NUCLIDEID(sample_id, GARDS_NUCL_IDED), FK_SID_NUCLIDEID(name, GARDS_NUCL_IDED), FK_NUCL_IDED_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	NUCL_IDED_ORIG_B_I_U, GARDS_NUCL_IDED_ORIG_I_U_D

5.1.91. GARDS_NUCL_IDED_ORIG

Owner(s)	RMSMAN
Description	The gards_nucl_ided_orig table contains information regarding nuclides identified during automated analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
I	nuclide_id	*number*	Nuclide identifier	Y
PI	name	*varchar2(8)*	Nuclide name	N
	type	*varchar2(16)*	Nuclide type	Y
	halflife	*varchar2(23)*	Half-life	Y
	ave_activ	*number*	Average activity concentration	Y
	ave_activ_err	*number*	Uncertainty of the average activity concentration	Y
	activ_key	*number*	Activity concentration of the key line	Y
	activ_key_err	*number*	Uncertainty of the activity concentration	Y

	mda	*number*	Minimum detectable activity concentration	Y
	mda_err	*number*	Uncertainty of the minimum detectable activity concentration	Y
	nid_flag	*number*	Flag of whether the nuclide was detected or not	Y
	activ_decay	*number*	Activity decay factor	Y
	activ_decay_err	*number*	Uncertainty of activity decay factor	Y
	comp_confid	*number*	Computed confidence index	Y
	report_mda	*number*	Indicator of whether MDA is to be reported	Y
	pd_mod_flag	*number*	Flag indicating whether or not the nuclide's activity was modified by the parent/daughter calculation.	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	ave_activ_acq_start	*number*	Average activity at the start of acquisition	Y
	ave_activ_err_acq_start	*number*	Uncertainty of the average activity at the start of acquisition	Y
	activ_key_acq_start	*number*	Activity of the key line at the start of acquisition	Y
	activ_key_err_acq_start	*number*	Uncertainty of the activity of the key line at the start of acquisition	Y
	mda_acq_start	*number*	Minimum detectable activity at the start of acquisition	Y
	mda_err_acq_start	*number*	Uncertainty of the minimum detectable activity at the start of acquisition	Y

Primary key	PK_NUCL_IDED_ORIG(sample_id, name)
Foreign keys	FK_NUCL_IDED_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	NUCL_IDED_ORIG_B_I_U

5.1.92. GARDS_NUCL_LIB

Owner(s)	RMSMAN
Description	The gards_nucl_lib table contains the nuclide library used in the analysis of particulate spectra.

Pk/Fk/I	Column	Storage type	Description	Nullable

	nuclide_id	*number*	Nuclide identifier	Y
PI	name	*varchar2(8)*	Nuclide name	N
	type	*varchar2(16)*	Nuclide type	Y
	halflife_sec	*number*	Half-life in seconds	Y
	halflife	*varchar2(23)*	Half-life	Y
	halflife_err	*varchar2(23)*	Uncertainty of half-life	Y
	num_lines	*number*	Number of decay lines in library	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	nuc_comment	*varchar2(1500)*	Nuclide comment	Y

Primary key	PK_GARDS_NUCL_LIB(name)
Triggers	NUCL_LIB_B_I_U

5.1.93. GARDS_NUCL_LIB_ARCHIVE

Owner(s)	RMSMAN
Description	The GARDS_NUCL_LIB_ARCHIVE contains the nuclide library archived.

Pk/Fk/I	Column	Storage type	Description	Nullable
	nuclide_id	*number*	Nuclide identifier	Y
PI	name	*varchar2(8)*	Nuclide name	N
	type	*varchar2(16)*	Nuclide type	Y
	halflife_sec	*number*	Half-life in seconds	Y
	halflife	*varchar2(23)*	Half-life	Y
	halflife_err	*varchar2(23)*	Uncertainty of half-life	Y
	num_lines	*number*	Number of decay lines in library	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	nuc_comment	*varchar2(1500)*	Nuclide comment	Y
PI	begin_date	*date*	Initialization date	N
	end_date	*date*	Decommissioning date	Y

Primary key	PK_ARCH_NUCL_LIB(name, begin_date)

5.1.94. GARDS_NUCL_LINES_IDED

Owner(s)	RMSAUTO
Description	The gards_nucl_lines_ided table contains information regarding energy lines identified.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	Y
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
I	name	*varchar2(8)*	Nuclide name	Y
	energy	*number*	Energy of the peak	Y
	energy_err	*number*	Uncertainty of the energy	Y
	abundance	*number*	Emission intensity per disintegration	Y
	abundance_err	*number*	Uncertainty of the abundance	Y
PI	peak	*number*	Peak identifier	Y
	activity	*number*	Activity concentration	Y
	activ_err	*number*	Uncertainty of the activity concentration	Y
	effic	*number*	Efficiency at the energy	Y
	effic_err	*number*	Uncertainty of the efficiency	Y
	mda	*number*	Minimum detectable activity concentration	Y
	key_flag	*number*	Key line indicator	Y
PI	nuclide_id	*number*	Nuclide identifier	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	id_percent	*number*	Nuclide percentage	Y

Primary key	PK_GARDS_NUCL_LINES_IDED(sample_id, nuclide_id, peak)
Foreign keys	FK_NUCL_LINES_IDED(sample_id, GARDS_SAMPLE_DATA)
Slaves	GARDS_NUCL_LINES_IDED_ORIG
Triggers	UPDATE_NUCL_ORIG_LINES_I_U_D, GARDS_NUCL_LINES_IDED_I_U_D, NUCL_LINES_IDED_B_I_U

5.1.95. GARDS_NUCL_LINES_IDED

Owner(s)	RMSMAN
Description	The gards_nucl_lines_ided table contains information regarding energy lines identified.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	sample_id	*number*	Sample identifier	Y

	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
I	name	*varchar2(8)*	Nuclide name	Y
I	energy	*number*	Energy of the peak	Y
	energy_err	*number*	Uncertainty of the energy	Y
	abundance	*number*	Emission intensity per disintegration	Y
	abundance_err	*number*	Uncertainty of the abundance	Y
I	peak	*number*	Peak identifier	Y
	activity	*number*	Activity concentration	Y
	activ_err	*number*	Uncertainty of the activity concentration	Y
	effic	*number*	Efficiency at the energy	Y
	effic_err	*number*	Uncertainty of the efficiency	Y
	mda	*number*	Minimum detectable activity concentration	Y
	key_flag	*number*	Key line indicator	Y
	nuclide_id	*number*	Nuclide identifier	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	id_percent	*number*	Nuclide percentage	Y

Foreign keys	FK_NUCL_LINES_IDED(sample_id, GARDS_SAMPLE_DATA)
Slaves	GARDS_NUCL_LINES_IDED_ORIG
Triggers	NUCL_LINES_IDED_B_I_U

5.1.96. GARDS_NUCL_LINES_IDED_ORIG

Owner(s)	RMSAUTO
Description	The gards_nucl_lines_ided_orig table contains information regarding energy lines identified during automated analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	Y
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
PI	name	*varchar2(8)*	Nuclide name	Y
PI	energy	*number*	Energy of the peak	Y

	energy_err	*number*	Uncertainty of the energy	Y
	abundance	*number*	Emission intensity per disintegration	Y
	abundance_err	*number*	Uncertainty of the abundance	Y
PFI	peak	*number*	Peak identifier	Y
	activity	*number*	Activity concentration	Y
	activ_err	*number*	Uncertainty of the activity concentration	Y
	effic	*number*	Efficiency	Y
	effic_err	*number*	Uncertainty of the efficiency	Y
	mda	*number*	Minimum detectable activity concentration	Y
	key_flag	*number*	Key line indicator	Y
FI	nuclide_id	*number*	Nuclide identifier	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	id_percent	*number*	Nuclide percentage	Y

Primary key	PK_GARDS_NUCL_LINES_IDED_ORIG(sample_id, name, energy, peak)
Foreign keys	FK_SID_ET_AL(nuclide_id, GARDS_NUCL_LINES_IDED), FK_NUCL_LINES_IDED_ORIG(sample_id, GARDS_SAMPLE_DATA), FK_SID_ET_AL(peak, GARDS_NUCL_LINES_IDED), FK_SID_ET_AL(sample_id, GARDS_NUCL_LINES_IDED)
Triggers	GARDS_NCL_LNS_IDD_ORG_I_U_D, NUCL_LINES_IDED_ORIG_B_I_U

5.1.97. GARDS_NUCL_LINES_IDED_ORIG

Owner(s)	RMSMAN
Description	The gards_nucl_lines_ided_orig table contains information regarding energy lines identified during automated analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
PI	name	*varchar2(8)*	Nuclide name	N
PI	energy	*number*	Energy of the peak	N
	energy_err	*number*	Uncertainty of the energy	Y
	abundance	*number*	Emission intensity per disintegration	Y

	abundance_err	*number*	Uncertainty of the abundance	Y
PI	peak	*number*	Peak identifier	N
	activity	*number*	Activity concentration	Y
	activ_err	*number*	Uncertainty of the activity concentration	Y
	effic	*number*	Efficiency	Y
	effic_err	*number*	Uncertainty of the efficiency	Y
	mda	*number*	Minimum detectable activity concentration	Y
	key_flag	*number*	Key line indicator	Y
	nuclide_id	*number*	Nuclide identifier	Y
	csc_ratio	*number*	Multiplier of the cascade summing correction	Y
	csc_ratio_err	*number*	Uncertainty in csc_ratio	Y
	csc_mod_flag	*number*	Flag indicating if the cascade summing correction is performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	id_percent	*number*	Nuclide percentage	Y

Primary key	PK_GARDS_NUCL_LINES_IDED_ORIG(sample_id, name, energy, peak)
Foreign keys	FK_NUCL_LINES_IDED_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	NUCL_LINES_IDED_ORIG_B_I_U

5.1.98. GARDS_NUCL_LINES_LIB

Owner(s)	RMSMAN
Description	The gards_nucl_lines_lib table contains the nuclide library regarding energy lines.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
PI	energy	*number*	Energy of the decay line	N
	energy_err	*number*	Uncertainty of the energy	Y
	abundance	*number*	Emission intensity per disintegration	Y
	abundance_err	*number*	Uncertainty of the abundance	Y
	key_flag	*number*	Key line indicator	Y
	nuclide_id	*number*	Nuclide identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	linecomment	*varchar2(1500)*	Comment associated with the decay line	Y

Primary key	PK_GARDS_NUCL_LINES_LIB(name, energy)
Triggers	NUCL_LINES_LIB_B_I_U

5.1.99. GARDS_NUCL_LINES_LIB_ARCHIVE

Owner(s)	RMSMAN
Description	The GARDS_NUCL_LINES_LIB_ARCHIVE contains the archived nuclide library regarding energy lines.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
PI	energy	*number*	Energy of the decay line	N
	energy_err	*number*	Uncertainty of the energy	Y
	abundance	*number*	Emission intensity per disintegration	Y
	abundance_err	*number*	Uncertainty of the abundance	Y
	key_flag	*number*	Key line indicator	Y
	nuclide_id	*number*	Nuclide identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	linecomment	*varchar2(1500)*	Comment associated with the decay line	Y
PI	begin_date	*date*	Initialization date	N
	end_date	*date*	Decommissioning date	Y

Primary key	PK_ARCH_NUCL_LINES_LIB(name, energy, begin_date)

5.1.100. GARDS_PEAKS

Owner(s)	RMSAUTO
Description	The gards_peaks table contains information regarding peaks identified.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	peak_id	*number*	Peak identifier	N
	centroid	*number*	Peak centroid	Y
	centroid_err	*number*	Uncertainty of the peak centroid.	Y
	energy	*number*	Energy of the peak	Y
	energy_err	*number*	Uncertainty of the energy	Y
	left_chan	*number*	Left channel of the peak	Y
	width	*number*	Width of the peak	Y
	back_count	*number*	Continuum background counts	Y

	back_uncer	*number*	Uncertainty of the continuum background counts	Y
	fwhm	*number*	Full width at half maximum	Y
	fwhm_err	*number*	Uncertainty of full width at half maximum	Y
	area	*number*	Peak area (counts)	Y
	area_err	*number*	Uncertainty of the peak area (counts)	Y
	original_area	*number*	Original peak area (counts)	Y
	original_uncer	*number*	Uncertainty of the original peak area (counts)	Y
	counts_sec	*number*	Counts per second	Y
	counts_sec_err	*number*	Uncertainty of the counts per second	Y
	efficiency	*number*	Efficiency	Y
	eff_error	*number*	Uncertainty of the efficiency	Y
	back_channel	*number*	Number of the average background counts per channel	Y
	ided	*number*	Peak identification indicator	Y
	fitted	*number*	Peak fit indicator	Y
	multiplet	*number*	Multiplet peak indicator	Y
	lc	*number*	Critical level for the peak	Y
	peak_sig	*number*	Peak significance	Y
	peak_tol	*number*	Peak energy tolerance	Y
	pss	*number*	Peak search significance parameter	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	detectability	*number*	Peak detectability	Y
	bkgnd_area	*number*	Background area	Y
	bkgnd_area_err	*number*	Uncertainty of the background area	Y
	net_area	*number*	Net area	Y
	net_area_err	*number*	Uncertainty of the net area	Y

Primary key	PK_GARDS_PEAKS(sample_id, peak_id)
Foreign keys	FK_PEAKS(sample_id, GARDS_SAMPLE_DATA)
Indexes	PEAKS_NDX(sample_id, peak_id)
Slaves	GARDS_PEAKS_ORIG
Triggers	UPDATE_PEAKS_ORIG_I_U_D, GARDS_PEAKS_I_U_D, PEAKS_B_I_U

5.1.101. GARDS_PEAKS

Owner(s)	RMSMAN
Description	The gards_peaks table contains information regarding peaks identified.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	peak_id	*number*	Peak identifier	N
	centroid	*number*	Peak centroid	Y
	centroid_err	*number*	Uncertainty of the peak centroid.	Y
I	energy	*number*	Energy of the peak	Y
	energy_err	*number*	Uncertainty of the energy	Y
	left_chan	*number*	Left channel of the peak	Y
	width	*number*	Width of the peak	Y
	back_count	*number*	Continuum background counts	Y
	back_uncer	*number*	Uncertainty of the continuum background counts	Y
	fwhm	*number*	Full width at half maximum	Y
	fwhm_err	*number*	Uncertainty of full width at half maximum	Y
	area	*number*	Peak area (counts)	Y
	area_err	*number*	Uncertainty of the peak area (counts)	Y
	original_area	*number*	Original peak area (counts)	Y
	original_uncer	*number*	Uncertainty of the original peak area (counts)	Y
	counts_sec	*number*	Counts per second	Y
	counts_sec_err	*number*	Uncertainty of the counts per second	Y
	efficiency	*number*	Efficiency	Y
	eff_error	*number*	Uncertainty of the efficiency	Y
	back_channel	*number*	Number of the average background counts per channel	Y
I	ided	*number*	Peak identification indicator	Y
	fitted	*number*	Peak fit indicator	Y
	multiplet	*number*	Multiplet peak indicator	Y
	lc	*number*	Critical level for the peak	Y
	peak_sig	*number*	Peak significance	Y
	peak_tol	*number*	Peak energy tolerance	Y
	pss	*number*	Peak search significance parameter	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	detectability	*number*	Peak detectability	Y
	bkgnd_area	*number*	Background area	Y
	bkgnd_area_err	*number*	Uncertainty of the background area	Y
	net_area	*number*	Net area	Y

	net_area_err	*number*	Uncertainty of the net area	Y

Primary key	PK_GARDS_PEAKS(sample_id, peak_id)
Foreign keys	FK_PEAKS(sample_id, GARDS_SAMPLE_DATA)
Indexes	PK_GARDS_PEAKS(sample_id, peak_id), PEAKS_NDX(sample_id, peak_id, energy, ided)
Slaves	GARDS_PEAKS_ORIG
Triggers	PEAKS_B_I_U

5.1.102. GARDS_PEAKS_ORIG

Owner(s)	RMSMAN
Description	The gards_peaks_orig table contains information regarding peaks identified during automatic analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	peak_id	*number*	Peak identifier	N
	centroid	*number*	Peak centroid	Y
	centroid_err	*number*	Uncertainty of the peak centroid.	Y
I	energy	*number*	Energy of the peak	Y
	energy_err	*number*	Uncertainty of the energy	Y
	left_chan	*number*	Left channel of the peak	Y
	width	*number*	Width of the peak	Y
	back_count	*number*	Continuum background counts	Y
	back_uncer	*number*	Uncertainty of the continuum background counts	Y
	fwhm	*number*	Full width at half maximum	Y
	fwhm_err	*number*	Uncertainty of full width at half maximum	Y
	area	*number*	Peak area (counts)	Y
	area_err	*number*	Uncertainty of the peak area (counts)	Y
	original_area	*number*	Original peak area (counts)	Y
	original_uncer	*number*	Uncertainty of the original peak area (counts)	Y
	counts_sec	*number*	Counts per second	Y
	counts_sec_err	*number*	Uncertainty of the counts per second	Y
	efficiency	*number*	Efficiency	Y
	eff_error	*number*	Uncertainty of the efficiency	Y
	back_channel	*number*	Number of the average background counts per channel	Y
I	ided	*number*	Peak identification indicator (0 or 1)	Y

	fitted	*number*	Peak fit indicator	Y
	multiplet	*number*	Multiplet peak indicator	Y
	lc	*number*	Critical level for the peak	Y
	peak_sig	*number*	Peak significance	Y
	peak_tol	*number*	Peak energy tolerance	Y
	pss	*number*	Peak search significance parameter	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	detectability	*number*	Peak detectability	Y

Primary key	PK_GARDS_PEAKS_ORIG(sample_id, peak_id)
Foreign keys	FK_PEAKS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	PEAKS_ORIG_B_I_U

5.1.103. GARDS_PEAKS_ORIG

Owner(s)	RMSAUTO
Description	The gards_peaks_orig table contains information regarding peaks identified during automatic analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PFI	peak_id	*number*	Peak identifier	N
	centroid	*number*	Peak centroid	Y
	centroid_err	*number*	Uncertainty of the peak centroid.	Y
I	energy	*number*	Energy of the peak	Y
	energy_err	*number*	Uncertainty of the energy	Y
	left_chan	*number*	Left channel of the peak	Y
	width	*number*	Width of the peak	Y
	back_count	*number*	Continuum background counts	Y
	back_uncer	*number*	Uncertainty of the continuum background counts	Y
	fwhm	*number*	Full width at half maximum	Y
	fwhm_err	*number*	Uncertainty of full width at half maximum	Y
	area	*number*	Peak area (counts)	Y
	area_err	*number*	Uncertainty of the peak area (counts)	Y
	original_area	*number*	Original peak area (counts)	Y
	original_uncer	*number*	Uncertainty of the original peak area (counts)	Y
	counts_sec	*number*	Counts per second	Y

	counts_sec_err	*number*	Uncertainty of the counts per second	Y
	efficiency	*number*	Efficiency	Y
	eff_error	*number*	Uncertainty of the efficiency	Y
	back_channel	*number*	Number of the average background counts per channel	Y
I	ided	*number*	Peak identification indicator (0 or 1)	Y
	fitted	*number*	Peak fit indicator	Y
	multiplet	*number*	Multiplet peak indicator	Y
	lc	*number*	Critical level for the peak	Y
	peak_sig	*number*	Peak significance	Y
	peak_tol	*number*	Peak energy tolerance	Y
	pss	*number*	Peak search significance parameter	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	detectability	*number*	Peak detectability	Y

Primary key	PK_GARDS_PEAKS_ORIG(sample_id, peak_id)
Foreign keys	FK_PEAKS_ORIG(sample_id, GARDS_SAMPLE_DATA), FK_SID_PEAKID(sample_id, GARDS_PEAKS), FK_SID_PEAKID(peak_id, GARDS_PEAKS)
Triggers	GARDS_PEAKS_ORIG_I_U_D, PEAKS_ORIG_B_I_U

5.1.104. GARDS_PERMISSIONS

Owner(s)	RMSMAN
Description	The gards_permissions table contains a list of permissions.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	permission_id	*number*	Permission identifier	N
	permission_name	*varchar2(30)*	Permission name.	N
	db_name	*varchar2(35)*	Database name	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_PERMISSIONS(permission_id)
Slaves	GARDS_ROLES_PERMISSIONS
Triggers	PERMISSIONS_B_I_U

5.1.105. GARDS_POC

Owner(s)	RMSMAN
Description	The gards_poc table contains information points of contact (POC) for radionuclide

	monitoring system.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	pocid	*number*	Point of contact identifier	N
	email_address	*varchar2(50)*	Email address of point of contact	Y
	first_name	*varchar2(20)*	First name of point of contact	Y
	last_name	*varchar2(50)*	Last name of point of contact	Y
	telephone	*varchar2(20)*	Telephone number of point of contact	Y
	address	*varchar2(500)*	Address of point of contact	Y
	additional_info	*varchar2(500)*	Additional information for point of contact	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_POC(pocid)
Indexes	PK_POC(pocid)
Slaves	GARDS_NOTIFY
Triggers	POC_B_I_U

5.1.106. GARDS_PROCESSING_ERRORS

Owner(s)	RMSMAN
Description	The gards_processing_errors table contains sample information, if applicable, and message information for RMS messages that have failed processing.

Pk/Fk/I	Column	Storage type	Description	Nullable
	rms_id	*number*	Internal identifier for message	Y
	sample_ref_id	*varchar2(32)*	Sample reference identification	Y
	msg_id	*number*	Message identifier	Y
	filename	*varchar2(512)*	Input data filename	Y
	station_code	*varchar2(5)*	Station code	Y
	detector_code	*varchar2(9)*	Detector code	Y
	data_type	*varchar2(12)*	Type of the data message	Y
	error_text	*varchar2(2048)*	Text associated with the error	Y
	entry_date	*date*	Date entered into the IDC database	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Triggers	PROCESSING_ERRORS_B_I_U

5.1.107. GARDS_PROC_PARAMS_TEMPLATE

Owner(s)	RMSMAN
Description	The gards_proc_params_template table is a template data record that contains parameters used by Automatic Analysis process software for peak search and nuclide identification. These parameters are used unless overridden at the command line. (Note: not used)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	station_id	*number*	Station identifier	N
PFI	detector_id	*number*	Detector identifier	N
	sample_type	*varchar2(2)*	Sample type	Y
PI	data_type	*char(1)*	Data type of the spectrum	N
PI	spectral_qualifier	*varchar2(5)*	Indicator of full or preliminary spectrum	N
	begin_date	*date*	Initialization date	Y
	end_date	*date*	Decommissioning date	Y
	do_back	*number*	Background subtraction indicator	Y
	back_data_type	*char(1)*	Spectral type of the detector background	Y
	nuclide_lib	*varchar2(96)*	Nuclide library to be used during analysis	Y
	mda_level	*number*	MDA confidence factor (percent)	Y
	nid_confid	*number*	NID confidence factor (percent)	Y
	squant_err	*number*	Uncertainty of sample quantity	Y
	buildtype	*varchar2(8)*	Flag indicating activity or concentration calculations	Y
	peak_sense	*number*	Peak search sensitivity	Y
	peak_start	*number*	Peak search start (keV)	Y
	peak_end	*number*	Peak search end (keV)	Y
	fwhm_mult_width	*number*	Number of FWHMs to search to determine multiplets	Y
	left_fwhm_lim	*number*	Left ROI FWHM limit	Y
	right_fwhm_lim	*number*	Right ROI FWHM limit	Y
	back_chan	*number*	Number of continuous channels	Y
	back_type	*varchar2(6)*	Background type: LINEAR or STEP	Y
	fit_singlets	*number*	Fit singlets flag	Y
	crit_level	*number*	Critical level indicator	Y
	fix_fwhm	*number*	Fixed FWHM during peak search flag	Y
	area_reject	*number*	Indicator of whether or not to reject peaks with zero area indicator	Y
	mdc_width	*number*	Baseline width used for MDA calculation	Y
	lc_abscissa	*number*	Abscissas of the normal distribution corresponding to a predefined confidence level for	Y

			estimating the critical level	
	do_pd_calc	*number*	Flag indicates if the parent/daughter calculations should be run	Y
	do_csc	*number*	Flag to turn off cascade summing correction	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_PROC_PARAMS_TEMPLATE(station_id, detector_id, data_type, spectral_qualifier)
Foreign keys	FK_PPTEMPL_DET(detector_id, GARDS_DETECTORS), FK_PPTEMPL_STA(station_id, GARDS_STATIONS)
Triggers	PROC_PARAMS_TEMPLATE_B_I_U

5.1.108. GARDS_PRODUCT

Owner(s)	RMSAUTO
Description	The gards_product table temporarily stores radionuclide reviewed products, baseline and SCAC information for quality assurance purposes.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	foff	*number*	Offset into the data file (bytes)	Y
	dsize	*number*	Size of data file in bytes	Y
	dir	*varchar2(256)*	Full path to the data file	Y
	dfile	*varchar2(32)*	Name of file containing the data	Y
	author	*varchar2(30)*	User name	Y
PI	revision	*number*	Revision identifier	N
PI	typeid	*number*	Unique type identifier	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_PRODUCT(sample_id, typeid, revision)
Foreign keys	FK_PRODUCT(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_PRODUCT_I_U_D, PRODUCT_B_I_U

5.1.109. GARDS_PRODUCT

Owner(s)	RMSMAN
Description	The gards_product table temporarily stores radionuclide reviewed products, baseline and SCAC information for quality assurance purposes.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	sample_id	*number*	Sample identifier	Y

	foff	*number*	Offset into the data file (bytes)	Y
	dsize	*number*	Size of data file in bytes	Y
	dir	*varchar2(256)*	Full path to the data file	Y
	dfile	*varchar2(32)*	Name of file containing the data	Y
	author	*varchar2(30)*	User name	Y
	revision	*number*	Revision identifier	Y
	typeid	*number*	Unique type identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Foreign keys	FK_PRODUCT(sample_id, GARDS_SAMPLE_DATA)
Triggers	PRODUCT_B_I_U

5.1.110. GARDS_QAT_CONFIG

Owner(s)	RMSMAN
Description	The gards_qat_config table contains q_time and t_time used by rms_QAT_auto. (Note: there is only one record in the database.)

Pk/Fk/I	Column	Storage type	Description	Nullable
	t_time	*number*	Number of minutes after gards_sample_status.entry_date that rms_QAT_auto waits before releasing a sample	Y
	min_t_time	*number*	Minimum allowable value for t_time	Y
	max_t_time	*number*	Maximum allowable value for t_time	Y
	q_time	*number*	Number of minutes after gards_sample_status.review_date that rms_QAT_auto waits before releasing a sample	Y
	poll_time	*number*	Number of minutes rms_QAT sleeps between each database poll	Y
	allow_release	*number*	Flag indicating whether or not release is permitted	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Triggers	QAT_CONFIG_B_I_U

5.1.111. GARDS_QAT_NOTIFY

Owner(s)	RMSMAN
Description	The gards_qat_notify table contains comments created by rms_QAT. (Note: not available)

Pk/Fk/I	Column	Storage type	Description	Nullable
	sample_id	*number*	Sample identifier	N
	comment_type	*number*	Comment code	Y
	comment_text	*varchar2(2048)*	Comment text	Y
	nucl_name	*varchar2(8)*	Nuclide name	Y
	energy	*number*	Gamma energy	Y
	dtg	*date*	Date time group for QAT comment notification	Y
	author	*varchar2(30)*	User name	Y
	recipient_list	*varchar2(128)*	List of recipients for QAT comment notification	Y
	moddate	*date*	Date/time at which the row was last modified	Y

5.1.112. GARDS_QAT_QUERY_FILTER

Owner(s)	RMSMAN
Description	Gards_qat_query_filter (Note: to be added.)

Pk/Fk/I	Column	Storage type	Description	Nullable
	user_id	*number*	User identifier	Y
	min_sample_id	*number*	QAT user query, lower boundary for the sample id	Y
	max_sample_id	*number*	QAT user query, upper boundary for the sample id	Y
	status_list	*varchar2(128)*	Comma-separated list of statuses	Y
	station_list	*varchar2(1024)*	Comma-separated list of stations	Y
	include_stations	*number*	Flag to include or exclude stations	Y
	category_list	*varchar2(32)*	Comma-separated list of categories	Y
	review_cat	*number*	Review category	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Triggers	QAT_QUERY_FILTER_B_I_U

5.1.113. GARDS_QCHISTORY

Owner(s)	RMSAUTO
Description	The gards_qchistory table contains output messages from quality control tests for each sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
PFI	sample_id	*number*	Sample identifier	N
PI	state	*varchar2(128)*	Text results of quality control hypothesis tests	N
	lddate	*date*	Load date	Y

Primary key	PK_GARDS_QCHISTORY(sample_id, state)
Foreign keys	FK_QCHISTORY(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_QCHISTORY_I_U_D

5.1.114. GARDS_QCHISTORY

Owner(s)	RMSMAN
Description	The gards_qchistory table contains output messages from quality control tests for each sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
	station_id	*number*	Station identifier	Y
	detector_id	*number*	Detector identifier	Y
PFI	sample_id	*number*	Sample identifier	N
PI	state	*varchar2(128)*	Text results of quality control hypothesis tests	N
	lddate	*date*	Load date	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_QCHISTORY(sample_id, state)
Foreign keys	FK_QCHISTORY(sample_id, GARDS_SAMPLE_DATA)
Triggers	QCHISTORY_B_I_U

5.1.115. GARDS_QCPARAMS

Owner(s)	RMSMAN
Description	The gards_qcparams table contains parameters for quality control processes.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	detector_id	*number*	Detector identifier	N
	gainchange	*number*	Gain first difference tolerance	Y
	sdmult	*number*	Location test standard deviation multiplier	Y
	width_adj	*number*	Peak width standard deviation inflation	Y

			factor	
	area_adj	*number*	Peak area standard deviation inflation factor	Y
	etol	*number*	Energy tolerance used to match peaks with energies (keV)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_QCPARAMS(detector_id)
Foreign keys	FK_QCPARAMS_DET(detector_id, GARDS_DETECTORS)
Triggers	QCPARAMS_B_I_U

5.1.116. GARDS_QCTARGETS

Owner(s)	RMSMAN
Description	The gards_qctargets table contains detector-specific target values for the quality control width and area tests.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	station_id	*number*	Station identifier	N
PFI	detector_id	*number*	Detector identifier	N
	dtg	*date*	Decay correction reference date	Y
PI	name	*varchar2(8)*	Nuclide name	N
PI	energy	*number*	Gamma energy	N
	muwidth	*number*	Target width for each nuclide line energy (keV)	Y
	muarea	*number*	Target area for each nuclide line energy (counts)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_QCTARGETS(name, detector_id, station_id, energy)
Foreign keys	FK_QCTARGS_STA(station_id, GARDS_STATIONS), FK_QCTARGS_DET(detector_id, GARDS_DETECTORS)
Triggers	QCTARGETS_B_I_U

5.1.117. GARDS_QC_RESULTS

Owner(s)	RMSAUTO
Description	Contains the result of QC performed for the given sample_id

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(22,0)*	Sample identifier	N
PI	test_name	*varchar2(32)*	Quality control test name	N

	flag	*char(1)*	Quality control flag	Y
	qc_comment	*varchar2(255)*	Quality control comments	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_QC_RESULTS(sample_id, test_name)
Foreign keys	FK_QC_RESULTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_QC_RESULTS_I_U_D, QC_RESULTS_B_I_U

5.1.118. GARDS_QC_RESULTS

Owner(s)	RMSMAN
Description	Contains the result of QC performed for the given sample_id

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(22,0)*	Sample identifier	N
PI	test_name	*varchar2(32)*	Quality control test name	N
	flag	*char(1)*	Quality control flag	Y
	qc_comment	*varchar2(255)*	Quality control comments	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_QC_RESULTS(sample_id, test_name)
Foreign keys	FK_QC_RESULTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	QC_RESULTS_B_I_U

5.1.119. GARDS_QUERY_RESULTS

Owner(s)	RMSMAN
Description	The gards_query_results table contains output of a user-initiated query from the event screening tool. This table is purged at the end of the request.

Pk/Fk/I	Column	Storage type	Description	Nullable
	rqst_id	*number*	Unique query identifier	Y
	rqst_string	*varchar2(100)*	Output of the user query	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Triggers	QUERY_RESULTS_B_I_U

5.1.120. GARDS_QUERY_RESULTS

Owner(s)	RMSAUTO

Description	The gards_query_results table contains output of a user-initiated query from the event screening tool. This table is purged at the end of the request.

Pk/Fk/I	Column	Storage type	Description	Nullable
	rqst_id	*number*	Unique query identifier	Y
	rqst_string	*varchar2(100)*	Output of the user query	Y
	moddate	*date*	Date/time at which the row was last modified	Y

5.1.121. GARDS_RECEIPT_LOG

Owner(s)	RMSAUTO
Description	(Note: Seems obsoleted, newest DTG date from 2001, no reference in RN software)

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	pocid	*number*	Point of contact identifier	N
PI	subid	*number*	Subscription id of obsolete table?	N
PI	prodid	*number*	Product id of obsolete table?	N
	type	*varchar2(5)*	Acknowledgment type of obsolete table?	Y
	dtg	*date*	Transmit date and time	Y
	kb_sent	*number*	Transmitted data size of obsolete table	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_RECEIPT_LOG(pocid, subid, prodid)

5.1.122. GARDS_REFLINE_MASTER

Owner(s)	RMSMAN
Description	The gards_refline_master table contains energy lines to be used for updating energy and resolution calibration based on peaks in the spectrum itself.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	refpeak_energy	*number*	Reference energy of the decay line	N
PI	data_type	*char(1)*	Data type of the spectrum	N
PI	spectral_qualifier	*varchar2(5)*	Indicator of full or preliminary spectrum	N
	calibration_type	*varchar2(3)*	Calibration type	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_REFLINE_MASTER(refpeak_energy, data_type, spectral_qualifier)
Triggers	REFLINE_MASTER_B_I_U

5.1.123. GARDS_RELEVANT_NUCLIDES

Owner(s)	RMSMAN
Description	The gards_relevant_nuclides table contains the list of CTBT relevant nuclides.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
P	type	*varchar2(16)*	Nuclide type	Y
PI	sample_type	*char(1)*	Sample type	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_RELEVANT_NUCL(name, sample_type)
Triggers	RELEVANT_NUCLIDES_B_I_U

5.1.124. GARDS_RESOLUTION_CAL

Owner(s)	RMSMAN
Description	The gards_resolution_cal table contains resolution calibration coefficients calculated during analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_RESOLUTION_CAL(sample_id, type)
Foreign keys	FK_RESOLUTION_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	RESOLUTION_CAL_B_I_U

5.1.125. GARDS_RESOLUTION_CAL

Owner(s)	RMSAUTO
Description	The gards_resolution_cal table contains resolution calibration coefficients calculated during analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	coeff1	*number*	zero order coefficient	Y
	coeff2	*number*	First order coefficient	Y
	coeff3	*number*	Second order coefficient	Y
	coeff4	*number*	Third order coefficient	Y
	coeff5	*number*	Fourth order coefficient	Y
	coeff6	*number*	Fifth order coefficient	Y
	coeff7	*number*	Sixth order coefficient	Y
	coeff8	*number*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_RESOLUTION_CAL(sample_id, type)
Foreign keys	FK_RESOLUTION_CAL(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_RESOLUTION_CAL_I_U_D, RESOLUTION_CAL_B_I_U

5.1.126. GARDS_RESOLUTION_CAL_COV

Owner(s)	RMSMAN
Description	The GARDS_RESOLUTION_CAL_COV table contains the covariance table for each coefficient of the resolution calibration from a given sample_id

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(22,0)*	Sample identifier	N
PI	row_index	*number(10,0)*	Row index	N
PI	col_index	*number(10,0)*	Column index	N
	moddate	*date*	Date/time at which the row was last modified	Y

	coeff	*float(126)*	Calibration coefficient	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_RESOLUTION_CAL_COV(sample_id, type, row_index, col_index)
Foreign keys	FK_RESOLUTION_CAL_COV(sample_id, GARDS_SAMPLE_DATA)
Triggers	RESOLUTION_CAL_COV_B_I_U

5.1.127. GARDS_RESOLUTION_CAL_COV

Owner(s)	RMSAUTO
Description	The GARDS_RESOLUTION_CAL_COV table contains the covariance table for each coefficient of the resolution calibration from a given sample_id

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(22,0)*	Sample identifier	N
PI	row_index	*number(10,0)*	Row index	N
PI	col_index	*number(10,0)*	Column index	N
	moddate	*date*	Date/time at which the row was last modified	Y
	coeff	*float(126)*	Calibration coefficient	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_RESOLUTION_CAL_COV(sample_id, type, row_index, col_index)
Foreign keys	FK_RESOLUTION_CAL_COV(sample_id, GARDS_SAMPLE_DATA)
Triggers	RESOLUTION_CAL_COV_B_I_U, GARDS_RESOLUTION_CAL_COV_I_U_D

5.1.128. GARDS_RESOLUTION_CAL_ORIG

Owner(s)	RMSMAN
Description	The gards_resolution_cal_orig table contains the original resolution calibration coefficients calculated associated with spectral PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(8,0)*	Sample identifier	N
	coeff1	*number(8,0)*	zero order coefficient	Y
	coeff2	*number(8,0)*	First order coefficient	Y
	coeff3	*number(8,0)*	Second order coefficient	Y
	coeff4	*number(8,0)*	Third order coefficient	Y

	coeff5	*number(8,0)*	Fourth order coefficient	Y
	coeff6	*number(8,0)*	Fifth order coefficient	Y
	coeff7	*number(8,0)*	Sixth order coefficient	Y
	coeff8	*number(8,0)*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_RESOLUTION_CAL_ORIG(sample_id, type)
Foreign keys	FK_RESOLUTION_CAL_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	RESOLUTION_CAL_ORIG_B_I_U

5.1.129. GARDS_RESOLUTION_CAL_ORIG

Owner(s)	RMSAUTO
Description	The gards_resolution_cal_orig table contains the original resolution calibration coefficients calculated associated with spectral PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(8,0)*	Sample identifier	N
	coeff1	*number(8,0)*	zero order coefficient	Y
	coeff2	*number(8,0)*	First order coefficient	Y
	coeff3	*number(8,0)*	Second order coefficient	Y
	coeff4	*number(8,0)*	Third order coefficient	Y
	coeff5	*number(8,0)*	Fourth order coefficient	Y
	coeff6	*number(8,0)*	Fifth order coefficient	Y
	coeff7	*number(8,0)*	Sixth order coefficient	Y
	coeff8	*number(8,0)*	Seventh order coefficient	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_RESOLUTION_CAL_ORIG(sample_id, type)
Foreign keys	FK_RESOLUTION_CAL_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_RES_CAL_ORIG_I_U_D, RESOLUTION_CAL_ORIG_B_I_U

5.1.130. GARDS_RESOLUTION_PAIRS

Owner(s)	RMSMAN
Description	The gards_resolution_pairs table contains resolution calibration pairs information calculated during analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	res_energy	*number*	Gamma energy	N
	resolution	*number*	Resolution	Y
	res_error	*number*	Uncertainty of the resolution	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_RESOLUTION_PAIRS(sample_id, res_energy, type)
Foreign keys	FK_RESOLUTION_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	RESOLUTION_PAIRS_B_I_U

5.1.131. GARDS_RESOLUTION_PAIRS

Owner(s)	RMSAUTO
Description	The gards_resolution_pairs table contains resolution calibration pairs information calculated during analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	res_energy	*number*	Gamma energy	N
	resolution	*number*	Resolution	Y
	res_error	*number*	Uncertainty of the resolution	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_RESOLUTION_PAIRS(sample_id, res_energy, type)
Foreign keys	FK_RESOLUTION_PAIRS(sample_id, GARDS_SAMPLE_DATA)
Triggers	RESOLUTION_PAIRS_B_I_U, GARDS_RESOLUTION_PAIRS_I_U_D

5.1.132. GARDS_RESOLUTION_PAIRS_ORIG

Owner(s)	RMSAUTO
Description	The gards_resolution_pairs_orig table contains the original resolution calibration pairs information associated with spectral PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	res_energy	*number*	Gamma energy	N
	resolution	*number*	Resolution	Y
	res_error	*number*	Uncertainty of the resolution	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y
PI	type	*char(8)*	Calibration type	N

Primary key	PK_GARDS_RESOLUTION_PAIRS_ORIG(sample_id, res_energy, type)
Foreign keys	FK_RESOLUTION_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_RES_PAIRS_ORIG_I_U_D, RESOLUTION_PAIRS_ORIG_B_I_U

5.1.133. GARDS_RESOLUTION_PAIRS_ORIG

Owner(s)	RMSMAN
Description	The gards_resolution_pairs_orig table contains the original resolution calibration pairs information associated with spectral PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	res_energy	*number*	Gamma energy	N
	resolution	*number*	Resolution	Y
	res_error	*number*	Uncertainty of the resolution	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PI	type	*char(8)*	Calibration type	N
	winner	*char(1)*	Flag indicating whether the calibration type is used (Y) or not (N)	Y

Primary key	PK_GARDS_RESOLUTION_PAIRS_ORIG(sample_id, res_energy, type)
Foreign keys	FK_RESOLUTION_PAIRS_ORIG(sample_id, GARDS_SAMPLE_DATA)
Triggers	RESOLUTION_PAIRS_ORIG_B_I_U

5.1.134. GARDS_RLR

Owner(s)	RMSAUTO
Description	The gards_rlr table contains the data related to a Radionuclide Laboratory Report (RLR) message.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PI	rlr_id	*number*	Unique RLR identifier	N
	station_id	*number*	Station identifier	Y
	lab_id	*number*	Unique laboratory identifier	Y
	lab_detector_id	*number*	Unique laboratory detector identifier	Y
	orig_sample_ref_id	*varchar2(32)*	Original/entire sample reference identification	Y
	split_sample_ref_id	*varchar2(32)*	Split sample reference identification	Y
	report_type	*varchar2(3)*	Report type	Y
	lab_sample_category	*varchar2(2)*	Report sample category	Y
	report_number	*number*	Report number	Y
	priority_level	*varchar2(10)*	Priority level (Urgent or Routine)	Y
	collect_start	*date*	Collection start date and time	Y
	collect_stop	*date*	Collection stop date and time	Y
	entry_date	*date*	Date entered into the IDC database	Y
	test_date	*date*	Test completion date at lab	Y
	transmit_date	*date*	Transmit date and time	Y
	filename	*varchar2(256)*	Input data filename	Y
	lab_receipt_date	*date*	Date of sample receipts at the laboratory	Y
	activity_date	*date*	Activity reference date	Y
	concentration_date	*date*	Concentration reference date	Y
	test_type	*varchar2(256)*	Type of test performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	RLRID_PK(rlr_id)
Indexes	RLRID_PK(rlr_id)
Slaves	GARDS_RLR_CONCLUSIONS, GARDS_RLR_OBJECTIVE, GARDS_RLR_RATIOS, GARDS_RLR_RESULTS, GARDS_RLR_SSREB
Triggers	GARDS_RLR_B_I_U

5.1.135. GARDS_RLR_CONCLUSIONS

Owner(s)	RMSAUTO
Description	The gards_rlr_conclusions table contains IDC findings, lab findings and comparisons of IDC and lab results.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	rlr_id	*number*	Unique RLR identifier	N
	idc_summary	*varchar2(2048)*	IDC findings	Y
	lab_summary	*varchar2(2048)*	Laboratory findings and conclusions	Y
	result_comparison	*varchar2(2048)*	Comparison of IDC and laboratory results	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RLR_CONCLUSIONS(rlr_id)
Foreign keys	SYS_C0028565(rlr_id, GARDS_RLR)
Triggers	RLR_CONCLUSIONS_B_I_U

5.1.136. GARDS_RLR_OBJECTIVE

Owner(s)	RMSAUTO
Description	The gards_rlr_objective table contains the objectives of the lab analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	rlr_id	*number*	Unique RLR identifier	N
	analysis_purpose	*varchar2(2048)*	Analysis purpose	Y
	tests_authorized	*varchar2(2048)*	Tests authorized	Y
	special_instructions	*varchar2(2048)*	Special instructions	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RLR_OBJECTIVE(rlr_id)
Foreign keys	SYS_C0028566(rlr_id, GARDS_RLR)
Triggers	RLR_OBJECTIVE_B_I_U

5.1.137. GARDS_RLR_RATIOS

Owner(s)	RMSAUTO

Description	The gards_rlr_ratios table contains the nuclides activity ratios

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	rlr_id	*number*	Unique RLR identifier	N
PI	nuclide1	*varchar2(8)*	Nuclide 1	N
PI	nuclide2	*varchar2(8)*	Nuclide 2	N
PI	activity_ratio	*number*	Activity ratio of nuclide 1 to nuclide 2	N
	activity_ratio_err	*number*	Relative uncertainty of the activity ratio	Y
	reference_date	*date*	Reference date of the activity ratio	Y
	zero_date	*date*	Zero date	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RLR_RATIOS(rlr_id, nuclide1, nuclide2, activity_ratio)
Foreign keys	SYS_C0028567(rlr_id, GARDS_RLR)
Triggers	RLR_RATIOS_B_I_U

5.1.138. GARDS_RLR_RESULTS

Owner(s)	RMSAUTO
Description	The gards_rlr_results table contains the results of tests conducted at the laboratory.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	rlr_id	*number*	Unique RLR identifier	N
PI	nuclide_name	*varchar2(8)*	Nuclide name	N
PI	activity	*number*	Activity at the start of acquisition	N
	activity_err	*number*	Relative uncertainty of the activity	Y
	activ_coverage_factor	*number*	Coverage factor	Y
	activ_confidence_level	*number*	Level of confidence	Y
	concentration	*number*	Concentration	Y
	concentration_err	*number*	Relative uncertainty of the activity concentration	Y
	conc_coverage_factor	*number*	Coverage factor	Y
	conc_confidence_level	*number*	Level of confidence	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RLR_RESULTS(rlr_id, nuclide_name, activity)
Foreign keys	SYS_C0028568(rlr_id, GARDS_RLR)
Triggers	RLR_RESULTS_B_I_U

5.1.139. GARDS_RLR_SSREB

Owner(s)	RMSAUTO
Description	The gards_rlr_ssreb table contains the list of RLRs that have been added to their corresponding SSREB.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	rlr_id	*number*	Unique RLR identifier	N
PFI	sample_id	*number*	Sample identifier	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RLR_SSREB(rlr_id, sample_id)
Foreign keys	SYS_C0028569(rlr_id, GARDS_RLR), SYS_C0028570(sample_id, GARDS_SAMPLE_DATA)
Triggers	RLR_SSREB_B_I_U

5.1.140. GARDS_RN_INTERVAL

Owner(s)	RMSAUTO
Description	Used for RN monitoring with Workflow

Pk/Fk/I	Column	Storage type	Description	Nullable
I	class	*varchar2(16)*	Workflow field indicating the station	N
I	name	*varchar2(16)*	Workflow field indicating the message type	N
	start_time	*float(126)*	Start time of the period	N
I	end_time	*float(126)*	End time of the period	N
I	state	*varchar2(16)*	Workflow field indicating the status	N
PI	intvlid	*number*	Unique interval identifier	N
I	refid	*number(10,0)*	Workflow field indicating the sample id	N
I	refidtype	*varchar2(16)*	Workflow field indicating whether it is an input interval, PHD, SOH	N
I	moddate	*date*	Date/time at which the row was last modified	N
	lddate	*date*	Load date	N
	station_code	*varchar2(5)*	Station code	Y

Primary key	PK_RN_INTERVAL(intvlid)
Triggers	RN_INTERVAL_S_I

5.1.141. GARDS_ROI_CHANNELS

Owner(s)	RMSAUTO
Description	The gards_roi_channels table contains the ROI boundaries in channel units.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	b_chan_start	*number*	Beta boundary start channel	Y
	b_chan_stop	*number*	Beta boundary stop channel	Y
	g_chan_start	*number*	Gamma boundary start channel	Y
	g_chan_stop	*number*	Gamma boundary stop channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_CHANNELS(sample_id, roi)
Foreign keys	FK_ROI_CHANNELS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_ROI_CHANNELS_I_U_D, ROI_CHANNELS_B_I_U

5.1.142. GARDS_ROI_CHANNELS

Owner(s)	RMSMAN
Description	The gards_roi_channels table contains the ROI boundaries in channel units.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	b_chan_start	*number*	Beta boundary start channel	Y
	b_chan_stop	*number*	Beta boundary stop channel	Y
	g_chan_start	*number*	Gamma boundary start channel	Y
	g_chan_stop	*number*	Gamma boundary stop channel	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_CHANNELS(sample_id, roi)
Foreign keys	FK_ROI_CHANNELS(sample_id, GARDS_SAMPLE_DATA)
Triggers	ROI_CHANNELS_B_I_U

5.1.143. GARDS_ROI_CONCS

Owner(s)	RMSAUTO
Description	The gards_roi_concs table contains information regarding the concentration of each ROI. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	activity	*number*	Activity concentration	Y
	activ_err	*number*	Uncertainty of the activity concentration	Y
	mda	*number*	Minimum detectable concentration	Y
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	report_mda	*number*	Indicator of whether MDA is to be reported	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_CONCS(sample_id, roi)
Foreign keys	FK_ROI_CONCS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_ROI_CONCS_I_U_D, ROI_CONCS_B_I_U

5.1.144. GARDS_ROI_CONCS

Owner(s)	RMSMAN
Description	The gards_roi_concs table contains information regarding the concentration of each ROI. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	activity	*number*	Activity concentration	Y
	activ_err	*number*	Uncertainty of the activity concentration	Y
	mda	*number*	Minimum detectable concentration	Y
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	report_mda	*number*	Indicator of whether MDA is to be reported	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_CONCS(sample_id, roi)
Foreign keys	FK_ROI_CONCS(sample_id, GARDS_SAMPLE_DATA)
Triggers	ROI_CONCS_B_I_U

5.1.145. GARDS_ROI_COUNTS

Owner(s)	RMSMAN
Description	The gards_roi_counts table contains information regarding the counts for each ROI. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	gross	*number*	Gross number of counts in the ROI	Y
	gross_err	*number*	Uncertainty of the gross number of counts in the ROI	Y
	compton	*number*	Counts of Compton continuum background	Y
	compton_err	*number*	Uncertainty of Compton counts	Y
	interference	*number*	Counts from interference nuclides	Y
	interference_err	*number*	Uncertainty of interference (counts)	Y
	memory	*number*	Gas background counts	Y
	memory_err	*number*	Uncertainty of memory (counts)	Y
	detector_back	*number*	Detector background counts	Y
	detector_back_err	*number*	Uncertainty of detector_back (counts)	Y
	net	*number*	Counts in ROI after processing	Y
	net_err	*number*	Uncertainty of net (counts)	Y
	lc	*number*	Critical level (counts)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_COUNTS(sample_id, roi)
Foreign keys	FK_ROI_COUNTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	ROI_COUNTS_B_I_U

5.1.146. GARDS_ROI_COUNTS

Owner(s)	RMSAUTO
Description	The gards_roi_counts table contains information regarding the counts for each ROI. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N

PI	roi	*number*	ROI identifier	N
	gross	*number*	Gross number of counts in the ROI	Y
	gross_err	*number*	Uncertainty of the gross number of counts in the ROI	Y
	compton	*number*	Counts of Compton continuum background	Y
	compton_err	*number*	Uncertainty of Compton counts	Y
	interference	*number*	Counts from interference nuclides	Y
	interference_err	*number*	Uncertainty of interference (counts)	Y
	memory	*number*	Gas background counts	Y
	memory_err	*number*	Uncertainty of memory (counts)	Y
	detector_back	*number*	Detector background counts	Y
	detector_back_err	*number*	Uncertainty of detector_back (counts)	Y
	net	*number*	Counts in ROI after processing	Y
	net_err	*number*	Uncertainty of net (counts)	Y
	lc	*number*	Critical level (counts)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_COUNTS(sample_id, roi)
Foreign keys	FK_ROI_COUNTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_ROI_COUNTS_I_U_D, ROI_COUNTS_B_I_U

5.1.147. GARDS_ROI_LIB

Owner(s)	RMSMAN
Description	The gards_roi_lib table contains the information between each ROI and specific nuclide. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	roi	*number*	ROI identifier	N
	name	*varchar2(8)*	Nuclide name	Y
	halflife	*varchar2(23)*	Half-life	Y
	halflife_err	*varchar2(23)*	Uncertainty of half-life	Y
	halflife_sec	*number*	Half-life in seconds	Y
	abundance	*number*	Emission intensity per disintegration	Y
	abundance_err	*number*	Uncertainty of the abundance	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_LIB(roi)
Triggers	ROI_LIB_B_I_U

5.1.148. GARDS_ROI_LIMITS

Owner(s)	RMSAUTO
Description	The gards_roi_limits table contains the ROI boundaries in energy units as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	b_energy_start	*number*	Beta boundary start energy (keV)	Y
	b_energy_stop	*number*	Beta boundary stop energy (keV)	Y
	g_energy_start	*number*	Gamma boundary start energy (keV)	Y
	g_energy_stop	*number*	Gamma boundary stop energy (keV)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_LIMITS(sample_id, roi)
Foreign keys	FK_ROI_LIMITS(sample_id, GARDS_SAMPLE_DATA)
Triggers	ROI_LIMITS_B_I_U, GARDS_ROI_LIMITS_I_U_D

5.1.149. GARDS_ROI_LIMITS

Owner(s)	RMSMAN
Description	The gards_roi_limits table contains the ROI boundaries in energy units as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	b_energy_start	*number*	Beta boundary start energy (keV)	Y
	b_energy_stop	*number*	Beta boundary stop energy (keV)	Y
	g_energy_start	*number*	Gamma boundary start energy (keV)	Y
	g_energy_stop	*number*	Gamma boundary stop energy (keV)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_ROI_LIMITS(sample_id, roi)
Foreign keys	FK_ROI_LIMITS(sample_id, GARDS_SAMPLE_DATA)
Triggers	ROI_LIMITS_B_I_U

5.1.150. GARDS_ROLES

Owner(s)	RMSMAN

Description	The gards_roles table contains the list of roles.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	role_id	*number*	Role identifier	N
	role_name	*varchar2(30)*	Role name	N
	db_name	*varchar2(35)*	Database name	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_ROLES(role_id)
Indexes	PK_ROLES(role_id)
Slaves	GARDS_USERS_ROLES, GARDS_ROLES_PERMISSIONS
Triggers	ROLES_B_I_U

5.1.151. GARDS_ROLES_PERMISSIONS

Owner(s)	RMSMAN
Description	The gards_roles_permissions table contains a mapping of which permissions are assigned to which roles.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	role_id	*number*	Role identifier	N
PFI	permission_id	*number*	Permission identifier	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_ROLES_PERM(role_id, permission_id)
Foreign keys	SYS_C0028509(permission_id, GARDS_PERMISSIONS), SYS_C0028510(role_id, GARDS_ROLES)
Triggers	ROLES_PERMISSIONS_B_I_U

5.1.152. GARDS_SAINT_DEFAULT_PARAMS

Owner(s)	RMSAUTO, RMSMAN
Description	The GARDS_DEFAULT_PARAMS table contains default processing parameters for HPGe spectra.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(40)*	Parameter name	N
	value	*varchar2(255)*	Parameter default setting	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAINT_DEFAULT_PARAMS(name)
Triggers	DEFAULT_PARAMS_B_I_U

5.1.153. GARDS_SAINT_PROCESS_PARAMS

Owner(s)	RMSMAN
Description	The GARDS_SAINT_PROCESS_PARAMS table contains spectrum specific processing parameters for HPGe spectra

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	name	*varchar2(40)*	Parameter name	N
	value	*varchar2(255)*	Parameter setting	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAINT_PROCESS_PARAMS(sample_id, name)
Foreign keys	FK_SAINT_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	PROCESS_PARAMS_B_I_U

5.1.154. GARDS_SAINT_PROCESS_PARAMS

Owner(s)	RMSAUTO
Description	The GARDS_SAINT_PROCESS_PARAMS table contains spectrum specific processing parameters for HPGe spectra

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	name	*varchar2(40)*	Parameter name	N
	value	*varchar2(255)*	Parameter setting	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAINT_PROCESS_PARAMS(sample_id, name)
Foreign keys	FK_SAINT_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_SAINT_PROC_PAR_I_U_D, PROCESS_PARAMS_B_I_U

5.1.155. GARDS_SAMPLE (VIEW)

Owner(s)	RMSMAN
Description	The GARDS_SAMPLE is a view regarding the sample data and information related to sample analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable

	site_det_code	*char(15)*	Station code concatenated with detector code	Y
	sample_id	*number*	Sample identifier	N
	station_id	*number*	Station identifier	Y
	station_code	*varchar2(5)*	Station code	Y
	detector_id	*number*	Detector identifier	Y
	detector_code	*varchar2(9)*	Detector code	Y
	input_file_name	*varchar2(255)*	Input data filename	Y
	sample_type	*char(1)*	Sample type	Y
	data_type	*char(1)*	Data type of the spectrum	Y
	geometry	*varchar2(17)*	Sample geometry	Y
	spectral_qualifier	*varchar2(5)*	Indicator of full or preliminary spectrum	Y
	transmit_dtg	*date*	Transmit date and time	Y
	collect_start	*date*	Collection start date and time	Y
	collect_stop	*date*	Collection stop date and time	Y
	acquisition_start	*date*	Acquisition start date and time	Y
	acquisition_stop	*date*	Acquisition stop date and time	Y
	acquisition_real_sec	*number*	Total time between the acquisition stop and acquisition start (seconds)	Y
	acquisition_live_sec	*number*	Active detection time (seconds)	Y
	quantity	*number*	Air volume sampled	Y
	entry_date	*date*	Date entered into the IDC database	Y
	cnf_begin_date	*date*	Date when last analysis began	Y
	cnf_end_date	*date*	Date when last analysis ended	Y
	review_date	*date*	Date sample was last reviewed	Y
	review_time	*number*	Amount of time to review sample (minutes)	Y
	analyst	*varchar2(30)*	Name of the analyst who reviewed the sample	Y
	status	*char(1)*	Current processing status of sample.	Y
	category	*number*	Categorization level	Y

SQL Body	SELECT site_det_code, gards_sample_data.sample_id, gards_sample_data.station_id, station_code, gards_sample_data.detector_id, detector_code, input_file_name, sample_type, data_type, gards_sample_data.geometry, spectral_qualifier, transmit_dtg, collect_start, collect_stop, acquisition_start, acquisition_stop, acquisition_real_sec, acquisition_live_sec, quantity, entry_date, cnf_begin_date, cnf_end_date, review_date, review_time, analyst, gards_sample_status.status, category FROM gards_sample_data,gards_sample_status,gards_stations,gards_de- tectors WHERE gards_sample_data.sample_id = gards_sample_status.sample_id AND gards_sample_data.station_id= gards_stations.station_id AND gards_sample_data.detector_id = gards_detectors.de- tector_id

5.1.156. GARDS_SAMPLE_AUX

Owner(s)	RMSAUTO
Description	The gards_sample_aux table contains auxiliary information related to raw sample data.

Pk/Fk/I	**Column**	**Storage type**	**Description**	**Nullable**
PFI	sample_id	*number*	Sample identifier	N
I	sample_ref_id	*varchar2(32)*	Sample reference identification	N

I	measurement_id	*varchar2(31)*	Measurement identifier	N
	bkgd_measure-ment_id	*varchar2(31)*	Detector background measurement identifier	N
	sample_height	*number*	Height of sample (cm)	N
	calibration_dtg	*date*	Date of most recent calibration at detector site	Y
I	msg_id	*number*	Message identifier	N
	archive_bottle_id	*char(2)*	Unique identifier of the archive bottle in which the sample is stored	Y
	gas_bkgd_meas-urement_id	*varchar2(31)*	Gas background measurement identifier	N
	xe_collect_yield	*number*	Xenon collection yield of the system	N
	xe_collect_yield_err	*number*	Uncertainty of the xenon collection yield	N
	xe_volume	*number*	Xenon volume	N
	xe_volume_err	*number*	Uncertainty of the xenon volume	N
	sample_diameter	*number*	Diameter of the sample	N
	moddate	*date*	Date/time at which the row was last modi-fied	N

Primary key	PK_SAMPLE_AUX(sample_id)
Foreign keys	FK_SAMPLE_AUX(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_AUX_B_I_U

5.1.157. GARDS_SAMPLE_CAT

Owner(s)	RMSMAN
Description	The gards_sample_cat table contains nuclide categorization results for particulate samples.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	name	*varchar2(8)*	Nuclide name	N
PI	method_id	*number*	Method identifier	N
	category	*number*	Categorization level	Y
	upper_bound	*number*	Upper limit of the amount of a nuclide that can be acceptably found	Y
	lower_bound	*number*	Lower limit of the amount of a nuclide that can be acceptably found	Y
	central_value	*number*	Current estimate of the level of the random process	Y
	delta	*number*	Value of a variable used in bounds estima-tion (EWMA algorithm)	Y

	activity	*number*	Activity concentration of the nuclide detected.	Y
	hold	*number*	Flag for holding current sample from being used in the history for categorizing subsequent samples (0 = use; 1 = skip)	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_CAT(sample_id, name, method_id)
Foreign keys	FK_SAMP_ID(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_CAT_B_I_U

5.1.158. GARDS_SAMPLE_CERT

Owner(s)	RMSAUTO
Description	The gards_sample_cert table contains the total activity and reference date from the source certificate of QCPHD and CALIBPHD messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	quantity	*number*	Total activity of the source	Y
	assay_date	*date*	Date sample was assayed	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_CERT(sample_id)
Foreign keys	FK_SAMPLE_CERT(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_CERT_B_I_U

5.1.159. GARDS_SAMPLE_CERT_LINES

Owner(s)	RMSAUTO
Description	The gards_sample_cert_lines table contains the nuclide data from the source certificate of QCPHD and CALIBPHD messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	nucl_name	*varchar2(8)*	Nuclide name	N
	halflife	*varchar2(23)*	Half-life	Y
PI	energy	*number*	Energy of the decay line	N
	activity	*number*	Activity at time of assay	Y
	error	*number*	Relative uncertainty of the activity	Y
	abundance	*number*	Emission intensity per disintegration	Y

	b_abundance	*number*	Emission intensity (%) of beta particle or conversion electron per disintegration of one nucleus	Y
PI	b_energy	*number*	Energy of conversion electron or maximum energy of beta particle (keV)	N
	decay_mode	*char(1)*	Decay type	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_CERT_LINES(sample_id, nucl_name, energy, b_energy)
Foreign keys	FK_SAMPLE_CERT_LINES(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_CERT_LINES_B_I_U

5.1.160. GARDS_SAMPLE_DATA

Owner(s)	RMSMAN
Description	The gards_sample_data table contains the basic data of sample collection and spectrum acquisition in PHD messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
	site_det_code	*char(15)*	Station code concatenated with detector code	Y
PI	sample_id	*number*	Sample identifier	N
I	station_id	*number*	Station identifier	Y
I	detector_id	*number*	Detector identifier	Y
	input_file_name	*varchar2(255)*	Input data filename	Y
I	sample_type	*char(1)*	Sample type	Y
I	data_type	*char(1)*	Data type of the spectrum	Y
	geometry	*varchar2(17)*	Sample geometry	Y
I	spectral_qualifier	*varchar2(5)*	Indicator of full or preliminary spectrum	Y
	transmit_dtg	*date*	Transmit date and time	Y
I	collect_start	*date*	Collection start date and time	Y
I	collect_stop	*date*	Collection stop date and time	Y
I	acquisition_start	*date*	Acquisition start date and time	Y
I	acquisition_stop	*date*	Acquisition stop date and time	Y
	acquisition_real_sec	*number*	Total time between the acquisition stop and acquisition start (seconds)	Y
	acquisition_live_sec	*number*	Active detection time (seconds)	Y
	quantity	*number*	Air volume sampled	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_SAMPLE_DATA(sample_id)
Triggers	SAMPLE_DATA_B_I_U, SAMPLE_DATA_INSERT

5.1.161. GARDS_SAMPLE_DATA

Owner(s)	RMSAUTO
Description	The gards_sample_data table contains the basic data of sample collection and spectrum acquisition in PHD messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
	site_det_code	*char(15)*	Station code concatenated with detector code	Y
PI	sample_id	*number*	Sample identifier	N
I	station_id	*number*	Station identifier	Y
I	detector_id	*number*	Detector identifier	Y
	input_file_name	*varchar2(255)*	Input data filename	Y
I	sample_type	*char(1)*	Sample type	Y
I	data_type	*char(1)*	Data type of the spectrum	Y
	geometry	*varchar2(17)*	Sample geometry	Y
I	spectral_qualifier	*varchar2(5)*	Indicator of full or preliminary spectrum	Y
I	transmit_dtg	*date*	Transmit date and time	Y
	collect_start	*date*	Collection start date and time	Y
I	collect_stop	*date*	Collection stop date and time	Y
I	acquisition_start	*date*	Acquisition start date and time	Y
I	acquisition_stop	*date*	Acquisition stop date and time	Y
	acquisition_real_sec	*number*	Total time between the acquisition stop and acquisition start (seconds)	Y
	acquisition_live_sec	*number*	Active detection time (seconds)	Y
	quantity	*number*	Air volume sampled	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_SAMPLE_DATA(sample_id)
Triggers	SAMPLE_DATA_B_I_U, GARDS_SAMPLE_DATA_I_U_D

5.1.162. GARDS_SAMPLE_DESCRIPTION

Owner(s)	RMSAUTO
Description	The gards_sample_description table contains description and comment text as specified in a PHD file.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	description	*varchar2(512)*	Comments	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_SAMPLE_DESCRIPTION(sample_id)
Foreign keys	FK_SAMPLE_DESCRIPTION(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_DESCRIPTION_B_I_U

5.1.163. GARDS_SAMPLE_FLAGS

Owner(s)	RMSAUTO
Description	The gards_sample_flags table contains the results of the rms_DBflags analysis for each sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	flag_id	*number*	Flag identifier	N
	result	*number*	Indicator of whether or not the test passed	Y
	value	*number*	Threshold value used for the test	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_FLAGS(sample_id, flag_id)
Foreign keys	FK_SAMPLE_FLAGS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_SAMPLE_FLAGS_I_U_D, SAMPLE_FLAGS_B_I_U

5.1.164. GARDS_SAMPLE_FLAGS

Owner(s)	RMSMAN
Description	The gards_sample_flags table contains the results of the rms_DBflags analysis for each sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	flag_id	*number*	Flag identifier	N
	result	*number*	Indicator of whether or not the test passed	Y
	value	*number*	Threshold value used for the test	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_FLAGS(sample_id, flag_id)
Foreign keys	FK_SAMPLE_FLAGS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_FLAGS_B_I_U

5.1.165. GARDS_SAMPLE_PROC_PARAMS

Owner(s)	RMSMAN
Description	The gards_sample_proc_params table contains specific processing parameters to be used for a particular spectral analysis; the table contains a combination of values from gards_proc_params_template and values entered at the command line. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	do_back	*number*	Background subtraction indicator	Y
	back_data_type	*char(1)*	Spectral type of the detector background	Y
	back_sample_id	*number*	Sample identifier for the sample used in background subtraction	Y
	nuclide_lib	*varchar2(96)*	Nuclide library to be used during analysis	Y
	energy_tol	*number*	Tolerance for nuclide identification	Y
	mda_level	*number*	MDA confidence factor (percent)	Y
	nid_confid	*number*	NID confidence factor (percent)	Y
	squant_err	*number*	Uncertainty of sample quantity	Y
	buildtype	*varchar2(8)*	Flag indicating activity or concentration calculations	Y
	peak_sense	*number*	Peak search sensitivity	Y
	peak_start	*number*	Peak search start (keV)	Y
	peak_end	*number*	Peak search end (keV)	Y
	fwhm_mult_width	*number*	Number of FWHMs to search to determine multiplets	Y
	left_fwhm_lim	*number*	Left ROI FWHM limit	Y
	right_fwhm_lim	*number*	Right ROI FWHM limit	Y
	back_chan	*number*	Number of continuous channels	Y
	back_type	*varchar2(6)*	Background type: LINEAR or STEP	Y
	fit_singlets	*number*	Fit singlets flag	Y
	crit_level	*number*	Critical level indicator	Y
	fix_fwhm	*number*	Fixed FWHM during peak search flag	Y
	area_reject	*number*	Indicator of whether or not to reject peaks with zero area indicator	Y
	mdc_width	*number*	Baseline width used for MDA calculation	Y
	lc_abscissa	*number*	Abscissas of the normal distribution corres-	Y

			ponding to a predefined confidence level for estimating the critical level	
	do_pd_calc	*number*	Flag indicates if the parent/daughter calculations should be run	Y
	do_csc	*number*	Flag to turn off cascade summing correction	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_PROC_PARS(sample_id)
Foreign keys	FK_SAMPLE_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_PROC_PARAMS_B_I_U

5.1.166. GARDS_SAMPLE_PROC_PARAMS

Owner(s)	RMSAUTO
Description	The gards_sample_proc_params table contains specific processing parameters to be used for a particular spectral analysis; the table contains a combination of values from gards_proc_params_template and values entered at the command line. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	do_back	*number*	Background subtraction indicator	Y
	back_data_type	*char(1)*	Spectral type of the detector background	Y
	back_sample_id	*number*	Sample identifier for the sample used in background subtraction	Y
	nuclide_lib	*varchar2(96)*	Nuclide library to be used during analysis	Y
	energy_tol	*number*	Tolerance for nuclide identification	Y
	mda_level	*number*	MDA confidence factor (percent)	Y
	nid_confid	*number*	NID confidence factor (percent)	Y
	squant_err	*number*	Uncertainty of sample quantity	Y
	buildtype	*varchar2(8)*	Flag indicating activity or concentration calculations	Y
	peak_sense	*number*	Peak search sensitivity	Y
	peak_start	*number*	Peak search start (keV)	Y
	peak_end	*number*	Peak search end (keV)	Y
	fwhm_mult_width	*number*	Number of FWHMs to search to determine multiplets	Y
	left_fwhm_lim	*number*	Left ROI FWHM limit	Y
	right_fwhm_lim	*number*	Right ROI FWHM limit	Y
	back_chan	*number*	Number of continuous channels	Y
	back_type	*varchar2(6)*	Background type: LINEAR or STEP	Y

	fit_singlets	*number*	Fit singlets flag	Y
	crit_level	*number*	Critical level indicator	Y
	fix_fwhm	*number*	Fixed FWHM during peak search flag	Y
	area_reject	*number*	Indicator of whether or not to reject peaks with zero area indicator	Y
	mdc_width	*number*	Baseline width used for MDA calculation	Y
	lc_abscissa	*number*	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level	Y
	do_pd_calc	*number*	Flag indicates if the parent/daughter calculations should be run	Y
	do_csc	*number*	Flag to turn off cascade summing correction	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_PROC_PARAMS(sample_id)
Foreign keys	FK_SAMPLE_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_PROC_PARAMS_B_I_U, GARDS_SAMPLE_PROC_PARAMS_I_U_D

5.1.167. GARDS_SAMPLE_RATIOS

Owner(s)	RMSAUTO
Description	The gards_sample_ratios table contains count ratios of a higher energy ROI to a lower energy ROI specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	ratio_id	*varchar2(15)*	Ratio identifier	N
	upper_roi_number	*number*	Upper ROI number	Y
	lower_roi_number	*number*	Lower ROI number	Y
	count_ratio	*number*	Ratio of the upper ROI counts to lower ROI counts	Y
	count_ratio_err	*number*	Uncertainty of the ratio	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_RATIOS(sample_id, ratio_id)
Foreign keys	FK_SAMPLE_RATIOS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_RATIOS_B_I_U, GARDS_SAMPLE_RATIOS_I_U_D

5.1.168. GARDS_SAMPLE_RATIOS

Owner(s)	RMSMAN
Description	The gards_sample_ratios table contains count ratios of a higher energy ROI to a lower energy ROI specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	ratio_id	*varchar2(15)*	Ratio identifier	N
	upper_roi_number	*number*	Upper ROI number	Y
	lower_roi_number	*number*	Lower ROI number	Y
	count_ratio	*number*	Ratio of the upper ROI counts to lower ROI counts	Y
	count_ratio_err	*number*	Uncertainty of the ratio	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_RATIOS(sample_id, ratio_id)
Foreign keys	FK_SAMPLE_RATIOS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_RATIOS_B_I_U

5.1.169. GARDS_SAMPLE_STATUS

Owner(s)	RMSAUTO
Description	The gards_sample_status table contains spectral processing status.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	entry_date	*date*	Date entered into the IDC database	Y
	cnf_begin_date	*date*	Date when last analysis began	Y
	cnf_end_date	*date*	Date when last analysis ended	Y
	review_date	*date*	Date sample was last reviewed	Y
	review_time	*number*	Amount of time to review sample (minutes)	Y
	analyst	*varchar2(30)*	Name of the analyst who reviewed the sample	Y
I	status	*char(1)*	Current processing status of sample.	Y
	category	*number*	Categorization level	Y
	auto_category	*number*	Sample categorization level after automated analysis	Y
	release_date	*date*	Release date	Y

	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_SAMPLE_STATUS(sample_id)
Foreign keys	FK_SAMPLE_STATUS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_SAMPLE_STATUS_I_U_D, SAMPLE_STATUS_B_I_U

5.1.170. GARDS_SAMPLE_STATUS

Owner(s)	RMSMAN
Description	The gards_sample_status table contains spectral processing status.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	entry_date	*date*	Date entered into the IDC database	Y
	cnf_begin_date	*date*	Date when last analysis began	Y
	cnf_end_date	*date*	Date when last analysis ended	Y
	review_date	*date*	Date sample was last reviewed	Y
	review_time	*number*	Amount of time to review sample (minutes)	Y
	analyst	*varchar2(30)*	Name of the analyst who reviewed the sample	Y
I	status	*char(1)*	Current processing status of sample.	Y
	category	*number*	Categorization level	Y
	auto_category	*number*	Sample categorization level after automated analysis	Y
	release_date	*date*	Release date	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_SAMPLE_STATUS(sample_id)
Foreign keys	FK_SAMPLE_STATUS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_STATUS_UPDATE, SAMPLE_STATUS_B_I_U

5.1.171. GARDS_SAMPLE_TESTS

Owner(s)	RMSMAN
Description	The gards_sample_tests table contains thresholds for sample metrics as used interactively for checking the data quality according to the IDC review policy.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	test_id	*number*	Unique identifier	N
	station_id	*number*	Station identifier	Y

	sample_type	*char(1)*	Sample type	Y
	test_code	*varchar2(24)*	Unique test code identifier string	Y
	test_name	*varchar2(48)*	Test name label for display	Y
	lower_limit_green	*number*	Green flag lower limit	Y
	upper_limit_green	*number*	Green flag upper limit	Y
	lower_limit_yellow	*number*	Yellow flag lower limit	Y
	upper_limit_yellow	*number*	Yellow flag upper limit	Y
	units	*varchar2(16)*	Units of test bounds	Y
	active_flag	*number*	Boolean flag whether it is active	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_SAMPLE_TESTS(test_id)

5.1.172. GARDS_SAMPLE_UPDATE_PARAMS

Owner(s)	RMSAUTO
Description	The gards_sample_update_params table holds the actual parameters used for energy calibration updating during analysis. This table may contain a combination of defaults from gards_update_params_template and values from the command line. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	mrp_used	*number*	Indicator of whether or not the Most Recent Prior (MRP) sample should be used	Y
	mrp_sample_id	*number*	The sample_id of the MRP sample to be used	Y
	gainshift	*number*	Gain change for matching, percent (typically 0.1)	Y
	zeroshift	*number*	Zero change for matching, channels (typically 1)	Y
	area_lim	*number*	Minimum peak area for inclusion in matching (counts)	Y
	use_weight	*number*	0/1 flag for use of weights in ECR updating (typically 1)	Y
	use_mult	*number*	0/1 flag for use of multiplets in ECR updating (typically 1)	Y
	f_linear	*number*	0/1 flag for forcing ECR to be first-order (typically 0)	Y

	bootstrap	*number*	0/1 flag for ignoring prior ECR/RER (typically 0)	Y
	min_lookup	*number*	Minimum library lookup tolerance, keV (typically 0.2)	Y
	rer_intercept	*number*	Default RER intercept (typically 1.2)	Y
	rer_slope	*number*	Default RER coeff (typically 0.03)	Y
	ecr_slope	*number*	Default ECR slope	Y
	do_reru	*number*	Flag that indicates if a resolution update was performed	Y
	rer_mrp_used	*number*	Indicates what RER was chosen during analysis	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_SAMPLE_UPD_PAR(sample_id)
Foreign keys	FK_SAMPLE_UPDATE_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_UPDATE_PARAMS_B_I_U, GARDS_SAMP_UPD_PRMS_I_U_D

5.1.173. GARDS_SAMPLE_UPDATE_PARAMS

Owner(s)	RMSMAN
Description	The gards_sample_update_params table holds the actual parameters used for energy calibration updating during analysis. This table may contain a combination of defaults from gards_update_params_template and values from the command line. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	mrp_used	*number*	Indicator of whether or not the Most Recent Prior (MRP) sample should be used	Y
	mrp_sample_id	*number*	The sample_id of the MRP sample to be used	Y
	gainshift	*number*	Gain change for matching, percent (typically 0.1)	Y
	zeroshift	*number*	Zero change for matching, channels (typically 1)	Y
	area_lim	*number*	Minimum peak area for inclusion in matching (counts)	Y
	use_weight	*number*	0/1 flag for use of weights in ECR updating (typically 1)	Y
	use_mult	*number*	0/1 flag for use of multiplets in ECR updating (typically 1)	Y
	f_linear	*number*	0/1 flag for forcing ECR to be first-order (typically 0)	Y

	bootstrap	*number*	0/1 flag for ignoring prior ECR/RER (typically 0)	Y
	min_lookup	*number*	Minimum library lookup tolerance, keV (typically 0.2)	Y
	rer_intercept	*number*	Default RER intercept (typically 1.2)	Y
	rer_slope	*number*	Default RER coeff (typically 0.03)	Y
	ecr_slope	*number*	Default ECR slope	Y
	do_reru	*number*	Flag that indicates if a resolution update was performed	Y
	rer_mrp_used	*number*	Indicates what RER was chosen during analysis	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_SAMPLE_UPD_PAR(sample_id)
Foreign keys	FK_SAMPLE_UPDATE_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_UPDATE_PARAMS_B_I_U

5.1.174. GARDS_SAMPLE_XE_PROC_PARAMS

Owner(s)	RMSAUTO
Description	The gards_sample_xe_proc_params table contains specific processing parameters to be used for a beta-gamma coincidence spectral analysis. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	lc_abscissa	*number*	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level	Y
	beta_ecr_order	*number*	Order of the beta energy calibration function	Y
	gamma_ecr_order	*number*	Order of the gamma energy calibration function	Y
	compton	*number*	Counts of Compton continuum background	Y
	det_back_used	*number*	Flag indicating whether the detector background is in use or not	Y
	gas_back_used	*number*	Flag indicating whether the gas background is in use or not	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_XE_PROC_PARAMS(sample_id, roi)
Foreign keys	FK_SAMPLE_XE_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_XE_PROC_PARAMS_B_I_U, GARDS_SMPLE_XE_P_PRMS_I_U_D

5.1.175. GARDS_SAMPLE_XE_PROC_PARAMS

Owner(s)	RMSMAN
Description	The gards_sample_xe_proc_params table contains specific processing parameters to be used for a beta-gamma coincidence spectral analysis. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	roi	*number*	ROI identifier	N
	lc_abscissa	*number*	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level	Y
	beta_ecr_order	*number*	Order of the beta energy calibration function	Y
	gamma_ecr_order	*number*	Order of the gamma energy calibration function	Y
	compton	*number*	Counts of Compton continuum background	Y
	det_back_used	*number*	Flag indicating whether the detector background is in use or not	Y
	gas_back_used	*number*	Flag indicating whether the gas background is in use or not	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_SAMPLE_XE_PROC_PARAMS(sample_id, roi)
Foreign keys	FK_SAMPLE_XE_PROC_PARAMS(sample_id, GARDS_SAMPLE_DATA)
Triggers	SAMPLE_XE_PROC_PARAMS_B_I_U

5.1.176. GARDS_SOH_CHAR_DATA

Owner(s)	RMSAUTO
Description	The gards_soh_char_data table contains the state of health (SOH) character data.

Pk/Fk/I	Column	Storage type	Description	Nullable
I	station_id	*number*	Station identifier	N
	detector_id	*number*	Detector identifier	Y
I	param_code	*number*	SOH parameter code	N
I	value	*varchar2(30)*	Value of the specified SOH parameter	N
I	dtg_begin	*date*	Begin date and time	N

I	dtg_end	*date*	End date and time	N
	moddate	*date*	Date/time at which the row was last modified	N
PI	soh_char_id	*number*	Unique state of health character data identifier	N
FI	soh_id	*number*	SOH identifier	N

Primary key	PK_CHAR_DATA(soh_char_id)
Foreign keys	FK_SOH_ID_CHAR_DATA(soh_id, GARDS_SOH_HEADER)
Triggers	SOH_CHAR_DATA_S_I, SOH_CHAR_DATA_B_I_U

5.1.177. GARDS_SOH_CODE

Owner(s)	RMSMAN
Description	The gards_soh_code table contains the state of health (SOH) parameter names and their codes.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	param	*varchar2(30)*	SOH parameter name	N
	param_code	*number*	SOH parameter code	Y
	param_display	*varchar2(30)*	Text of the parameter to be displayed in Trendvue	Y
	param_display_flag	*number*	Flag indicating whether or not a particular parameter is displayed	Y
	display_station	*number*	Flag indicating whether the parameter is associated with a station	Y
	display_detector	*number*	Flag indicating whether the parameter is associated with a detector	Y
	unit	*varchar2(32)*	Unit that the parameter is stored in	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_SOH_CODE(param)
Triggers	SOH_CODE_B_I_U

5.1.178. GARDS_SOH_EVALUATION

Owner(s)	RMSAUTO
Description	The GARDS_SOH_EVALUATION contains results of SOH data evaluated.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(19,0)*	Sample identifier	N
	srid	*varchar2(32)*	Sample reference identification	N

	soh_status	*varchar2(1)*	Processing status	Y
	soh_flag	*varchar2(1)*	SOH flag	Y
	author	*varchar2(30)*	User name	Y
	user_comment	*varchar2(2048)*	Comment	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	lddate	*date*	Load date	Y

Primary key	PK_GSE(sample_id)
Foreign keys	FK_GSE_SID(sample_id, GARDS_SAMPLE_DATA)
Triggers	SOH_EVALUATION_B_I_U, SOH_EVALUATION_B_I, GARDS_SOH_EVALUATION_I_U_D

5.1.179. GARDS_SOH_EVALUATION

Owner(s)	RMSMAN
Description	The GARDS_SOH_EVALUATION contains results of SOH data evaluated.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number(19,0)*	Sample identifier	N
	srid	*varchar2(32)*	Sample reference identification	N
	soh_status	*varchar2(1)*	Processing status	Y
	soh_flag	*varchar2(1)*	SOH flag	Y
	author	*varchar2(30)*	User name	Y
	user_comment	*varchar2(2048)*	Comment	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	lddate	*date*	Load date	Y

Primary key	SYS_C002723457(sample_id)
Foreign keys	FK_BGE_SID(sample_id, GARDS_SAMPLE_DATA)
Triggers	SOH_EVALUATION_B_I_U

5.1.180. GARDS_SOH_HEADER

Owner(s)	RMSAUTO
Description	The gards_soh_header table contains the header information for SOH messages.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	soh_id	*number*	SOH identifier	N

FI	station_id	*number*	Station identifier	N
	detector_id	*number*	Detector identifier	Y
I	dtg_begin	*date*	Begin date and time	N
I	dtg_end	*date*	End date and time	N
	transmit_dtg	*date*	Transmit date and time	N
	moddate	*date*	Date/time at which the row was last modified	N

Primary key	PK_SOH_HEADER(soh_id)
Foreign keys	FK_SOH_HEADER(station_id, GARDS_STATIONS)
Slaves	GARDS_SOH_SENSOR_DATA, GARDS_SOH_CHAR_DATA, GARDS_SOH_NUM_DATA
Triggers	SOH_HEADER_B_I_U

5.1.181. GARDS_SOH_NUM_DATA

Owner(s)	RMSAUTO
Description	The gards_soh_num_data table contains the state of health (SOH) numerical data

Pk/Fk/I	Column	Storage type	Description	Nullable
I	station_id	*number*	Station identifier	N
I	detector_id	*number*	Detector identifier	Y
I	param_code	*number*	SOH parameter code	N
	value	*number*	Value of the specified SOH parameter	N
I	dtg_begin	*date*	Begin date and time	N
I	dtg_end	*date*	End date and time	N
	moddate	*date*	Date/time at which the row was last modified	N
PI	soh_num_id	*number*	State of health identifier	N
FI	soh_id	*number*	SOH identifier	N

Primary key	PK_NUM_DATA(soh_num_id)
Foreign keys	FK_SOH_ID_NUM_DATA(soh_id, GARDS_SOH_HEADER)
Triggers	SOH_NUM_DATA_B_I_U, SOH_NUM_DATA_S_I

5.1.182. GARDS_SOH_SENSOR_DATA

Owner(s)	RMSAUTO
Description	The gards_soh_sensor_data table contains the data of sensors

Pk/Fk/I	Column	Storage type	Description	Nullable
I	station_id	*number*	Station identifier	N

	detector_id	*number*	Detector identifier	Y
I	sensor_type	*varchar2(20)*	Sensor type	N
I	sensor_name	*varchar2(20)*	Unique sensor name to distinguish between sensors of the same type	N
	value	*number*	Value of the specified SOH parameter	N
I	dtg_begin	*date*	Begin date and time	N
I	dtg_end	*date*	End date and time	N
	moddate	*date*	Date/time at which the row was last modified	N
PI	soh_sensor_id	*number*	State of health sensor id	N
FI	soh_id	*number*	SOH identifier	N

Primary key	PK_SENSOR_DATA(soh_sensor_id)
Foreign keys	FK_SOH_ID_SENSOR_DATA(soh_id, GARDS_SOH_HEADER)
Triggers	SOH_SENSOR_DATA_S_I, SOH_SENSOR_DATA_B_I_U

5.1.183. GARDS_SPECTRUM

Owner(s)	RMSAUTO
Description	The gards_spectrum table contains information regarding the spectral data information reported in the PHD file.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	sample_type	*char(1)*	Sample type	N
	filename	*varchar2(256)*	Name of the file containing the spectrum data	Y
	channels	*number*	Number of channels in spectrum	Y
	energy_span	*number*	Total energy span of detector (keV)	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	start_channel	*number*	Start channel	Y

Primary key	PK_GARDS_SPECTRUM(sample_id, sample_type)
Foreign keys	FK_SPECTRUM(sample_id, GARDS_SAMPLE_DATA)
Triggers	SPECTRUM_B_I_U

5.1.184. GARDS_STADET

Owner(s)	RMSMAN
Description	The gards_stadet table contains a complete list of station/detector identifier code combinations. The entries in this table are generated via a trigger on the gards_sample_data table.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	station_id	*number*	Station identifier	N
PFI	detector_id	*number*	Detector identifier	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_STADET(station_id, detector_id)
Foreign keys	FK_STADET_STA(station_id, GARDS_STATIONS), FK_STADET_DET(detector_id, GARDS_DETECTORS)
Indexes	PK_STADET(station_id, detector_id), GARDS_STADET_DID_NDX(detector_id)
Triggers	STADET_B_I_U

5.1.185. GARDS_STATIONS

Owner(s)	RMSMAN
Description	The gards_stations table contains a station overview and station characteristics.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	station_id	*number*	Station identifier	N
I	station_code	*varchar2(5)*	Station code	Y
	country_code	*varchar2(2)*	Country code	Y
	type	*varchar2(6)*	Station type	Y
	description	*varchar2(40)*	Station description	Y
	lat	*number*	Latitude (degrees)	Y
	lon	*number*	Longitude (degrees)	Y
	elevation	*number*	Elevation (m)	Y
	date_begin	*date*	Initialization date	Y
	date_end	*date*	Decommissioning date	Y
	status	*varchar2(2)*	Status of station, if NULL then station is fully operational otherwise the status is I	Y
	pocid	*number*	Point of contact identifier	Y
	split_station	*number*	Flag indicating whether or not station is split	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_STATIONS(station_id)
Slaves	GARDS_PROC_PARAMS_TEMPLATE, GARDS_UPDATE_REFLINES, GARDS_MET_DATA, GARDS_MET_HEADER, GARDS_SOH_HEADER, GARDS_STATIONS_SCHEDULE, GARDS_QCTARGETS, GARDS_STADET, GARDS_CAT_TEMPLATE, GARDS_STATION_ASSIGNMENTS
Triggers	STATIONS_INSERT, STATIONS_B_I_U

5.1.186. GARDS_STATIONS_SCHEDULE

Owner(s)	RMSMAN
Description	The gards_stations_schedule table contains schedule descriptions for stations reporting to a data center.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	station_id	*number*	Station identifier	N
PI	begin_date	*date*	Initialization date	N
	end_date	*date*	Decommissioning date	Y
PI	code	*char(5)*	Code symbol SCHEDULE as found in the gards_codes table. SF_GP for full sample grace period or SPHDF for full sample PHD	N
	temporal_value	*number*	Expected station schedule period	Y
	temporal_unit	*char(5)*	Units for schedule period - currently only the value DAYS is supported	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_STATIONS_SCHED(station_id, code, begin_date)
Foreign keys	FK_STA_SCHEDULE_STAID(station_id, GARDS_STATIONS)
Triggers	STATIONS_SCHEDULE_B_I_U

5.1.187. GARDS_STATION_ASSIGNMENTS

Owner(s)	RMSMAN
Description	The gards_station_assignments table contains a mapping of which stations are assigned to which users.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	station_id	*number*	Station identifier	N
FI	user_id	*number*	User identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RMSMAN_GARDS_STATION_ASS(station_id)
Foreign keys	FK_STA_ASSIGN_STAID(station_id, GARDS_STATIONS), FK_STA_ASSIGN_USRID(user_id, GARDS_USERS)
Triggers	STATION_ASSIGNMENTS_B_I_U

5.1.188. GARDS_STATUS_HISTORY

Owner(s)	RMSMAN
Description	The gards_status_history table contains a list of changes in gards_sample_status for

	the reviewed spectrum.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	sample_id	*number*	Sample identifier	Y
	user_name	*varchar2(30)*	User name	Y
	old_status	*char(1)*	Sample status before change	Y
	new_status	*char(1)*	Sample status after change	Y
	dtg	*date*	Entry date and time	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Foreign keys	FK_STATUS_HISTORY(sample_id, GARDS_SAMPLE_DATA)
Triggers	STATUS_HISTORY_B_I_U

5.1.189. GARDS_TOTAL_EFFIC

Owner(s)	RMSAUTO
Description	The gards_total_effic table contains the total efficiency data as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	effic_energy	*number*	Gamma energy	N
	efficiency	*number*	Efficiency	Y
	effic_error	*number*	Uncertainty of the efficiency	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_TOTAL_EFFIC(sample_id, effic_energy)
Foreign keys	FK_TOTAL_EFFIC(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_TOTAL_EFFIC_I_U_D, TOTAL_EFFIC_B_I_U

5.1.190. GARDS_TOTAL_EFFIC

Owner(s)	RMSMAN
Description	The gards_total_effic table contains the total efficiency data as specified in the PHD.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	effic_energy	*number*	Gamma energy	N
	efficiency	*number*	Efficiency	Y
	effic_error	*number*	Uncertainty of the efficiency	Y
	moddate	*date*	Date/time at which the row was last modi-	Y

			fied	

Primary key	PK_GARDS_TOTAL_EFFIC(sample_id, effic_energy)
Foreign keys	FK_TOTAL_EFFIC(sample_id, GARDS_SAMPLE_DATA)
Triggers	TOTAL_EFFIC_B_I_U

5.1.191. GARDS_TRENDVUE

Owner(s)	RMSAUTO, RMSMAN
Description	The gards_trendvue table contains the data produced by the Trendvue application and is purged at the end of each session. (Note: it is obsoleted.)

Pk/Fk/I	Column	Storage type	Description	Nullable
	session_id	*number*	Unique session identifier	Y
	date_value	*date*	Date of the Trendvue plot	Y
	num_value1	*number*	First plot value	Y
	num_value2	*number*	Second plot value	Y
	num_value3	*number*	Third plot value	Y
	num_value4	*number*	Fourth plot value	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Triggers	TRENDVUE_B_I_U

5.1.192. GARDS_UPDATE_PARAMS_TEMPLATE

Owner(s)	RMSMAN
Description	The gards_update_params_template table holds default parameters used for energy calibration updating during analysis; these values can be overridden at the command line.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	detector_id	*number*	Detector identifier	N
	gainshift	*number*	Gain change for matching, percent (typically 0.1)	Y
	zeroshift	*number*	Zero change for matching, channels (typically 1)	Y
	area_lim	*number*	Minimum peak area for inclusion in matching (counts)	Y
	use_weight	*number*	0/1 flag for use of weights in ECR updating (typically 1)	Y
	use_mult	*number*	0/1 flag for use of multiplets in ECR updating (typically 1)	Y
	f_linear	*number*	0/1 flag for forcing ECR to be first-order	Y

			(typically 0)	
	bootstrap	*number*	0/1 flag for ignoring prior ECR/RER (typically 0)	Y
	min_lookup	*number*	Minimum library lookup tolerance, keV (typically 0.2)	Y
	rer_intercept	*number*	Default RER intercept (typically 1.2)	Y
	rer_slope	*number*	Default RER slope (typically 0.03)	Y
	do_reru	*number*	Flag that indicates if a resolution update was performed	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_UPDATE_PAR_TEMPL(detector_id)
Foreign keys	FK_UPD_PARAMS_TMPL(detector_id, GARDS_DETECTORS)
Triggers	UPDATE_PARAMS_TEMPLATE_B_I_U

5.1.193. GARDS_UPDATE_REFLINES

Owner(s)	RMSMAN
Description	The gards_update_reflines table contains energies of reference lines used in ECR/RER updating functions.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	station_id	*number*	Station identifier	N
PFI	detector_id	*number*	Detector identifier	N
PI	refpeak_energy	*number*	Reference energy of the decay line	N
PI	data_type	*char(1)*	Data type of the spectrum	N
PI	spectral_qualifier	*varchar2(5)*	Indicator of full or preliminary spectrum	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_UPDATE_REFLINES(station_id, detector_id, refpeak_energy, data_type, spectral_qualifier)
Foreign keys	FK_UPD_REFLINES_STA(station_id, GARDS_STATIONS), FK_UPD_REFLINES_DET(detector_id, GARDS_DETECTORS)
Triggers	UPDATE_REFLINES_B_I_U

5.1.194. GARDS_USERENV

Owner(s)	RMSMAN
Description	The gards_userenv table contains configurable environment variables used by the Automatic Processing software.

Pk/Fk/I	Column	Storage type	Description	Nullable

PI	name	*varchar2(40)*	Variable name	N
	value	*varchar2(256)*	Variable value	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_USERENV(name)
Triggers	USERENV_B_I_U

5.1.195. GARDS_USERS

Owner(s)	RMSMAN
Description	Contains details of radionuclide-related personnel.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	user_id	*number*	User identifier	N
	user_name	*varchar2(30)*	User name	N
	assignable	*number*	Indicator of whether or not the user can be assigned stations	N
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_USERS(user_id)
Indexes	PK_USERS(user_id)
Slaves	GARDS_USERS_ROLES, GARDS_STATION_ASSIGNMENTS
Triggers	GARDS_USER_DELETE, USERS_B_I_U

5.1.196. GARDS_USERS_ROLES

Owner(s)	RMSMAN
Description	The gards_users_roles table contains a mapping of which roles are assigned to which user.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	user_id	*number*	User identifier	N
PFI	role_id	*number*	Role identifier	N
	default_role	*number*	Integer representation of the default role name to be assigned to the user	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_USERS_ROLES(user_id, role_id)
Foreign keys	FK_USER_ID(user_id, GARDS_USERS), FK_ROLE_ID(role_id, GARDS_ROLES)
Triggers	USERS_ROLES_B_I_U

5.1.197. GARDS_USER_COMMENTS

Owner(s)	RMSMAN
Description	The gards_user_comments table contains the text associated with user-defined comments.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	sample_id	*number*	Sample identifier	Y
PFI	comment_id	*number*	Comment identifier	N
	comment_text	*varchar2(2048)*	Comment text	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_USER_COMMENTS(comment_id)
Foreign keys	FK_USER_COMMENTS_SID(sample_id, GARDS_SAMPLE_DATA), FK_USER_COMMENTS(comment_id, GARDS_COMMENTS)
Triggers	USER_COMMENTS_B_I_U

5.1.198. GARDS_USER_COMMENTS

Owner(s)	RMSAUTO
Description	The gards_user_comments table contains the text associated with user-defined comments.

Pk/Fk/I	Column	Storage type	Description	Nullable
FI	sample_id	*number*	Sample identifier	Y
PFI	comment_id	*number*	Comment identifier	N
	comment_text	*varchar2(2048)*	Comment text	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_USER_COMMENTS(comment_id)
Foreign keys	FK_USER_COMMENTS_SID(sample_id, GARDS_SAMPLE_DATA), FK_USER_COMMENTS(comment_id, GARDS_COMMENTS)
Triggers	USER_COMMENTS_B_I_U, GARDS_USER_COMMENTS_I_U_D

5.1.199. GARDS_XE_CATEGORY_PARAMS

Owner(s)	RMSMAN
Description	The GARDS_XE_CATEGORY_PARAMS contains parameters used for categorization of the noble gas sample.

Pk/Fk/I	Column	Storage type	Description	Nullable

PI	start_date	*date*	Initialization date	N
	end_date	*date*	Decommissioning date	Y
	moving_avg_days	*number*	Period of the inter-quartile algorithm used for abnormal threshold determination for noble gas sample categorization	Y
	abnormal_conc_factor	*number*	Factor used in the algorithm of the abnormal threshold	Y
	xe133m_xe131m_limit	*number*	Ratio limit of Xe-133m to Xe-131m	Y
	xe133m_xe133_limit	*number*	Ratio limit of Xe-133m to Xe-133	Y
	xe135_xe133_limit	*number*	Ratio limit of Xe-135 to Xe-133	Y
	bayes_plus	*number*	Percentile related to the upper bound of Bayesian ratio	Y
	bayes_minus	*number*	Percentile related to the lower bound of Bayesian ratio	Y
	analyst	*varchar2(30)*	Name of the analyst who reviewed the sample	Y
	comment_txt	*varchar2(300)*	Free-form user comments	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	shortterm_moving_avg_days	*number*	period in which the average activity concentration is determined for a short term abnormal threshold determination	Y
	xe133_xe131m_limit	*number*	Ratio limit of Xe-133 to Xe-131m	Y

Primary key	GARDS_XE_CAT_PAR_PK(start_date)

5.1.200. GARDS_XE_NUCL_LIB

Owner(s)	RMSMAN
Description	The gards_xe_nucl_lib table contains nuclide library for noble gas analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
	nuclide_id	*number*	Nuclide identifier	Y
PI	name	*varchar2(8)*	Nuclide name	N
	type	*varchar2(16)*	Nuclide type	Y
	halflife_sec	*number*	Half-life in seconds	Y
	halflife	*varchar2(23)*	Half-life	Y
	halflife_err	*varchar2(23)*	Uncertainty of half-life	Y
	num_lines	*number*	Number of decay lines in library	Y

	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_XE_NUCL_LIB(name)
Triggers	XE_NUCL_LIB_B_I_U

5.1.201. GARDS_XE_NUCL_LINES_LIB

Owner(s)	RMSMAN
Description	The gards_xe_nucl_lines_lib table contains nuclide library of energy lines for noble gas analysis.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	name	*varchar2(8)*	Nuclide name	N
PI	energy	*number*	Energy of the decay line	N
	energy_err	*number*	Uncertainty of the energy	Y
	abundance	*number*	Emission intensity per disintegration	Y
	abundance_err	*number*	Uncertainty of the abundance	Y
	key_flag	*number*	Key line indicator	Y
	nuclide_id	*number*	Nuclide identifier	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_XE_NUCL_LINES_LIB(name, energy)
Triggers	XE_NUCL_LINES_LIB_B_I_U

5.1.202. GARDS_XE_PROC_PARAMS_TEMPLATE

Owner(s)	RMSMAN
Description	The gards_xe_proc_params_template table contains the information used for noble gas spectra analysis. (Note: it is not used in the current IDC analysis.)

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	detector_id	*number*	Detector identifier	N
PI	roi	*number*	ROI identifier	N
	lc_abscissa	*number*	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level	Y
	beta_ecr_order	*number*	Order of the beta energy calibration function	Y
	gamma_ecr_order	*number*	Order of the gamma energy calibration function	Y
	compton	*number*	Counts of Compton continuum background	Y
	det_back_used	*number*	Flag indicating whether the detector back-	Y

			ground is in use or not	
	gas_back_used	*number*	Flag indicating whether the gas background is in use or not	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_XE_PROC_PARAMS_TEMP(detector_id, roi)
Foreign keys	XE_PPT_DET(detector_id, GARDS_DETECTORS)
Triggers	XE_PROC_PARAMS_TEMPLATE_B_I_U

5.1.203. GARDS_XE_REFLINE_MASTER

Owner(s)	RMSMAN
Description	The GARDS_XE_REFLINE_MASTER table contains the reference peak information for updating energy and resolution calibrations of HPGe based noble gas spectra (SPALAX).

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	refpeak_energy	*number*	Reference energy of the decay line	N
PI	data_type	*char(1)*	Data type of the spectrum	N
PI	spectral_qualifier	*varchar2(5)*	Indicator of full or preliminary spectrum	N
	calibration_type	*varchar2(3)*	Calibration type	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_GARDS_XE_REFLINE_MASTER(refpeak_energy, data_type, spectral_qualifier)
Triggers	XE_REFLINE_MASTER_B_I_U

5.1.204. GARDS_XE_RESULTS

Owner(s)	RMSAUTO
Description	The GARDS_XE_RESULTS table contains results of gamma spectrum analysis for noble gas systems with high resolution.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	method_id	*number*	Method identifier	N
PI	nuclide_id	*number*	Nuclide identifier	N
	conc	*float(126)*	Activity concentration	Y
	conc_err	*float(126)*	Uncertainty of the activity concentration	Y
	mdc	*float(126)*	Minimum detectable concentration	Y
	mdi	*float(126)*	Minimum detectable intensity for the meta-	Y

			stable radioxenon	
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	lc	*float(126)*	Critical level in activity concentration	Y
	ld	*float(126)*	Detection limit in activity concentration	Y
	sample_act	*float(126)*	Spectrum activity without decay	Y
	cov_xe_131m	*float(126)*	Covariance matrix line 1 for Xe-131m	Y
	cov_xe_133m	*float(126)*	Covariance matrix line 2 for Xe-133m	Y
	cov_xe_133	*float(126)*	Covariance matrix line 3 for Xe-133	Y
	cov_xe_135	*float(126)*	Covariance matrix line 4 for Xe-135	Y
	cov_radon	*float(126)*	Covariance matrix line 5 for radon	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	activity_acq_start	*float(126)*	Activity at the start of acquisition	Y
	activity_err_acq_start	*float(126)*	Uncertainty of the activity at the start of acquisition	Y
	mda_acq_start	*float(126)*	Minimum detectable activity at the start of acquisition	Y

Primary key	PK_GARDS_XE_RESULTS(sample_id, method_id, nuclide_id)
Foreign keys	FK_XE_RESULTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_XE_RESULTS_I_U_D, XE_RESULTS_B_I_U

5.1.205. GARDS_XE_RESULTS

Owner(s)	RMSMAN
Description	The GARDS_XE_RESULTS table contains results of gamma spectrum analysis for noble gas systems with high resolution.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	method_id	*number*	Method identifier	N
PI	nuclide_id	*number*	Nuclide identifier	N
	conc	*float(126)*	Activity concentration	Y
	conc_err	*float(126)*	Uncertainty of the activity concentration	Y
	mdc	*float(126)*	Minimum detectable concentration	Y
	mdi	*float(126)*	Minimum detectable intensity for the metastable radioxenon	Y
	nid_flag	*number*	Flag of whether the xenon isotope is detected or not	Y
	lc	*float(126)*	Critical level in activity concentration	Y

	ld	*float(126)*	Detection limit in activity concentration	Y
	sample_act	*float(126)*	Spectrum activity without decay	Y
	cov_xe_131m	*float(126)*	Covariance matrix line 1 for Xe-131m	Y
	cov_xe_133m	*float(126)*	Covariance matrix line 2 for Xe-133m	Y
	cov_xe_133	*float(126)*	Covariance matrix line 3 for Xe-133	Y
	cov_xe_135	*float(126)*	Covariance matrix line 4 for Xe-135	Y
	cov_radon	*float(126)*	Covariance matrix line 5 for radon	Y
	moddate	*date*	Date/time at which the row was last modified	Y
	activity_acq_start	*float(126)*	Activity at the start of acquisition	Y
	activity_err_acq_start	*float(126)*	Uncertainty of the activity at the start of acquisition	Y
	mda_acq_start	*float(126)*	Minimum detectable activity at the start of acquisition	Y

Primary key	PK_GARDS_XE_RESULTS(sample_id, method_id, nuclide_id)
Foreign keys	FK_XE_RESULTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	XE_RESULTS_B_I_U

5.1.206. GARDS_XE_SAMPLE_CATEGORY

Owner(s)	RMSMAN
Description	The GARDS_XE_SAMPLE_CATEGORY contains results of categorization of the noble gas sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
	prev_valid_samples	*number*	Number of the previous valid samples	Y
	abn_xe131m	*number*	Abnormal threshold of the activity concentration for Xe-131m	Y
	abn_xe133m	*number*	Abnormal threshold of the activity concentration for Xe-133m	Y
	abn_xe133	*number*	Abnormal threshold of the activity concentration for Xe-133	Y
	abn_xe135	*number*	Abnormal threshold of the activity concentration for Xe-135	Y
	median_xe131m	*number*	Median of the activity concentration for Xe-131m	Y
	median_xe133m	*number*	Median of the activity concentration for Xe-133m	Y
	median_xe133	*number*	Median of the activity concentration for Xe-133	Y

	median_xe135	*number*	Median of the activity concentration for Xe-135	Y
	bayes_133m_131m	*number*	Bayesian minus ratio of Xe-133m to Xe-131m	Y
	bayes_133m_133	*number*	Bayesian minus ratio of Xe-133m to Xe-133	Y
	bayes_135_133	*number*	Bayesian minus ratio of Xe-135 to Xe-133	Y
	cat_xe131m	*number*	Categorization level for Xe-131m	Y
	cat_xe133m	*number*	Categorization level for Xe-133m	Y
	cat_xe133	*number*	Categorization level for Xe-133	Y
	cat_xe135	*number*	Categorization level for Xe-135	Y
	flag_133m_131m	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133m to Xe-131m	Y
	flag_133m_133	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133m to Xe-133	Y
	flag_135_133	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-135 to Xe-133	Y
	cat_sample	*number*	Categorization level of the sample	Y
	flag_atm	*number*	Flag whether or not a known source would be detected in the sample, which is determined using ATM simulations.	Y
	moddate	*date*	Date/time at which the row was last modified	Y
PFI	flag_shortterm	*number*	Flag whether or not the categorization is determined using the short term approach	N
	conc_133m_131m	*number*	Ratio of activity concentrations for Xe-133m to Xe-131m	Y
	conc_133m_133	*number*	Ratio of activity concentrations for Xe-133m to Xe-133	Y
	conc_135_133	*number*	Ratio of activity concentrations for Xe-135 to Xe-133	Y
	conc_133_131m_mdc	*number*	Ratio of activity concentrations for Xe-133 to Xe-131m, where the Xe-131m MDC is used when Xe-131m is not detected.	Y
	bayes_133_131m	*number*	Bayesian minus ratio of Xe-133 to Xe-131m	Y
	flag_133_131m	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133 to Xe-131m	Y
	conc_133_131m	*number*	Ratio of activity concentrations for Xe-133 to Xe-131m	Y

Primary key	PK_SAMPLE_ID(sample_id, flag_shortterm)
Foreign keys	FK_XESAMPLECAT(flag_shortterm, GARDS_XE_SAMPLE_CATEGORY), FK_XESAMPLECAT(sample_id, GARDS_XE_SAMPLE_CATEGORY)
Slaves	GARDS_XE_SAMPLE_CATEGORY

5.1.207. GARDS_XE_SAMPLE_CATEGORY

Owner(s)	RMSAUTO
Description	The GARDS_XE_SAMPLE_CATEGORY contains results of categorization of the noble gas sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	sample_id	*number*	Sample identifier	N
	prev_valid_samples	*number*	Number of the previous valid samples	Y
	abn_xe131m	*number*	Abnormal threshold of the activity concentration for Xe-131m	Y
	abn_xe133m	*number*	Abnormal threshold of the activity concentration for Xe-133m	Y
	abn_xe133	*number*	Abnormal threshold of the activity concentration for Xe-133	Y
	abn_xe135	*number*	Abnormal threshold of the activity concentration for Xe-135	Y
	median_xe131m	*number*	Median of the activity concentration for Xe-131m	Y
	median_xe133m	*number*	Median of the activity concentration for Xe-133m	Y
	median_xe133	*number*	Median of the activity concentration for Xe-133	Y
	median_xe135	*number*	Median of the activity concentration for Xe-135	Y
	bayes_133m_131m	*number*	Bayesian minus ratio of Xe-133m to Xe-131m	Y
	bayes_133m_133	*number*	Bayesian minus ratio of Xe-133m to Xe-133	Y
	bayes_135_133	*number*	Bayesian minus ratio of Xe-135 to Xe-133	Y
	cat_xe131m	*number*	Categorization level for Xe-131m	Y
	cat_xe133m	*number*	Categorization level for Xe-133m	Y
	cat_xe133	*number*	Categorization level for Xe-133	Y
	cat_xe135	*number*	Categorization level for Xe-135	Y
	flag_133m_131m	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133m to Xe-131m	Y
	flag_133m_133	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133m to Xe-133	Y
	flag_135_133	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-135 to Xe-133	Y
	cat_sample	*number*	Categorization level of the sample	Y
	flag_atm	*number*	Flag whether or not a known source would be detected in the sample, which is determ-	Y

			ined using ATM simulations.	
	moddate	*date*	Date/time at which the row was last modified	Y
PI	flag_shortterm	*number*	Flag whether or not the categorization is determined using the short term approach	N
	conc_133m_131m	*number*	Ratio of activity concentrations for Xe-133m to Xe-131m	Y
	conc_133m_133	*number*	Ratio of activity concentrations for Xe-133m to Xe-133	Y
	conc_135_133	*number*	Ratio of activity concentrations for Xe-135 to Xe-133	Y
	conc_133_131m_mdc	*number*	Ratio of activity concentrations for Xe-133 to Xe-131m, where the Xe-131m MDC is used when Xe-131m is not detected.	Y
	bayes_133_131m	*number*	Bayesian minus ratio of Xe-133 to Xe-131m	Y
	flag_133_131m	*number*	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133 to Xe-131m	Y
	conc_133_131m	*number*	Ratio of activity concentrations for Xe-133 to Xe-131m	Y

Primary key	PK_SAMPLE_ID(sample_id, flag_shortterm)
Slaves	GARDS_XE_SAMPLE_CATEGORY
Triggers	GARDS_XE_SAMPLE_CATEGORY_I_U

5.1.208. GARDS_XE_UNCORRECTED_RESULTS

Owner(s)	RMSMAN
Description	Contains analysis results without decay corrections for the noble gas sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	method_id	*number*	Method identifier	N
PI	nuclide_id	*number*	Nuclide identifier	N
	conc	*number*	Activity concentration	Y
	conc_error	*number*	Uncertainty of the activity concentration	Y
	mdc	*number*	Minimum detectable concentration	Y
	mdi	*number*	Minimum detectable intensity for the metastable radioxenon	Y
	lc	*number*	Critical level in activity concentration	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_XE_UNCORRECTED_RESULTS(sample_id, method_id, nuclide_id)
Foreign keys	FK_XE_UNCORRECTED_RESULTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_XE_UNCORR_RES_A_I_U

5.1.209. GARDS_XE_UNCORRECTED_RESULTS

Owner(s)	RMSAUTO
Description	Contains analysis results without decay corrections for the noble gas sample.

Pk/Fk/I	Column	Storage type	Description	Nullable
PFI	sample_id	*number*	Sample identifier	N
PI	method_id	*number*	Method identifier	N
PI	nuclide_id	*number*	Nuclide identifier	N
	conc	*number*	Activity concentration	Y
	conc_error	*number*	Uncertainty of the activity concentration	Y
	mdc	*number*	Minimum detectable concentration	Y
	mdi	*number*	Minimum detectable intensity for the meta-stable radioxenon	Y
	lc	*number*	Critical level in activity concentration	Y
	moddate	*date*	Date/time at which the row was last modi-fied	Y

Primary key	PK_XE_UNCORRECTED_RESULTS(sample_id, method_id, nuclide_id)
Foreign keys	FK_XE_UNCORRECTED_RESULTS(sample_id, GARDS_SAMPLE_DATA)
Triggers	GARDS_XE_UNCORR_RES_I_U_D, GARDS_XE_UNCORR_RES_A_I_U

5.1.210. NMS_RECEIVED_SOFAR

Owner(s)	RMSAUTO
Description	Helper table of RMS processing script to check new messages. It is a lookup table that stores latest nms_msgid fetched from acq.acq_msg table.

Pk/Fk/I	Column	Storage type	Description	Nullable
	nms_msgid	*number*	Message identifier	Y
	creation_time	*date*	Latest access time	Y

5.1.211. PIPELINE_CONTROL

Owner(s)	RMSMAN
Description	Control table for RN message processing. It holds configuration parameters needed to control the behaviour of the RnMessagesProcessor.pl program

Pk/Fk/I	Column	Storage type	Description	Nullable
	stop_acquisition	*char(1)*	Y to stop/exit program RnMessagesProcessor.pl	Y
	stop_pipeline	*char(1)*	Y to stop calling rms_pipeline but the program will continue read messages from acq.acq_msg table	Y
	max_running	*number*	Determines the maximum number of RUNNING massages that the program can handle at any time	Y
	sleep_time	*number*	Determines the sleep time in seconds before processing RECEIVED messages	Y
	instance_running	*char(1)*	Running flag	Y
	instance_info	*varchar2(30)*	Determines the current instance information	Y
	instance_time	*date*	Determines when the instance started	Y
	scan_msgtsd	*char(1)*	Scan flag	Y
	nms_pull_frequency	*number*	Time in seconds between reading RN messages from acq.acq_msg table	Y
	pull_method	*varchar2(15)*	Currently is DB	Y
	pull_host	*varchar2(15)*	Database name to be connected in order to read RN messages	Y
	pull_connection	*varchar2(30)*	User name and password for DB connection	Y
	pull_owner_or_queue	*varchar2(30)*	The user name who owns acq_msg table (default Acq)	Y

5.1.212. RMS_MSGDISC

Owner(s)	RMSAUTO
Description	The rms_msgdisc table holds information about new RN messages read from acq.ac_msg table and necessary related information needed by rms_pipeline script in order to process the RN messages. Also, it contains status of processing as well as authentication Status.

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	msgid	*number*	Message identifier	N
	nms_msgid	*varchar2(50)*	Message identifier	Y
	sta_msgid	*varchar2(50)*	Message identifier	Y
	dir	*varchar2(100)*	Full path to the data file	Y
	filename	*varchar2(100)*	Input data filename	Y
	station_code	*varchar2(10)*	Station code	Y
	data_type	*varchar2(30)*	Type of the data message	Y
	data_offset	*number*	Offset of the data type within the RN message	Y

	data_size	*number*	Size of the data type in the RN message	Y
I	status	*varchar2(15)*	Status of the RN message. Values: SKIPPED, RECEIVED, QUEUED, RUNNING, L-RUNNING, DONE, PARTIAL_DONE or FAILED.	Y
I	lddate	*date*	Load date	Y
I	moddate	*date*	Date/time at which the row was last modified	Y
	subject	*varchar2(64)*	Subject of the RN message extracted of the mail header of the message	Y
	auth_status	*varchar2(50)*	Authentication status of the associated message	Y

Primary key	PK_RMS_MSGDISC(msgid)
Indexes	PK_RMS_MSGDISC(msgid), RMS_MSGDISC_NDX(status, msgid), RMS_MSGDISC_LDDATE_NDX(lddate), RMS_MSGDISC_MOD_NDX(moddate)

5.1.213. RN_LOGS

Owner(s)	RMSMAN
Description	Contains related info for analyst comments from rms_log

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	comment_id	*number*	Comment identifier	N
	analyst	*varchar2(15)*	Name of the analyst who reviewed the sample	Y
	sample_id	*number*	Sample identifier	Y
	station_code	*varchar2(20)*	Station code	Y
	comment_type	*number*	Comment code	Y
	severity	*number*	Severity level null, or between 1 and 4 for this analyst comment	Y
	ref_id	*number*	Reference to related analyst comment	Y
	ext_ref_id	*number*	External reference to related analyst comment, not used	Y
	comment_date	*date*	Date of issuing the analyst comment	Y
	begin_date	*date*	Initialization date	Y
	end_date	*date*	Decommissioning date	Y
	subject	*varchar2(50)*	Subject of the analyst comment	Y
	notify_list	*varchar2(50)*	List of notification recipients for this analyst comment.	Y
	moddate	*date*	Date/time at which the row was last modified	Y

Primary key	PK_RN_LOGS(comment_id)
Indexes	PK_RN_LOGS(comment_id)

5.1.214. RN_LOGS_TXT

Owner(s)	RMSMAN
Description	Contains analyst comments from rms_log

Pk/Fk/I	Column	Storage type	Description	Nullable
PI	comment_id	*number*	Comment identifier	N
	comment_text	*varchar2(4000)*	Comment text	Y

Primary key	PK_RN_LOGS_TXT(comment_id)
Indexes	PK_RN_LOGS_TXT(comment_id)

5.2. Radionuclide Column Descriptions

5.2.1. abn_xe131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Abnormal threshold of the activity concentration for Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.2. abn_xe133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Abnormal threshold of the activity concentration for Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.3. abn_xe133m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Abnormal threshold of the activity concentration for Xe-133m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.4. abn_xe135

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Abnormal threshold of the activity concentration for Xe-135			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.5. abnormal_conc_factor

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)
Description	Factor used in the algorithm of the abnormal threshold

Storage type	Unit	NA value	External	Nullable
number			g8.5	Y
Range				

5.2.6. *abscissa*

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Acceptable number of standard deviations away from the central_value where the amount of a nuclide may be found (RDC algorithm)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.7. *abundance*

Table(s)	GARDS_BG_PROC_PARAMS_ROI (RMSAUTO), GARDS_BG_PROC_PARAMS_ROI (RMSMAN)			
Description	Emission intensity (%) per disintegration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	percent		g8.5	Y
Range				

5.2.8. *abundance*

Table(s)	GARDS_CSC_MODCOEFF_LIB (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_ROI_LIB (RMSMAN), GARDS_SAMPLE_CERT_LINES (RMSAUTO), GARDS_XE_NUCL_LINES_LIB (RMSMAN)			
Description	Emission intensity (%) per disintegration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		g8.5	Y
Range				

5.2.9. *abundance_act*

Table(s)	GARDS_AUX_LINES_LIB (RMSMAN)			
Description	Emission intensity (%) per disintegration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		g8.5	N
Range				

5.2.10. abundance_act_err

Table(s)	GARDS_AUX_LINES_LIB (RMSMAN)			
Description	Uncertainty of the abundance_act			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		g8.5	Y
Range				

5.2.11. abundance_err

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_ROI_LIB (RMSMAN), GARDS_XE_NUCL_LINES_LIB (RMSMAN)			
Description	Uncertainty of the abundance			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		g8.5	Y
Range				

5.2.12. acquisition_live_sec

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Active detection time (seconds)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	seconds		g8.5	Y
Range				

5.2.13. acquisition_real_sec

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Total time between the acquisition stop and acquisition start (seconds)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	seconds		g8.5	Y
Range				

5.2.14. acquisition_start

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)
Description	Acquisition start date and time

Storage type	Unit	NA value	External	Nullable
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.15. *acquisition_stop*

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Acquisition stop date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.16. *activ_confidence_level*

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Level of confidence			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f5.2	Y
Range				

5.2.17. *activ_coverage_factor*

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Coverage factor			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.18. *activ_decay*

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Activity decay factor			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.19. *activ_decay_err*

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)

Description	Uncertainty of activity decay factor			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.20. activ_err

Table(s)	GARDS_ROI_CONCS (RMSAUTO), GARDS_ROI_CONCS (RMSMAN)			
Description	Uncertainty of the activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.21. activ_err

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the activity concentration (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.22. activ_key

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Activity concentration of the key line (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.23. activ_key_acq_start

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Activity of the key line at the start of acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq		g8.5	Y
Range				

5.2.24. activ_key_err

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the activity concentration (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.25. activ_key_err_acq_start

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the activity of the key line at the start of acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq		g8.5	Y
Range				

5.2.26. active_flag

Table(s)	GARDS_CAT_CRITERIA_TESTS (RMSMAN)			
Description	0 if test is inactive			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.27. active_flag

Table(s)	GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Boolean flag whether it is active			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.28. activity

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Activity at the start of acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Bq		e11.4	N
Range				

5.2.29. activity

Table(s)	GARDS_SAMPLE_CERT_LINES (RMSAUTO)			
Description	Activity at time of assay			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Bq		g8.5	Y
Range				

5.2.30. activity

Table(s)	GARDS_ROI_CONCS (RMSAUTO), GARDS_ROI_CONCS (RMSMAN)			
Description	Activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.31. activity

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Activity concentration (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.32. activity

Table(s)	GARDS_SAMPLE_CAT (RMSMAN)			
Description	Activity concentration of the nuclide detected (uBq/m3), related to gards_nucl_ided.activ_key			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.33. activity

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.34. activity_acq_start

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Activity at the start of acquisition (mBq)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq		g8.5	Y
Range				

5.2.35. activity_date

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Activity reference date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.36. activity_err

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Relative uncertainty of the activity (%)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f5.2	Y
Range				

5.2.37. activity_err_acq_start

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Uncertainty of the activity at the start of acquisition (mBq)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq		g8.5	Y
Range				

5.2.38. activity_ratio

Table(s)	GARDS_RLR_RATIOS (RMSAUTO)			
Description	Activity ratio of nuclide 1 to nuclide 2			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	N
Range				

5.2.39. activity_ratio_err

Table(s)	GARDS_RLR_RATIOS (RMSAUTO)			
Description	Relative uncertainty of the activity ratio (%)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f5.2	Y
Range				

5.2.40. additional_info

Table(s)	GARDS_POC (RMSMAN)			
Description	Additional information for point of contact			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(500)			a500	Y
Range				

5.2.41. address

Table(s)	GARDS_POC (RMSMAN)			
Description	Additional information for a contact. This column includes information such as full name, title, company or university name, country, and so on.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(500)			a500	Y
Range				

5.2.42. alert_id

Table(s)	GARDS_ALERTS (RMSAUTO)			
Description	Unique alert identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.43. alert_text

Table(s)	GARDS_ALERTS (RMSAUTO)			
Description	Free text describing the alert			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
clob(4000)			a80	N
Range				

5.2.44. alert_type

Table(s)	GARDS_ALERTS (RMSAUTO)			
Description	Type of the alert (current alert types are ALERT_TEMP, ALERT_FLOW, ALERT_SYSTEM, and ALERT_UPS)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)		NOT ALLOWED	a20	N
Range	alert_type IN { ALERT_TEMP, ALERT_FLOW, ALERT_SYSTEM, ALERT_UPS }			

5.2.45. allow_release

Table(s)	GARDS_QAT_CONFIG (RMSMAN)			
Description	Flag indicating whether or not release is permitted			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.46. alpha

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Value of a variable used in bounds estimation			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.47. alpha

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.48. amplitude

Table(s)	GARDS_BG_QC_RESULT (RMSAUTO), GARDS_BG_QC_RESULT (RMSMAN)			
Description	Amplitude of QC peak			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.49. analysis_purpose

Table(s)	GARDS_RLR_OBJECTIVE (RMSAUTO)			
Description	Analysis purpose			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.50. analyst

Table(s)	RN_LOGS (RMSMAN)			
Description	Name of the analyst who reviewed the sample			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)			a30	Y
Range				

5.2.51. analyst

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Name of the analyst who generated the statistics			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.52. analyst

Table(s)	GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN), GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN), GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Name of the analyst who reviewed the sample			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.53. analyst

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)				Y
Range				

5.2.54. ape

Table(s)	GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN)			
Description	Average prediction error, a measure of the average confidence interval for the calculated energy vs. channel regression (ECR) over the range of 100 keV			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.55. app_abundance

Table(s)	GARDS_CSC_MODCOEFF_LIB (RMSMAN)			
Description	Apparent abundance			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.56. app_abundance_err

Table(s)	GARDS_CSC_MODCOEFF_LIB (RMSMAN)			
Description	Absolute error in apparent abundance			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.57. archive_bottle_id

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Unique identifier of the archive bottle in which the sample is stored			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(2)			a2	Y
Range				

5.2.58. area

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak area (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.59. area_adj

Table(s)	GARDS_QCPARAMS (RMSMAN)			
Description	Peak area standard deviation inflation factor			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.60. area_err

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Uncertainty of the peak area (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.61. area_lim

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	Minimum peak area for inclusion in matching (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.62. area_reject

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Indicator of whether or not to reject peaks with zero area indicator			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.63. assay_date

Table(s)	GARDS_SAMPLE_CERT (RMSAUTO)			
Description	Date sample was assayed			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y

Range	any valid ORACLE date

5.2.64. *assignable*

Table(s)	GARDS_USERS (RMSMAN)			
Description	Indicator of whether or not the user can be assigned stations: 1 = user can be assigned stations for review or 0 = user not to be assigned stations			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	N
Range				

5.2.65. *auth*

Table(s)	GARDS_DATA_LOG (RMSAUTO)			
Description	Authentication status of the associated message; 0 = not authenticated, 1 = authenticated.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range	auth IN { 0, 1 }			

5.2.66. *auth_status*

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Authentication status of the associated message; 0 = not authenticated, 1 = authenticated. Authentication status of the RN message (extracted from acq.acq_message.AUTH STATUS)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)			i1	Y
Range	auth IN { 0, 1 }			

5.2.67. *author*

Table(s)	GARDS_PRODUCT (RMSAUTO), GARDS_PRODUCT (RMSMAN), GARDS_QAT_NOTIFY (RMSMAN), GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN)			
Description	User name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.68. *auto_category*

Table(s)	GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)

Description	Sample categorization level after automated analysis			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.69. ave_activ

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Average activity concentration (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.70. ave_activ_acq_start

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Average activity at the start of acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq		g8.5	Y
Range				

5.2.71. ave_activ_err

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the average activity concentration (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.72. ave_activ_err_acq_start

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the average activity at the start of acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq		g8.5	Y
Range				

5.2.73. ave_humidity

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	Average humidity (percent relative humidity)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f8.3	Y
Range	0 < ave_humidity <= 100			

5.2.74. ave_out_temp

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	Average outside temperature (deg. C)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees C		f5.1	Y
Range				

5.2.75. ave_pressure

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	Average atmospheric pressure.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	millibars		i4	Y
Range	0.0 <= ave_pressure			

5.2.76. ave_wind_dir

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	Average wind direction (degrees from North)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees (clockwise from North)		f6.1	Y
Range	0.0 <= ave_wind_dir <= 360.0			

5.2.77. ave_wind_speed

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	Average wind speed (km/h)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	km/h		f5.1	Y
Range				

5.2.78. b_abundance

Table(s)	GARDS_SAMPLE_CERT_LINES (RMSAUTO)			
Description	Emission intensity (%) of beta particle or conversion electron per disintegration of one nucleus			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		g8.5	Y
Range				

5.2.79. b_chan_start

Table(s)	GARDS_ROI_CHANNELS (RMSAUTO), GARDS_ROI_CHANNELS (RMS-MAN)			
Description	Beta boundary start channel			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.80. b_chan_stop

Table(s)	GARDS_ROI_CHANNELS (RMSAUTO), GARDS_ROI_CHANNELS (RMS-MAN)			
Description	Beta boundary start channel.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.81. b_channels

Table(s)	GARDS_HISTOGRAM (RMSAUTO)			
Description	Number of beta channels in histogram			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.82. b_energy

Table(s)	GARDS_SAMPLE_CERT_LINES (RMSAUTO)			
Description	Energy of conversion electron or maximum energy of beta particle (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.83. b_energy_span

Table(s)	GARDS_HISTOGRAM (RMSAUTO)			
Description	Beta energy span of detector calibration (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.84. b_energy_start

Table(s)	GARDS_ROI_LIMITS (RMSAUTO), GARDS_ROI_LIMITS (RMSMAN)			
Description	Beta boundary start energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.85. b_energy_stop

Table(s)	GARDS_ROI_LIMITS (RMSAUTO), GARDS_ROI_LIMITS (RMSMAN)			
Description	Beta boundary stop energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.86. back_chan

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Number of continuous channels			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.87. back_channel

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Number of the average background counts per channel			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.88. back_count

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Average number of the continuum background counts.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.89. back_data_type

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Spectral type of the detector background: Blank (B), or detector background (D)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range	back_data_type IN {B, D}			

5.2.90. back_sample_id

Table(s)	GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Sample identifier for the sample used in background subtraction			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.91. back_type

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Background type: LINEAR or STEP			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)			a6	Y
Range				

5.2.92. back_uncer

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Uncertainty of the continuum background counts			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.93. baseline_type

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN)			
Description	Baseline type regarding gamma (G) or beta (B) spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	N
Range				

5.2.94. bayes_133_131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Bayesian minus ratio of Xe-133 to Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.95. bayes_133m_131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Bayesian minus ratio of Xe-133m to Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.96. bayes_133m_133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Bayesian minus ratio of Xe-133m to Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.97. bayes_135_133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)
Description	Bayesian minus ratio of Xe-135 to Xe-133

Storage type	Unit	NA value	External	Nullable
number			g8.5	Y
Range				

5.2.98. bayes_minus

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Percentile related to the lower bound of Bayesian ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.99. bayes_plus

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Percentile related to the upper bound of Bayesian ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.100. begin_date

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN), GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN), GARDS_EFFICIENCY_VGSL_PAIRS (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_STATIONS_SCHEDULE (RMSMAN)			
Description	Initialization date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.101. begin_date

Table(s)	GARDS_IRF (RMSMAN), GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), RN_LOGS (RMSMAN)			
Description	Initialization date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.102. beta_coeff1

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	Zero order coefficient of the beta energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.103. beta_coeff1

Table(s)	GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN), GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN)			
Description	Zero order coefficient of the beta energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.104. beta_coeff2

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	First order coefficient of the beta energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.105. beta_coeff2

Table(s)	GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN), GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN)			
Description	First order coefficient of the beta energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.106. beta_coeff3

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)
Description	Second order coefficient of the beta energy calibration

Storage type	Unit	NA value	External	Nullable
float(126)			g8.5	Y
Range				

5.2.107. beta_coeff3

Table(s)	GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN), GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN)			
Description	Second order coefficient of the beta energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.108. beta_ecr_order

Table(s)	GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN), GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Order of the beta energy calibration function			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.109. bg_effic_error

Table(s)	GARDS_BG_EFFICIENCY_PAIRS (RMSAUTO), GARDS_BG_EFFICIENCY_PAIRS (RMSMAN)			
Description	Uncertainty of bg_efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.110. bg_efficiency

Table(s)	GARDS_BG_EFFICIENCY_PAIRS (RMSAUTO), GARDS_BG_EFFICIENCY_PAIRS (RMSMAN)			
Description	Detection efficiency of b-g coincidence event			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.111. bin_beta_start

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	The beta start offset in histogram			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.112. bin_gamma_start

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	The gamma start offset in histogram			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.113. bin_max_vector_size

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Maximum size for the bin vector			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.114. bin_min_count

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Minimum counts in each bin			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.115. bin_rows

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)

Description	Number of rows to merge in binning			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.116. bkgd_measurement_id

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Detector background measurement identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(31)			a31	N
Range				

5.2.117. bkgnd_area

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN)			
Description	Background area			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.118. bkgnd_area_err

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN)			
Description	Uncertainty of the background area			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.119. bootstrap

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	0/1 flag for ignoring prior ECR/RER (typically 0)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.120. branch_ratio

Table(s)	GARDS_AUX_LIB (RMSMAN)
Description	Percentage of decays to a given decay path

Storage type	Unit	NA value	External	Nullable
number	percent		g8.5	Y
Range				

5.2.121. buildtype

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Flag indicating activity or concentration calculations: activity (NONE), concentration (BUILD)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

5.2.122. cal_energy

Table(s)	GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN)			
Description	Beta energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.123. cal_energy

Table(s)	GARDS_ENERGY_PAIRS (RMSAUTO), GARDS_ENERGY_PAIRS (RMSMAN), GARDS_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_ENERGY_PAIRS_ORIG (RMSMAN)			
Description	Gamma energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.124. cal_error

Table(s)	GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_ENERGY_PAIRS (RMSAUTO), GARDS_ENERGY_PAIRS (RMSMAN), GARDS_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_ENERGY_PAIRS_ORIG (RMSMAN)			
Description	Uncertainty of the centroid channel			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.125. cal_mode

Table(s)	GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN)			
Description	Calibration mode used to determine the beta and gamma energy calibrations for the detectors			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	N
Range				

5.2.126. calibration_dtg

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Date of most recent calibration at detector site			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.127. calibration_type

Table(s)	GARDS_REFLINE_MASTER (RMSMAN), GARDS_XE_REFLINE_MASTER (RMSMAN)			
Description	Calibration type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(3)			a3	Y
Range				

5.2.128. cat_sample

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Categorization level of the sample depending on the maximum level of four xenon isotopes			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.129. cat_xe131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)
Description	Categorization level for Xe-131m (=1, 2 and 3 for level A, B and C respectively)

Storage type	Unit	NA value	External	Nullable
number			i1	Y
Range				

5.2.130. cat_xe133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Categorization level for Xe-133 (=1, 2 and 3 for level A, B and C respectively)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.131. cat_xe133m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Categorization level for Xe-133m (=1, 2 and 3 for level A, B and C respectively)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.132. cat_xe135

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Categorization level for Xe-135 (=1, 2 and 3 for level A, B and C respectively)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.133. category

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Categorization level			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.134. category

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN)

Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.135. category_list

Table(s)	GARDS_QAT_QUERY_FILTER (RMSMAN)			
Description	Comma-separated list of categories			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a1024	Y
Range				

5.2.136. central_value

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN)			
Description	Current estimate of the level of the random process			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.137. central_value

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN), GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.138. centroid

Table(s)	GARDS_BG_QC_RESULT (RMSAUTO), GARDS_BG_QC_RESULT (RMSMAN)			
Description	Centroid energy of QC peak			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	keV		g8.5	Y
Range				

5.2.139. centroid

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN),

	GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak centroid (channels)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.140. centroid_err

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Uncertainty of the peak centroid.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.141. chain_id

Table(s)	GARDS_AUX_LIB (RMSMAN)			
Description	Decay chain index			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.142. chain_pos

Table(s)	GARDS_AUX_LIB (RMSMAN)			
Description	Position in the decay chain			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.143. channel

Table(s)	GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_ENERGY_PAIRS (RMSAUTO), GARDS_ENERGY_PAIRS (RMSMAN), GARDS_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_ENERGY_PAIRS_ORIG (RMSMAN)			
Description	Peak centroid channel			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.144. channels

Table(s)	GARDS_DETECTORS (RMSMAN), GARDS_SPECTRUM (RMSAUTO)			
Description	Number of channels in spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.145. class

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Workflow field indicating the station			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	N
Range				

5.2.146. class

Table(s)	GARDS_INTERVAL (RMSAUTO)			
Description	Station name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	Y
Range				

5.2.147. cnf_begin_date

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Date when last analysis began			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.148. cnf_end_date

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Date when last analysis ended.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.149. cnv_factor

Table(s)	GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN)			
Description	Conversion to keV			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.150. code

Table(s)	GARDS_STATIONS_SCHEDULE (RMSMAN)			
Description	Code symbol SCHEDULE as found in the gards_codes table. SF_GP for full sample grace period or SPHDF for full sample PHD			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(5)			a5	N
Range	code IN {SF_GP, SPHDF}			

5.2.151. code

Table(s)	GARDS_CODES (RMSMAN)			
Description	Code symbol.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(5)			a5	N
Range				

5.2.152. coeff

Table(s)	GARDS_ENERGY_CAL_COV (RMSAUTO), GARDS_ENERGY_CAL_COV (RMSMAN), GARDS_RESOLUTION_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSMAN)			
Description	Calibration coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.153. coeff1

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the zero order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.154. coeff1

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			
Description	zero order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.155. coeff1

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	zero order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.156. coeff2

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			
Description	First order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.157. coeff2

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the first order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.158. coeff2

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	First order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.159. coeff3

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			
Description	Second order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.160. coeff3

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the second order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.161. coeff3

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	Second order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.162. coeff4

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			
Description	Third order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.163. coeff4

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the third order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.164. coeff4

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	Third order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.165. coeff5

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			
Description	Fourth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.166. coeff5

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the fourth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.167. coeff5

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	Fourth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.168. coeff6

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			
Description	Fifth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.169. coeff6

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the fifth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.170. coeff6

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)

Description	Fifth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.171. coeff7

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			
Description	Sixth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.172. coeff7

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the sixth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.173. coeff7

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	Sixth order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.174. coeff8

Table(s)	GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN)			

Description	Seventh order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.175. coeff8

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty of the seventh order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.176. coeff8

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	Seventh order coefficient			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			g8.5	Y
Range				

5.2.177. col_index

Table(s)	GARDS_ENERGY_CAL_COV (RMSMAN)			
Description	Column index			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	N
Range				

5.2.178. col_index

Table(s)	GARDS_ENERGY_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSMAN)			
Description	Column index			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	N
Range				

5.2.179. collect_start

Table(s)	GARDS_RLR (RMSAUTO), GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Collection start date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.180. collect_stop

Table(s)	GARDS_RLR (RMSAUTO), GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Collection stop date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.181. comment_category

Table(s)	GARDS_COMMENTS_DEFS (RMSMAN)			
Description	Comment category			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)			a10	Y
Range				

5.2.182. comment_date

Table(s)	RN_LOGS (RMSMAN)			
Description	Date of issuing the analyst comment			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date			a21	Y
Range				

5.2.183. comment_id

Table(s)	GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN), GARDS_USER_COMMENTS (RMSAUTO), GARDS_USER_COMMENTS (RMSMAN), RN_LOGS (RMSMAN), RN_LOGS_TXT (RMSMAN)			
Description	Unique comment identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N

Range	

5.2.184. comment_text

Table(s)	GARDS_QAT_NOTIFY (RMSMAN), GARDS_USER_COMMENTS (RMSAUTO), GARDS_USER_COMMENTS (RMSMAN)			
Description	Comment text related to the comment type and/or comment id			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.185. comment_text

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Comment on the filter			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a256	Y
Range				

5.2.186. comment_text

Table(s)	GARDS_COMMENTS_DEFS (RMSMAN)			
Description	Comment text related to the comment type and/or comment id			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a256	Y
Range				

5.2.187. comment_text

Table(s)	RN_LOGS_TXT (RMSMAN)			
Description	Comment text related to the comment type and/or comment id			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(4000)			a2048	Y
Range				

5.2.188. comment_txt

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Free-form user comments			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(300)			a300	Y
Range				

5.2.189. comment_type

Table(s)	GARDS_COMMENTS_DEFS (RMSMAN)			
Description	Comment code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.190. comment_type

Table(s)	GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN), GARDS_QAT_NOTIFY (RMSMAN), RN_LOGS (RMSMAN)			
Description	Comment code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.191. comments

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)				Y
Range				

5.2.192. comp_confid

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Computed confidence index			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.193. compton

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN), GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Counts of Compton continuum background			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y

Range	

5.2.194. compton_err

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of Compton counts			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.195. conc

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq/m3		g8.5	Y
Range				

5.2.196. conc

Table(s)	GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.197. conc_133_131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Ratio of activity concentrations for Xe-133 to Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.198. conc_133_131m_mdc

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)
Description	Ratio of activity concentrations for Xe-133 to Xe-131m, where the Xe-131m

	MDC is used when Xe-131m is not detected.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.199. conc_133m_131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Ratio of activity concentrations for Xe-133m to Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.200. conc_133m_133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Ratio of activity concentrations for Xe-133m to Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.201. conc_135_133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Ratio of activity concentrations for Xe-135 to Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.202. conc_confidence_level

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Level of confidence			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f5.2	Y
Range				

5.2.203. conc_coverage_factor

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)

Description	Coverage factor			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f5.2	Y
Range				

5.2.204. conc_err

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Uncertainty of the activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq/m3		g8.5	Y
Range				

5.2.205. conc_error

Table(s)	GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Uncertainty of the activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.206. concentration

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Concentration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Bq/m3		e11.4	Y
Range				

5.2.207. concentration_date

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Concentration reference date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.208. concentration_err

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Relative uncertainty of the activity concentration (%)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f5.2	Y
Range				

5.2.209. count_ratio

Table(s)	GARDS_SAMPLE_RATIOS (RMSAUTO), GARDS_SAMPLE_RATIOS (RMSMAN)			
Description	Ratio of the upper ROI counts to lower ROI counts			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.210. count_ratio_err

Table(s)	GARDS_SAMPLE_RATIOS (RMSAUTO), GARDS_SAMPLE_RATIOS (RMSMAN)			
Description	Uncertainty of the ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.211. country_code

Table(s)	GARDS_STATIONS (RMSMAN)			
Description	Country code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)			a2	Y
Range				

5.2.212. counts_sec

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Counts per second			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	cps		g8.5	Y
Range				

5.2.213. counts_sec_err

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Uncertainty of the counts per second			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	cps		g8.5	Y
Range				

5.2.214. cov_radon

Table(s)	GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Covariance matrix line 5 for radon			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.215. cov_xe_131m

Table(s)	GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Covariance matrix line 1 for Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.216. cov_xe_133

Table(s)	GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Covariance matrix line 3 for Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.217. cov_xe_133m

Table(s)	GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Covariance matrix line 2 for Xe-133m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.218. cov_xe_135

Table(s)	GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Covariance matrix line 4 for Xe-135			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.219. creation_time

Table(s)	NMS_RECEIVED_SOFAR (RMSAUTO)			
Description	Latest access time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date		a21	Y
Range				

5.2.220. crit_level

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Critical level indicator			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.221. critical_lev_gas

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Critical level in counts in the gas background spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.222. critical_lev_samp

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Critical level in counts in the sample spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.223. csc_mod_flag

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Flag indicating if the cascade summing correction is performed: 1 = nuclide was modified, 0 = nuclide was not modified			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.224. csc_ratio

Table(s)	GARDS_CSC_MODCOEFF_LIB (RMSMAN), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Multiplier of the cascade summing correction to update active_key and mda			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.225. csc_ratio_err

Table(s)	GARDS_CSC_MODCOEFF_LIB (RMSMAN), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Uncertainty in csc_ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.226. data_offset

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Offset of the data type within the RN message			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y

Range	

5.2.227. data_size

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Size of the data type in the RN message			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.228. data_type

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN), GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_REFLINE_MASTER (RMSMAN), GARDS_UPDATE_REFLINES (RMSMAN), GARDS_XE_REFLINE_MASTER (RMSMAN)			
Description	Data type of the spectrum: uncontaminated blank filter (B), calibration background (C), detector background (D), quality control (Q), or sample (S).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	N
Range	data_type IN {B, C, D, Q, S}			

5.2.229. data_type

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Data type of the spectrum: uncontaminated blank filter (B), calibration background (C), detector background (D), quality control (Q), or sample (S).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range	data_type IN {B, C, D, Q, S}			

5.2.230. data_type

Table(s)	GARDS_DATA_LOG (RMSAUTO)			
Description	Type of the data message (current choices include: ALERT, MET, SOH, SPHD-B, SPHD-C, SPHD-D, SPHD-F, SPHD-P, SPHD-Q, SPHD-G)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)			a10	Y
Range	data_type IN { ALERT, MET, SOH, SPHD-B, SPHD-C, SPHD-D, SPHD-F, SPHD-P, SPHD-Q, SPHD-G }			

5.2.231. data_type

Table(s)	GARDS_PROCESSING_ERRORS (RMSMAN)

Description	Type of the data message (current choices include: MET, RMSSOH, SAMPLEPHD, QCPHD, GASBKGPHD, SPIKEPHD, DETBKGPHD, BLANKPHD, ALERT_FLOW, ALERT_TEMP, ALERT_UPS, ALERT_SYSTEM, …)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(12)			a12	Y
Range				

5.2.232. data_type

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Type of the data message (current choices include: MET, RMSSOH, SAMPLEPHD, QCPHD, GASBKGPHD, SPIKEPHD, DETBKGPHD, BLANKPHD, ALERT_FLOW, ALERT_TEMP, ALERT_UPS, ALERT_SYSTEM, …)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a12	Y
Range				

5.2.233. date_begin

Table(s)	GARDS_DETECTORS (RMSMAN), GARDS_STATIONS (RMSMAN)			
Description	Initialization date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.234. date_end

Table(s)	GARDS_DETECTORS (RMSMAN), GARDS_STATIONS (RMSMAN)			
Description	Decommissioning date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.235. date_received

Table(s)	GARDS_DATA_LOG (RMSAUTO)			
Description	Date that sample was received			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.236. date_value

Table(s)	GARDS_TRENDVUE (RMSAUTO), GARDS_TRENDVUE (RMSMAN)			
Description	Date of the Trendvue plot			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.237. db_name

Table(s)	GARDS_PERMISSIONS (RMSMAN), GARDS_ROLES (RMSMAN)			
Description	Database name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(35)			a35	N
Range				

5.2.238. decay_mode

Table(s)	GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_SAMPLE_CERT_LINES (RMSAUTO)			
Description	Decay type: B = beta particle, C = conversion electron			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range	decay_mode IN {B, C}			

5.2.239. def_det_bkgnd_sample_id

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Default detector background sample			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.240. def_gas_decay_time

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Default gas decay time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.241. default_method

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Default method identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.242. default_role

Table(s)	GARDS_USERS_ROLES (RMSMAN)			
Description	Integer representation of the default role name to be assigned to the user			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.243. degree

Table(s)	GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Polynomial degree of the efficiency equation			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.244. del

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN), GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.245. delta

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN)			
Description	Value of a variable used in bounds estimation (EWMA algorithm)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.246. desciption

Table(s)	GARDS_NOTIFY (RMSAUTO), GARDS_NOTIFY (RMSMAN)			
Description	General description of occurrence			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(80)			a80	Y
Range				

5.2.247. description

Table(s)	GARDS_LAB_CATEGORY_DESCRIPTION (RMSMAN)			
Description	Description of the lab_sample_category			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a256	Y
Range				

5.2.248. description

Table(s)	GARDS_CODES (RMSMAN)			
Description	Description of the detector.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)			a40	Y
Range				

5.2.249. description

Table(s)	GARDS_STATIONS (RMSMAN)			
Description	Description of the station location			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)			a40	Y
Range				

5.2.250. description

Table(s)	GARDS_SAMPLE_DESCRIPTION (RMSAUTO)			
Description	Comments			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(512)			a512	Y
Range				

5.2.251. description

Table(s)	GARDS_DETECTORS (RMSMAN)

Description	Description of the code, detector, or station.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(80)			a80	Y
Range				

5.2.252. description

Table(s)	GARDS_NOTIFY (RMSAUTO)			
Description	General description of occurrence			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(80)			a80	Y
Range				

5.2.253. det

Table(s)	GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN)			
Description	Determinant of the weighted design matrix, a measure of the spread of fitted data points and the error in the fitted points			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.254. det_back_used

Table(s)	GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Flag indicating whether the detector background is in use or not (1 = YES, 0 = NO)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.255. det_bkgnd_count

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Gross number of counts in the detector background spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.256. det_bkgnd_count_err

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of the gross number of counts in the detector background spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.257. det_bkgnd_id

Table(s)	GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Unique identifier for the gas background used in the calculation			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.258. det_bkgnd_used

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Flag indicating whether the detector background measurement is used in the sample spectrum analysis. 0 means not used, 1 indicates used.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range	det_bkgnd_used IN { 0,1}			

5.2.259. detectability

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak detectability			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f9.6	Y
Range				

5.2.260. detector_back

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Detector background counts			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y

Range	

5.2.261. detector_back_err

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of detector_back (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.262. detector_code

Table(s)	GARDS_DETECTORS (RMSMAN), GARDS_PROCESSING_ERRORS (RMSMAN), GARDS_SAMPLE (RMSMAN)			
Description	Detector code parsed from import message			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(9)			a9	Y
Range				

5.2.263. detector_id

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN), GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_DETECTOR_STD_SPECTRA (RMSAUTO), GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_CSC_MODCOEFF_LIB (RMSMAN), GARDS_DETECTORS (RMSMAN), GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN), GARDS_EFFICIENCY_VGSL_PAIRS (RMSMAN), GARDS_IRF (RMSMAN), GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_QCPARAMS (RMSMAN), GARDS_QCTARGETS (RMSMAN), GARDS_STADET (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN), GARDS_UPDATE_REFLINES (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Unique detector identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.264. detector_id

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_QCHISTORY

	(RMSAUTO), GARDS_QCHISTORY (RMSMAN), GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN), GARDS_SOH_CHAR_DATA (RMSAUTO), GARDS_SOH_HEADER (RMSAUTO), GARDS_SOH_NUM_DATA (RMSAUTO), GARDS_SOH_SENSOR_DATA (RMSAUTO)			
Description	Unique detector identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.265. dev_tolerance

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Tolerance			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.266. dfile

Table(s)	GARDS_PRODUCT (RMSAUTO), GARDS_PRODUCT (RMSMAN)			
Description	Name of file containing the data			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	Y
Range				

5.2.267. dir

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Full path to the data file. Directory where the RN message is stored			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(100)			a100	Y
Range				

5.2.268. dir

Table(s)	GARDS_PRODUCT (RMSAUTO), GARDS_PRODUCT (RMSMAN)			
Description	Full path to the data file			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a100	Y
Range				

5.2.269. display_detector

Table(s)	GARDS_SOH_CODE (RMSMAN)			
Description	Flag indicating whether the parameter is associated with a detector			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.270. display_station

Table(s)	GARDS_SOH_CODE (RMSMAN)			
Description	Flag indicating whether the parameter is associated with a station			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.271. dlid

Table(s)	GARDS_DATA_LOG (RMSAUTO)			
Description	Unique data log identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.272. do_back

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Flag that indicates whether or not background subtraction should be performed: 1 = yes, 0 = no.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range	do_back IN {0,1}			

5.2.273. do_csc

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Flag that indicates whether or not cascade summing correction should be performed: 1 = yes, 0 = no.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y

Range	do_csc IN {0,1}

5.2.274. do_pd_calc

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Flag indicates if the parent/daughter calculations should be run; 1 = run calculations, 0 = do not run calculations			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.275. do_reru

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	Flag that indicates if a resolution update was performed			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.276. dsize

Table(s)	GARDS_PRODUCT (RMSAUTO), GARDS_PRODUCT (RMSMAN)			
Description	Size of data file in bytes			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.277. dtg

Table(s)	GARDS_ALERTS (RMSAUTO)			
Description	Transmit date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.278. dtg

Table(s)	GARDS_QAT_NOTIFY (RMSMAN)			
Description	Date time group for QAT comment notification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.279. dtg

Table(s)	GARDS_QCTARGETS (RMSMAN)			
Description	Decay correction reference date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.280. dtg

Table(s)	GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN), GARDS_FPE (RMSAUTO), GARDS_FPE (RMSMAN), GARDS_STATUS_HISTORY (RMSMAN)			
Description	Entry date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.281. dtg

Table(s)	GARDS_DATA_LOG (RMSAUTO), GARDS_RECEIPT_LOG (RMSAUTO)			
Description	Transmit date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.282. dtg

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date				Y
Range				

5.2.283. dtg_begin

Table(s)	GARDS_NOTIFY (RMSAUTO), GARDS_NOTIFY (RMSMAN), GARDS_NUCL2QUANTIFY (RMSMAN), GARDS_SOH_CHAR_DATA (RMSAUTO), GARDS_SOH_HEADER (RMSAUTO), GARDS_SOH_NUM_DATA (RMSAUTO), GARDS_SOH_SENSOR_DATA (RMSAUTO)

Description	Begin date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.284. dtg_begin

Table(s)	GARDS_MDAS2REPORT (RMSMAN)			
Description	Begin date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.285. dtg_end

Table(s)	GARDS_SOH_CHAR_DATA (RMSAUTO), GARDS_SOH_HEADER (RMSAUTO), GARDS_SOH_NUM_DATA (RMSAUTO), GARDS_SOH_SENSOR_DATA (RMSAUTO)			
Description	End date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.286. dtg_end

Table(s)	GARDS_MDAS2REPORT (RMSMAN), GARDS_NOTIFY (RMSAUTO), GARDS_NOTIFY (RMSMAN), GARDS_NUCL2QUANTIFY (RMSMAN)			
Description	End date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.287. ecal_range_max

Table(s)	GARDS_DETECTORS (RMSMAN)			
Description	Maximum detector calibration energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.288. ecr_slope

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO),

	GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN)			
Description	Default ECR slope			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.289. eff_error

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Uncertainty of the efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.290. effic

Table(s)	GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.291. effic

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN)			
Description	Efficiency at the energy			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.292. effic_energy

Table(s)	GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN), GARDS_EFFICIENCY_PAIRS (RMSAUTO), GARDS_EFFICIENCY_PAIRS (RMSMAN), GARDS_EFFICIENCY_VGSL_PAIRS (RMSMAN), GARDS_TOTAL_EFFIC (RMSAUTO), GARDS_TOTAL_EFFIC (RMSMAN)			
Description	Gamma energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number	keV		g8.5	N
Range				

5.2.293. effic_err

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.294. effic_error

Table(s)	GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN)			
Description	Uncertainty of the efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	N
Range				

5.2.295. effic_error

Table(s)	GARDS_EFFICIENCY_PAIRS (RMSAUTO), GARDS_EFFICIENCY_PAIRS (RMSMAN), GARDS_EFFICIENCY_VGSL_PAIRS (RMSMAN), GARDS_TOTAL_EFFIC (RMSAUTO), GARDS_TOTAL_EFFIC (RMSMAN)			
Description	Uncertainty of the efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.296. efficiency

Table(s)	GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN)			
Description	Efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	N
Range				

5.2.297. efficiency

Table(s)	GARDS_EFFICIENCY_PAIRS (RMSAUTO), GARDS_EFFICIENCY_PAIRS (RMSMAN), GARDS_EFFICIENCY_VGSL_PAIRS (RMSMAN), GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN), GARDS_TOTAL_EFFIC (RMSAUTO), GARDS_TOTAL_EFFIC (RMSMAN)			
Description	Efficiency			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.298. efftype

Table(s)	GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN)			
Description	Efficiency type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(8)			a8	Y
Range				

5.2.299. elevation

Table(s)	GARDS_STATIONS (RMSMAN)			
Description	Elevation (m)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	metres		f8.5	Y
Range				

5.2.300. email_addr

Table(s)	GARDS_NOTIFY (RMSAUTO), GARDS_NOTIFY (RMSMAN)			
Description	Recipient email addresses			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(80)			a80	N
Range				

5.2.301. email_address

Table(s)	GARDS_POC (RMSMAN)			
Description	Email address of point of contact			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)			a50	Y

Range	

5.2.302. end_date

Table(s)	GARDS_EFFICIENCY_VGSL_PAIRS (RMSMAN)			
Description	Decommissioning date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.303. end_date

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN), GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN), GARDS_IRF (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_STATIONS_SCHEDULE (RMSMAN), GARDS_XE_CATEGORY_PARAMS (RMSMAN), RN_LOGS (RMSMAN)			
Description	Decommissioning date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.304. end_env_time

Table(s)	GARDS_ENVIRONMENT (RMSAUTO)			
Description	End date of environment data			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date		NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.305. end_env_time_sec

Table(s)	GARDS_ENVIRONMENT (RMSAUTO)			
Description	End date of environmental data.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	seconds		g8.5	Y
Range				

5.2.306. end_time

Table(s)	GARDS_MET_DATA (RMSAUTO), GARDS_MET_HEADER (RMSAUTO)

Description	End time of the period			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.307. end_time

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	End time of the period			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.308. end_time

Table(s)	GARDS_INTERVAL (RMSAUTO)			
Description	End time of the period			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.309. energy

Table(s)	GARDS_AUX_LINES_LIB (RMSMAN), GARDS_CSC_MODCOEFF_LIB (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_SAMPLE_CERT_LINES (RMSAUTO), GARDS_XE_NUCL_LINES_LIB (RMSMAN)			
Description	Energy of the decay line (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.310. energy

Table(s)	GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Energy of the peak (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.311. energy

Table(s)	GARDS_QCTARGETS (RMSMAN)			
Description	Gamma energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.312. energy

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Energy of the peak (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.313. energy

Table(s)	GARDS_IRF (RMSMAN), GARDS_QAT_NOTIFY (RMSMAN)			
Description	Gamma energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.314. energy_err

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN), GARDS_XE_NUCL_LINES_LIB (RMSMAN)			
Description	Uncertainty of the energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.315. energy_high

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN)

Description	Higher energy of the bin			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.316. energy_low

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN)			
Description	Lower energy of the bin			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.317. energy_span

Table(s)	GARDS_SPECTRUM (RMSAUTO)			
Description	Total energy span of detector (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		f8.5	Y
Range				

5.2.318. energy_tol

Table(s)	GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Tolerance for nuclide identification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.319. energy_units

Table(s)	GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN)			
Description	Energy units			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(3)			a3	Y
Range	energy_units IN {MeV, keV}			

5.2.320. entry_date

Table(s)	GARDS_PROCESSING_ERRORS (RMSMAN), GARDS_RLR (RMSAUTO),

	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Date entered into the IDC database			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.321. error

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	Uncertainty of the estimate			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.322. error

Table(s)	GARDS_SAMPLE_CERT_LINES (RMSAUTO)			
Description	Relative uncertainty of the activity (%)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f8.3	Y
Range				

5.2.323. error_text

Table(s)	GARDS_PROCESSING_ERRORS (RMSMAN)			
Description	Text associated with the error			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a20	Y
Range				

5.2.324. estimate

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	The estimate of (SpectraContribution * NumberOfDecays)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.325. *etol*

Table(s)	GARDS_QCPARAMS (RMSMAN)			
Description	Energy tolerance used to match peaks with energies (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.326. *event*

Table(s)	GARDS_NOTIFY (RMSAUTO), GARDS_NOTIFY (RMSMAN)			
Description	Type of occurrence (choices are: ALERT, FISS_FOUND, NIC_SIG34, NIC_SIG5, QC_ERROR, QC_WARNING, RADOPSCAL, RMS_ADMIN)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)			a10	N
Range				

5.2.327. *ext_ref_id*

Table(s)	RN_LOGS (RMSMAN)			
Description	External reference to related analyst comment, not used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.328. *f_linear*

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	0/1 flag for forcing ECR to be first-order (typically 0)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.329. *file_size*

Table(s)	GARDS_DATA_LOG (RMSAUTO)			
Description	Size of data file in bytes			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.330. filename

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Input data filename			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(100)			a100	Y
Range				

5.2.331. filename

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Input data filename			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a100	Y
Range				

5.2.332. filename

Table(s)	GARDS_HISTOGRAM (RMSAUTO)			
Description	Name of the file containing the histogram data			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a256	Y
Range				

5.2.333. filename

Table(s)	GARDS_SPECTRUM (RMSAUTO)			
Description	Name of the file containing the spectrum data			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a256	Y
Range				

5.2.334. filename

Table(s)	GARDS_PROCESSING_ERRORS (RMSMAN)			
Description	Input data filename			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(512)			a100	Y
Range				

5.2.335. first_name

Table(s)	GARDS_POC (RMSMAN)

Description	First name of point of contact			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)			a20	Y
Range				

5.2.336. fit_singlets

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Fit singlets flag			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.337. fitted

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak fit indicator			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.338. fix_fwhm

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Fixed FWHM during peak search flag			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.339. flag

Table(s)	GARDS_QC_RESULTS (RMSAUTO), GARDS_QC_RESULTS (RMSMAN)			
Description	Quality control flag			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.340. flag_133_131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133 to Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.341. flag_133m_131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133m to Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.342. flag_133m_133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Flag whether or not the Bayesian minus ratio is above the limit for Xe-133m to Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.343. flag_135_133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Flag whether or not the Bayesian minus ratio is above the limit for Xe-135 to Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.344. flag_atm

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)
Description	Flag whether or not a known source would be detected in the sample, which is

	determined using ATM simulations.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.345. flag_id

Table(s)	GARDS_SAMPLE_FLAGS (RMSAUTO), GARDS_SAMPLE_FLAGS (RMSMAN)			
Description	Flag identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.346. flag_id

Table(s)	GARDS_FLAGS (RMSMAN)			
Description	Flag identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i10	N
Range				

5.2.347. flag_id

Table(s)	GARDS_FLAGS (RMSAUTO)			
Description	Flag identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i10	Y
Range				

5.2.348. flag_shortterm

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Flag on the statistics range for abnormal threshold determination (1 for short term, 0 for long term)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	N
Range				

5.2.349. flow_rate

Table(s)	GARDS_ENVIRONMENT (RMSAUTO)

Description	Flow rate through filter (m3/h)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	m3/h		g8.5	Y
Range				

5.2.350. foff

Table(s)	GARDS_PRODUCT (RMSAUTO), GARDS_PRODUCT (RMSMAN)			
Description	Offset into the data file (bytes)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Bytes		i10	Y
Range				

5.2.351. fpid

Table(s)	GARDS_FPE (RMSMAN)			
Description	Unique fission product identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.352. fpid

Table(s)	GARDS_FPE (RMSAUTO)			
Description	Unique fission product identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.353. fwhm

Table(s)	GARDS_BG_QC_RESULT (RMSAUTO), GARDS_BG_QC_RESULT (RMSMAN)			
Description	Full width at half maximum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	keV		g8.5	Y
Range				

5.2.354. fwhm

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)
Description	Full width at half maximum

Storage type	Unit	NA value	External	Nullable
number	keV		g8.5	Y
Range				

5.2.355. *fwhm_err*

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Uncertainty of full width at half maximum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.356. *fwhm_mult_width*

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Number of FWHMs to search to determine multiplets			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.357. *g_chan_start*

Table(s)	GARDS_ROI_CHANNELS (RMSAUTO), GARDS_ROI_CHANNELS (RMSMAN)			
Description	Gamma boundary start channel			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.358. *g_chan_stop*

Table(s)	GARDS_ROI_CHANNELS (RMSAUTO), GARDS_ROI_CHANNELS (RMSMAN)			
Description	Gamma boundary start channel.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.359. *g_channels*

Table(s)	GARDS_HISTOGRAM (RMSAUTO)

Description	Number of gamma channels in histogram			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.360. g_energy_span

Table(s)	GARDS_HISTOGRAM (RMSAUTO)			
Description	Number of gamma channels in histogram.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.361. g_energy_start

Table(s)	GARDS_ROI_LIMITS (RMSAUTO), GARDS_ROI_LIMITS (RMSMAN)			
Description	Gamma boundary start energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.362. g_energy_stop

Table(s)	GARDS_ROI_LIMITS (RMSAUTO), GARDS_ROI_LIMITS (RMSMAN)			
Description	Gamma boundary stop energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.363. gainchange

Table(s)	GARDS_QCPARAMS (RMSMAN)			
Description	Gain first difference tolerance			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.364. gainshift

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)
Description	Gain change for matching, percent (typically 0.1)

Storage type	Unit	NA value	External	Nullable
number			f8.3	Y
Range				

5.2.365. gamma

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Value of a variable used in bounds estimation			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.366. gamma

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.367. gamma_coeff1

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	Zero order coefficient of the gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.368. gamma_coeff1

Table(s)	GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN)			
Description	First order coefficient of the gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.369. gamma_coeff1

Table(s)	GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN)
Description	Zero order coefficient of the gamma energy calibration

Storage type	Unit	NA value	External	Nullable
number			g8.5	Y
Range				

5.2.370. gamma_coeff2

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	First order coefficient of the gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.371. gamma_coeff2

Table(s)	GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN)			
Description	First order coefficient of the gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.372. gamma_coeff2

Table(s)	GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN)			
Description	Second order coefficient of the gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.373. gamma_coeff3

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	Second order coefficient of the gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.374. gamma_coeff3

Table(s)	GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN),

	GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN)			
Description	Second order coefficient of the gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.375. gamma_ecr_order

Table(s)	GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN), GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Order of the gamma energy calibration function			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.376. gas_back_used

Table(s)	GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Flag indicating whether the gas background is in use or not (1 = YES, 0 = NO)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.377. gas_bkgd_measurement_id

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Gas background measurement identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(31)			a31	N
Range				

5.2.378. gas_bkgnd_count

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Net number of counts in the gas background spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y

Range	

5.2.379. gas_bkgnd_count_err

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of the net number of counts in the gas background spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.380. gas_bkgnd_gross

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Gross number of counts in the gas background spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.381. gas_bkgnd_id

Table(s)	GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Unique gas background identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.382. gas_bkgnd_used

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Flag indicating whether the gas background measurement is used in the sample spectrum analysis. 0 means not used, 1 indicates used, 2 means used if the counts are above LC.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range	gas_bkgnd_used IN { 0,1,2}			

5.2.383. geometry

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)

Description	Sample geometry			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(17)			a17	Y
Range				

5.2.384. gross

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Gross number of counts in the ROI			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.385. gross

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Gross number of counts in the sample spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.386. gross_err

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of the gross number of counts in the ROI			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.387. gross_err

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of the gross number of counts in the sample spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.388. halflife

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG

	(RMSMAN), GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_ROI_LIB (RMSMAN), GARDS_SAMPLE_CERT_LINES (RMSAUTO), GARDS_XE_NUCL_LIB (RMSMAN)			
Description	Half-life in seconds (S), hours (H), minutes (M), or years (Y)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(23)			a23	Y
Range				

5.2.389. halflife_act

Table(s)	GARDS_AUX_LIB (RMSMAN)			
Description	Half-life in seconds (S), hours (H), minutes (M), or years (Y)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(23)			a23	Y
Range				

5.2.390. halflife_act_err

Table(s)	GARDS_AUX_LIB (RMSMAN)			
Description	Uncertainty of half-life			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(23)			a23	Y
Range				

5.2.391. halflife_err

Table(s)	GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_ROI_LIB (RMSMAN), GARDS_XE_NUCL_LIB (RMSMAN)			
Description	Uncertainty of half-life			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(23)			a23	Y
Range				

5.2.392. halflife_sec

Table(s)	GARDS_BG_PROC_PARAMS_ROI (RMSAUTO), GARDS_BG_PROC_PARAMS_ROI (RMSMAN), GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_ROI_LIB (RMSMAN), GARDS_XE_NUCL_LIB (RMSMAN)			
Description	Half-life in seconds			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	seconds		g8.5	Y

Range	

5.2.393. hold

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_SAMPLE_CAT (RMSMAN)			
Description	Flag for holding current sample from being used in the history for categorizing subsequent samples (0 = use; 1 = skip)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.394. hold

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN), GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.395. id_percent

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Nuclide percentage			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f8.5	Y
Range				

5.2.396. idc_summary

Table(s)	GARDS_RLR_CONCLUSIONS (RMSAUTO)			
Description	IDC findings			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.397. ided

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)
Description	Peak identification indicator; 1 = peak was associated with a nuclide by the auto-

	mated analysis, 0 = peak was not associated with a nuclide by the automated analysis (not updated by the ROI interface)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.398. include_stations

Table(s)	GARDS_QAT_QUERY_FILTER (RMSMAN)			
Description	Flag to include or exclude stations			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.399. index_no

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN)			
Description	Index of the energy bin			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		NOT ALLOWED	i10	N
Range				

5.2.400. init_begin

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date				Y
Range				

5.2.401. init_begin_date

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Initialization date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.402. init_detector_id

Table(s)	GARDS_NIC_INIT (RMSMAN)
Description	

Storage type	Unit	NA value	External	Nullable
number				Y
Range				

5.2.403. init_end

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date				Y
Range				

5.2.404. init_end_date

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Decommissioning date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.405. input_file_name

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Input data filename			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(255)			a100	Y
Range				

5.2.406. instance_info

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Determines the current instance information regarding RnMessagesProcessor.pl program. The instance info as follows: USER_ID@machine:ProcessId			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.407. instance_running

Table(s)	PIPELINE_CONTROL (RMSMAN)
Description	Running flag: Y : There is a RUNNING instance. N or NULL : No thing is

	RUNNING			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.408. instance_time

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Determines when the instance started			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date		a21	Y
Range				

5.2.409. interference

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Counts from interference nuclides			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.410. interference_err

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of interference (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.411. interference_used

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Flag indicating whether the interference corrections are used in the sample spectrum analysis. 0 means not used, 1 indicates used, 2 means used if counts are above LC.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range	interference_used IN { 0,1,2}			

5.2.412. intvlid

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Unique interval identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.413. intvlid

Table(s)	GARDS_INTERVAL (RMSAUTO)			
Description	Unique interval identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.414. irf

Table(s)	GARDS_IRF (RMSMAN)			
Description	Isotope response function			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.415. irf_error

Table(s)	GARDS_IRF (RMSMAN)			
Description	Uncertainty of the isotope response function			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.416. kb_sent

Table(s)	GARDS_RECEIPT_LOG (RMSAUTO)			
Description	Transmitted data size of obsolete table			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.417. key_flag

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED

	(RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_XE_NUCL_LINES_LIB (RMSMAN)			
Description	Key line indicator			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.418. lab_detector_id

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Unique laboratory detector identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			a9	Y
Range				

5.2.419. lab_id

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Unique laboratory identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			a5	Y
Range				

5.2.420. lab_receipt_date

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Date of sample receipts at the laboratory			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.421. lab_sample_category

Table(s)	GARDS_LAB_CATEGORY_DESCRIPTION (RMSMAN)			
Description	Report sample category (A|B|C|D|E)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)			a1	N
Range				

5.2.422. lab_sample_category

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Report sample category (A|B|C|D|E)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)			a1	Y
Range				

5.2.423. lab_summary

Table(s)	GARDS_RLR_CONCLUSIONS (RMSAUTO)			
Description	Laboratory findings and conclusions			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.424. last_name

Table(s)	GARDS_POC (RMSMAN)			
Description	Last name of point of contact			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)			a50	Y
Range				

5.2.425. lat

Table(s)	GARDS_DETECTORS (RMSMAN), GARDS_STATIONS (RMSMAN)			
Description	Latitude (degrees)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees		f8.5	Y
Range				

5.2.426. lc

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Critical level in activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq/m3		g8.5	Y
Range				

5.2.427. lc

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN), GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Critical level for the peak (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.428. lc

Table(s)	GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Critical level in activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.429. lc_abscissa

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level, a value of 1.645 corresponds to a 95% confidence interval			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			f8.5	Y
Range				

5.2.430. lc_abscissa

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN), GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Abscissas of the normal distribution corresponding to a predefined confidence level for estimating the critical level, a value of 1.645 corresponds to a 95% confidence interval			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.5	Y
Range				

5.2.431. ld

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Detection limit in activity concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq/m3		g8.5	Y
Range				

5.2.432. lddate

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Load date. The date and time the record was inserted into the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.433. lddate

Table(s)	GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN), GARDS_INTERVAL (RMSAUTO), GARDS_QCHISTORY (RMSAUTO), GARDS_QCHISTORY (RMSMAN), GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN), RMS_MSGDISC (RMSAUTO)			
Description	Load date. The date and time the record was inserted into the database.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.434. left_chan

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Left channel of the peak			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.435. left_fwhm_lim

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)
Description	Left ROI FWHM limit

Storage type	Unit	NA value	External	Nullable
number			g8.5	Y
Range				

5.2.436. line_comment

Table(s)	GARDS_AUX_LINES_LIB (RMSMAN)			
Description	Comment associated with the decay line			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(500)			a150	Y
Range				

5.2.437. line_type

Table(s)	GARDS_AUX_LINES_LIB (RMSMAN)			
Description	Type of the decay line			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

5.2.438. linecomment

Table(s)	GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN)			
Description	Comment associated with the decay line			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1500)			a150	Y
Range				

5.2.439. lon

Table(s)	GARDS_DETECTORS (RMSMAN), GARDS_STATIONS (RMSMAN)			
Description	Longitude (degrees)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	degrees		f8.5	Y
Range				

5.2.440. lower_bound

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN)			
Description	Lower limit of the amount of a nuclide that can be acceptably found			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.441. lower_lim

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN), GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.442. lower_limit

Table(s)	GARDS_CAT_CRITERIA_TESTS (RMSMAN)			
Description	Lower limit bound of the test			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.443. lower_limit_green

Table(s)	GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Green flag lower limit			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.444. lower_limit_yellow

Table(s)	GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Yellow flag lower limit			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.445. lower_roi_number

Table(s)	GARDS_SAMPLE_RATIOS (RMSAUTO), GARDS_SAMPLE_RATIOS (RMSMAN)			
Description	Lower ROI number			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y

Range	

5.2.446. max_acquisition_time

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Maximum duration accepted for acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	seconds		g8.5	Y
Range				

5.2.447. max_collection_time

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Maximum duration accepted for collection			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	seconds		g8.5	Y
Range				

5.2.448. max_mdc

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Maximum MDC value			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq/m3		g8.5	Y
Range				

5.2.449. max_qc_dev

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Maximum deviation of the quality check (integer)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.450. max_reporting_time

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Maximum duration accepted for reporting			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	seconds		g8.5	Y
Range				

5.2.451. max_running

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Determines the maximum number of RUNNING massages that the program can handle at any time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.452. max_sample_id

Table(s)	GARDS_QAT_QUERY_FILTER (RMSMAN)			
Description	QAT user query, upper boundary for the sample id			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.453. max_t_time

Table(s)	GARDS_QAT_CONFIG (RMSMAN)			
Description	Maximum allowable value for t_time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Minutes		g8.5	Y
Range				

5.2.454. mda

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Minimum detectable activity concentration (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.455. mda

Table(s)	GARDS_ROI_CONCS (RMSAUTO), GARDS_ROI_CONCS (RMSMAN)			
Description	Minimum detectable concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y

Range	

5.2.456. mda_acq_start

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Minimum detectable activity at the start of acquisition (mBq)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq		g8.5	Y
Range				

5.2.457. mda_acq_start

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Minimum detectable activity at the start of acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq		g8.5	Y
Range				

5.2.458. mda_err

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the minimum detectable activity concentration (uBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq/m3		g8.5	Y
Range				

5.2.459. mda_err_acq_start

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Uncertainty of the minimum detectable activity at the start of acquisition			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	uBq		g8.5	Y
Range				

5.2.460. mda_level

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN),

	GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	MDA confidence factor (percent)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f8.3	Y
Range				

5.2.461. mda_max

Table(s)	GARDS_MDAS2REPORT (RMSMAN)			
Description	Maximum of the minimum detectable activities			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range	mda >= 0.0			

5.2.462. mda_min

Table(s)	GARDS_MDAS2REPORT (RMSMAN)			
Description	Minimum of the minimum detectable activities			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range	mda >= 0.0			

5.2.463. mdc

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Minimum detectable concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	mBq/m3		g8.5	Y
Range				

5.2.464. mdc

Table(s)	GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Minimum detectable concentration (mBq/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.465. mdc_width

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Baseline width used for MDA calculation			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.466. mdi

Table(s)	GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Minimum detectable intensity for metastable radioxenon (emissions/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	emissions/m3		g8.5	Y
Range				

5.2.467. mdi

Table(s)	GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Minimum detectable intensity for metastable radioxenon (emissions/m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	emissions/m3		g8.5	Y
Range				

5.2.468. measurement_id

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Measurement identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(31)			a31	N
Range				

5.2.469. median_xe131m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Median of the activity concentration for Xe-131m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.470. median_xe133

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Median of the activity concentration for Xe-133			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.471. median_xe133m

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Median of the activity concentration for Xe-133m			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.472. median_xe135

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Median of the activity concentration for Xe-135			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mBq/m3		g8.5	Y
Range				

5.2.473. memory

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Gas background counts			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.474. memory_err

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Gas background (that is, "memory effect") counts.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.475. met_group_id

Table(s)	GARDS_MET_HEADER (RMSAUTO)			
Description	MET group identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.476. met_group_id

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	MET group identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.477. met_id

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	Unique meteorological data identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.478. method

Table(s)	GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Method identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.479. method_id

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN), GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Method identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N

Range	

5.2.480. method_id

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Method identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.481. method_id

Table(s)	GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Method identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(38,0)			i10	N
Range				

5.2.482. method_type

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Method type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.483. min_lookup

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	Minimum library lookup tolerance, keV (typically 0.2)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.484. min_sample_air_flow

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Minimum air flow accepted for sampling			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

float(126)	m3/h		g8.5	Y
Range				

5.2.485. min_sample_id

Table(s)	GARDS_QAT_QUERY_FILTER (RMSMAN)			
Description	QAT user query, lower boundary for the sample id			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.486. min_sample_volume

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Minimum sample volume accepted for collection			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	m3		g8.5	Y
Range				

5.2.487. min_t_time

Table(s)	GARDS_QAT_CONFIG (RMSMAN)			
Description	Minimum allowable value for t_time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Minutes		g8.5	Y
Range				

5.2.488. min_xe_volume

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Minimum xenon volume			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	cm3		g8.5	Y
Range				

5.2.489. moddate

Table(s)	GARDS_ALERTS (RMSAUTO), GARDS_RN_INTERVAL (RMSAUTO), GARDS_SAMPLE_AUX (RMSAUTO), GARDS_SOH_CHAR_DATA (RMSAUTO), GARDS_SOH_HEADER (RMSAUTO), GARDS_SOH_NUM_DATA (RMSAUTO), GARDS_SOH_SENSOR_DATA (RMSAUTO)
Description	Date/time at which the row was last modified

Storage type	Unit	NA value	External	Nullable
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.490. moddate

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_AUX_LIB (RMSMAN), GARDS_AUX_LINES_LIB (RMSMAN), GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN), GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_DETECTOR_STD_SPECTRA (RMSAUTO), GARDS_BG_EFFICIENCY_PAIRS (RMSAUTO), GARDS_BG_EFFICIENCY_PAIRS (RMSMAN), GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN), GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN), GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS_ROI (RMSAUTO), GARDS_BG_PROC_PARAMS_ROI (RMSMAN), GARDS_BG_QC_RESULT (RMSAUTO), GARDS_BG_QC_RESULT (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN), GARDS_BG_STD_SPECTRA (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN), GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_B_RESOLUTION_PAIRS (RMSAUTO), GARDS_B_RESOLUTION_PAIRS (RMSMAN), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSMAN), GARDS_CAT_CRITERIA_TESTS (RMSMAN), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_CODES (RMSMAN), GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN), GARDS_COMMENTS_DEFS (RMSMAN), GARDS_CSC_MODCOEFF_LIB (RMSMAN), GARDS_DATA_LOG (RMSAUTO), GARDS_DBROLE_OWNER (RMSMAN), GARDS_DETECTORS (RMSMAN), GARDS_DIST_SAMPLE_QUEUE (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN), GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN), GARDS_EFFICIENCY_PAIRS (RMSAUTO), GARDS_EFFICIENCY_PAIRS (RMSMAN), GARDS_EFFICIENCY_VGSL_PAIRS (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_COV (RMSAUTO), GARDS_ENERGY_CAL_COV (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO),

	GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_ENERGY_PAIRS (RMSAUTO), GARDS_ENERGY_PAIRS (RMSMAN), GARDS_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_ENVIRONMENT (RMSAUTO), GARDS_FLAGS (RMSAUTO), GARDS_FLAGS (RMSMAN), GARDS_FPE (RMSAUTO), GARDS_FPE (RMSMAN), GARDS_HISTOGRAM (RMSAUTO), GARDS_INTERVAL (RMSAUTO), GARDS_IRF (RMSMAN), GARDS_LAB_CATEGORY_DESCRIPTION (RMSMAN), GARDS_MDAS2REPORT (RMSMAN), GARDS_MET_DATA (RMSAUTO), GARDS_MET_HEADER (RMSAUTO), GARDS_NOTIFY (RMSAUTO), GARDS_NOTIFY (RMSMAN), GARDS_NUCL2QUANTIFY (RMSMAN), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN), GARDS_PERMISSIONS (RMSMAN), GARDS_POC (RMSMAN), GARDS_PROCESSING_ERRORS (RMSMAN), GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_PRODUCT (RMSAUTO), GARDS_PRODUCT (RMSMAN), GARDS_QAT_CONFIG (RMSMAN), GARDS_QAT_NOTIFY (RMSMAN), GARDS_QAT_QUERY_FILTER (RMSMAN), GARDS_QCHISTORY (RMSMAN), GARDS_QCPARAMS (RMSMAN), GARDS_QCTARGETS (RMSMAN), GARDS_QC_RESULTS (RMSAUTO), GARDS_QC_RESULTS (RMSMAN), GARDS_QUERY_RESULTS (RMSAUTO), GARDS_QUERY_RESULTS (RMSMAN), GARDS_RECEIPT_LOG (RMSAUTO), GARDS_REFLINE_MASTER (RMSMAN), GARDS_RELEVANT_NUCLIDES (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN), GARDS_RESOLUTION_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSMAN), GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_PAIRS (RMSAUTO), GARDS_RESOLUTION_PAIRS (RMSMAN), GARDS_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_RESOLUTION_PAIRS_ORIG (RMSMAN), GARDS_RLR (RMSAUTO), GARDS_RLR_CONCLUSIONS (RMSAUTO), GARDS_RLR_OBJECTIVE (RMSAUTO), GARDS_RLR_RATIOS (RMSAUTO), GARDS_RLR_RESULTS (RMSAUTO), GARDS_RLR_SSREB (RMSAUTO), GARDS_ROI_CHANNELS (RMSAUTO), GARDS_ROI_CHANNELS (RMSMAN), GARDS_ROI_CONCS (RMSAUTO), GARDS_ROI_CONCS (RMSMAN), GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN), GARDS_ROI_LIB (RMSMAN), GARDS_ROI_LIMITS (RMSAUTO), GARDS_ROI_LIMITS (RMSMAN), GARDS_ROLES (RMSMAN), GARDS_ROLES_PERMISSIONS (RMSMAN), GARDS_SAINT_DEFAULT_PARAMS (RMSAUTO), GARDS_SAINT_DEFAULT_PARAMS (RMSMAN), GARDS_SAINT_PROCESS_PARAMS (RMSAUTO), GARDS_SAINT_PROCESS_PARAMS (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN), GARDS_SAMPLE_CERT (RMSAUTO), GARDS_SAMPLE_CERT_LINES (RMSAUTO),

	GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN), GARDS_SAMPLE_DESCRIPTION (RMSAUTO), GARDS_SAMPLE_FLAGS (RMSAUTO), GARDS_SAMPLE_FLAGS (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN), GARDS_SAMPLE_RATIOS (RMSAUTO), GARDS_SAMPLE_RATIOS (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN), GARDS_SAMPLE_TESTS (RMSMAN), GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_SOH_CODE (RMSMAN), GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN), GARDS_SPECTRUM (RMSAUTO), GARDS_STADET (RMSMAN), GARDS_STATIONS (RMSMAN), GARDS_STATIONS_SCHEDULE (RMSMAN), GARDS_STATION_ASSIGNMENTS (RMSMAN), GARDS_STATUS_HISTORY (RMSMAN), GARDS_TOTAL_EFFIC (RMSAUTO), GARDS_TOTAL_EFFIC (RMSMAN), GARDS_TRENDVUE (RMSAUTO), GARDS_TRENDVUE (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN), GARDS_UPDATE_REFLINES (RMSMAN), GARDS_USERENV (RMSMAN), GARDS_USERS (RMSMAN), GARDS_USERS_ROLES (RMSMAN), GARDS_USER_COMMENTS (RMSAUTO), GARDS_USER_COMMENTS (RMSMAN), GARDS_XE_CATEGORY_PARAMS (RMSMAN), GARDS_XE_NUCL_LIB (RMSMAN), GARDS_XE_NUCL_LINES_LIB (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_XE_REFLINE_MASTER (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN), GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN), GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN), RMS_MSGDISC (RMSAUTO), RN_LOGS (RMSMAN)			
Description	Date/time at which the row was last modified			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.491. moddate

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date				Y
Range				

5.2.492. moving_avg_days

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)

Description	Period of the inter-quartile algorithm used for abnormal threshold determination for noble gas sample categorization (365 days as default)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.493. mrp_sample_id

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN)			
Description	The sample_id of the MRP sample to be used			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.494. mrp_used

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN)			
Description	Column that lists the sample_id of the MRP sample in the event that the mrp_used flag has been set to 1.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.495. mse

Table(s)	GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN)			
Description	Mean-squared error of the fit, a measure of how large the residuals are			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.496. msg_id

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Message identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.497. msg_id

Table(s)	GARDS_DATA_LOG (RMSAUTO), GARDS_PROCESSING_ERRORS (RMSMAN)			
Description	Message identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.498. msgid

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Unique message identifier of the RN message. Generated by sequence rms_msg-discid_seq			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.499. muarea

Table(s)	GARDS_QCTARGETS (RMSMAN)			
Description	Target area for each nuclide line energy (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.500. mult

Table(s)	GARDS_BASELINE (RMSMAN)			
Description	Multiplicand in the baseline formula			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		NOT ALLOWED	i10	N
Range				

5.2.501. mult

Table(s)	GARDS_BASELINE (RMSAUTO)			
Description	Multiplicand in the baseline formula			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		NOT ALLOWED	i10	Y
Range				

5.2.502. multiplet

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Multiplet peak indicator			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.503. muwidth

Table(s)	GARDS_QCTARGETS (RMSMAN)			
Description	Target width for each nuclide line energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.504. name

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Workflow field indicating the message type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	N
Range				

5.2.505. name

Table(s)	GARDS_INTERVAL (RMSAUTO)			
Description	Detector/system name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	Y
Range				

5.2.506. name

Table(s)	GARDS_FLAGS (RMSAUTO), GARDS_FLAGS (RMSMAN)			
Description	Name of the Event Screening test.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	Y
Range				

5.2.507. name

Table(s)	GARDS_SAINT_DEFAULT_PARAMS (RMSAUTO), GARDS_SAINT_DEFAULT_PARAMS (RMSMAN), GARDS_SAINT_PROCESS_PARAMS (RMSAUTO), GARDS_SAINT_PROCESS_PARAMS (RMSMAN)			
Description	Parameter name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)			a40	N
Range				

5.2.508. name

Table(s)	GARDS_USERENV (RMSMAN)			
Description	Variable name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(40)			a40	N
Range				

5.2.509. name

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_AUX_LIB (RMSMAN), GARDS_AUX_LINES_LIB (RMSMAN), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_CSC_MODCOEFF_LIB (RMSMAN), GARDS_MDAS2REPORT (RMSMAN), GARDS_NUCL2QUANTIFY (RMSMAN), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_QCTARGETS (RMSMAN), GARDS_RELEVANT_NUCLIDES (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN), GARDS_XE_NUCL_LIB (RMSMAN), GARDS_XE_NUCL_LINES_LIB (RMSMAN)			
Description	Nuclide name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	N
Range				

5.2.510. name

Table(s)	GARDS_NIC (RMSAUTO)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)				N

Range	

5.2.511. name

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_ROI_LIB (RMSMAN)			
Description	Nuclide name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

5.2.512. name

Table(s)	GARDS_NIC (RMSMAN), GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)				Y
Range				

5.2.513. name_d

Table(s)	GARDS_CSC_MODCOEFF_LIB (RMSMAN)			
Description	Daughter nuclide name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

5.2.514. net

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Counts in ROI after processing			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.515. net_area

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN)			
Description	Net area			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.516. net_area_err

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN)			
Description	Uncertainty of the net area			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.517. net_count

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Net number of counts in the ROI			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.518. net_count_err

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of the net number of counts in the ROI			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.519. net_err

Table(s)	GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of net (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		i10	Y
Range				

5.2.520. new_status

Table(s)	GARDS_STATUS_HISTORY (RMSMAN)			
Description	Sample status after change			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.521. nid_confid

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	NID confidence factor (percent)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	percent		f8.3	Y
Range				

5.2.522. nid_flag

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_ROI_CONCS (RMSAUTO), GARDS_ROI_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Flag of whether the xenon isotope is detected or not			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.523. nid_flag

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Nuclide identification indicator. (0=nuclide was not identified by automatic analysis; 1=nuclide was identified by automatic analysis; -1=nuclide was initially identified by automatic analysis, but later rejected because the concentration was <0.0; 2= nuclide was initially identified by automatic analysis, but later removed by the Analyst).			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.524. nms_msgid

Table(s)	NMS_RECEIVED_SOFAR (RMSAUTO)			
Description	Message identifier. Maximum nms_msgid fetched from acq.acq_msg			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.525. nms_msgid

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Corresponding nms_msgid from acq.acq_msg			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)			i10	Y
Range				

5.2.526. nms_pull_frequency

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Time in seconds between reading RN messages from acq.acq_msg table			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.527. no_of_loops

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN)			
Description	Number of loops for this bin of the baseline calculation			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		NOT ALLOWED	i10	N
Range				

5.2.528. notify_list

Table(s)	RN_LOGS (RMSMAN)			
Description	List of notification recipients for this analyst comment.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)			a50	Y
Range				

5.2.529. nuc_comment

Table(s)	GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN)			
Description	Nuclide comment			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1500)			a1500	Y
Range				

5.2.530. nucl_name

Table(s)	GARDS_SAMPLE_CERT_LINES (RMSAUTO)			
Description	Nuclide name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	N
Range				

5.2.531. nucl_name

Table(s)	GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN), GARDS_QAT_NOTIFY (RMSMAN)			
Description	Nuclide name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

5.2.532. nuclide1

Table(s)	GARDS_RLR_RATIOS (RMSAUTO)			
Description	Nuclide 1			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	N
Range				

5.2.533. nuclide2

Table(s)	GARDS_RLR_RATIOS (RMSAUTO)			
Description	Nuclide 2			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	N
Range				

5.2.534. nuclide_id

Table(s)	GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN), GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Nuclide identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N

Range	

5.2.535. nuclide_id

Table(s)	GARDS_BG_PROC_PARAMS_ROI (RMSAUTO), GARDS_BG_PROC_PARAMS_ROI (RMSMAN), GARDS_BG_STD_SPECTRA (RMSAUTO), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_LIB (RMSMAN), GARDS_NUCL_LINES_LIB_ARCHIVE (RMSMAN), GARDS_XE_NUCL_LIB (RMSMAN), GARDS_XE_NUCL_LINES_LIB (RMSMAN)			
Description	Nuclide identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.536. nuclide_lib

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Nuclide library to be used during analysis			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(96)			a96	Y
Range				

5.2.537. nuclide_name

Table(s)	GARDS_RLR_RESULTS (RMSAUTO)			
Description	Nuclide name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	N
Range				

5.2.538. nuclide_name

Table(s)	GARDS_IRF (RMSMAN)			
Description	Nuclide name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y

Range	

5.2.539. num_lines

Table(s)	GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_XE_NUCL_LIB (RMSMAN)			
Description	Number of decay lines in library			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.540. num_samples

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Number of samples used to collect the average and define the standard deviation (RDC algorithm)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.541. num_value1

Table(s)	GARDS_TRENDVUE (RMSAUTO), GARDS_TRENDVUE (RMSMAN)			
Description	First plot value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.542. num_value2

Table(s)	GARDS_TRENDVUE (RMSAUTO), GARDS_TRENDVUE (RMSMAN)			
Description	Second plot value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.543. num_value3

Table(s)	GARDS_TRENDVUE (RMSAUTO), GARDS_TRENDVUE (RMSMAN)			
Description	Third plot value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.544. num_value4

Table(s)	GARDS_TRENDVUE (RMSAUTO), GARDS_TRENDVUE (RMSMAN)			
Description	Fourth plot value.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.545. number_of_decays

Table(s)	GARDS_BG_STD_SPECTRA (RMSAUTO)			
Description	Number of decays for this standard spectra			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.546. offset

Table(s)	GARDS_BG_QC_RESULT (RMSAUTO), GARDS_BG_QC_RESULT (RMSMAN)			
Description	Noise offset in QC spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.547. old_status

Table(s)	GARDS_STATUS_HISTORY (RMSMAN)			
Description	Sample status before change			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.548. ondate

Table(s)	GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN)			
Description	Julian start date of the beta and gamma energy calibration			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i8	N
Range				

5.2.549. orig_sample_ref_id

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Original/entire sample reference identification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	Y
Range				

5.2.550. original_area

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Original peak area (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.551. original_uncer

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Uncertainty of the original peak area (counts)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.552. owner

Table(s)	GARDS_DBROLE_OWNER (RMSMAN)			
Description	Name of database user who owns the RMS roles for the MAR tool			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(12)			a12	Y
Range				

5.2.553. param

Table(s)	GARDS_SOH_CODE (RMSMAN)			
Description	SOH parameter name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	N
Range				

5.2.554. param_code

Table(s)	GARDS_SOH_CHAR_DATA (RMSAUTO), GARDS_SOH_NUM_DATA (RMSAUTO)			
Description	SOH parameter code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.555. param_code

Table(s)	GARDS_SOH_CODE (RMSMAN)			
Description	SOH parameter code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.556. param_display

Table(s)	GARDS_SOH_CODE (RMSMAN)			
Description	Parameter text to be displayed in Trendvue			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.557. param_display_flag

Table(s)	GARDS_SOH_CODE (RMSMAN)			
Description	Flag indicating whether or not a particular parameter is displayed			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.558. pd_mod_flag

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN)			
Description	Flag indicating whether or not the nuclide's activity was modified by the parent/ daughter calculation. A value of 1= nuclide was modified, zero indicates that the nuclide was not modified.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y

Range	

5.2.559. peak

Table(s)	GARDS_NUCL_LINES_IDED_ORIG (RMSMAN)			
Description	Peak identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.560. peak

Table(s)	GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO)			
Description	Peak identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.561. peak_end

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Peak search end (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.562. peak_id

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.563. peak_id

Table(s)	GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN)			
Description	Peak identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			i10	Y
Range				

5.2.564. peak_sense

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Peak search sensitivity			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.565. peak_sig

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak significance, e.g. peak area divided by Lc			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.566. peak_start

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Peak search start (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.567. peak_tol

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak energy tolerance; approximates the energy tolerance at a given peak			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.568. permission_id

Table(s)	GARDS_PERMISSIONS (RMSMAN), GARDS_ROLES_PERMISSIONS

	(RMSMAN)			
Description	Permission identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.569. permission_name

Table(s)	GARDS_PERMISSIONS (RMSMAN)			
Description	Permission name.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	N
Range				

5.2.570. poc_id

Table(s)	GARDS_NOTIFY (RMSAUTO), GARDS_NOTIFY (RMSMAN)			
Description	Point of contact identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.571. pocid

Table(s)	GARDS_POC (RMSMAN), GARDS_RECEIPT_LOG (RMSAUTO)			
Description	Point of contact identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.572. pocid

Table(s)	GARDS_STATIONS (RMSMAN)			
Description	Point of contact identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.573. poll_time

Table(s)	GARDS_QAT_CONFIG (RMSMAN)
Description	Number of minutes rms_QAT sleeps between each database poll

Storage type	Unit	NA value	External	Nullable
number	Minutes		g8.5	Y
Range				

5.2.574. pressure

Table(s)	GARDS_ENVIRONMENT (RMSAUTO)			
Description	Outside pressure (mbar)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mbar		i4	Y
Range				

5.2.575. prev_valid_samples

Table(s)	GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN)			
Description	Number of the previous valid samples			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.576. priority_level

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Priority level (Urgent or Routine)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)			a7	Y
Range				

5.2.577. prodid

Table(s)	GARDS_RECEIPT_LOG (RMSAUTO)			
Description	Product id of obsolete table?			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.578. pss

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Peak search significance parameter			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.579. pull_connection

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	User name and password for DB connection			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.580. pull_host

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Database name to be connected in order to read RN messages (ODB, TODB, or IDCDEV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)			a15	Y
Range				

5.2.581. pull_method

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Currently is DB			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)			a15	Y
Range				

5.2.582. pull_owner_or_queue

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	The user name who owns acq_msg table (default Acq)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.583. q_time

Table(s)	GARDS_QAT_CONFIG (RMSMAN)			
Description	Number of minutes after gards_sample_status.review_date that rms_QAT_auto waits before releasing a sample			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Minutes		g8.5	Y

Range	

5.2.584. qc_b_threshold

Table(s)	GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	The value that is used to mask the gamma spectrum in QC control of the spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.585. qc_b_threshold

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	Threshold for channels in the histogram used when calculating beta and gamma spectra			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.586. qc_comment

Table(s)	GARDS_QC_RESULTS (RMSAUTO), GARDS_QC_RESULTS (RMSMAN)			
Description	Quality control comments			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(255)			a225	Y
Range				

5.2.587. qc_id

Table(s)	GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Unique quality control identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.588. qc_method_id

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN)			
Description	QC analysis method ID - possible values (0,1,2,3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			i10	Y
Range	{0,1,2,3}			

5.2.589. quantity

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Air volume sampled (m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	m3		f8.5	Y
Range				

5.2.590. quantity

Table(s)	GARDS_SAMPLE_CERT (RMSAUTO)			
Description	Total activity of the source			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Bq		g8.5	Y
Range				

5.2.591. rated_efficiency

Table(s)	GARDS_DETECTORS (RMSMAN)			
Description	Efficiency of the detector relative to a standard 3 in. x 3 in. (7.62 cm x 7.62 cm) cylindrical NaI(Tl) detector at 1332 keV. The rated efficiency is usually supplied by the detector manufacturer.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.592. rated_resolution

Table(s)	GARDS_DETECTORS (RMSMAN)			
Description	Rated resolution of detector			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.593. ratio_id

Table(s)	GARDS_SAMPLE_RATIOS (RMSAUTO), GARDS_SAMPLE_RATIOS (RMSMAN)
Description	Ratio identifier

Storage type	Unit	NA value	External	Nullable
varchar2(15)			a15	N
Range				

5.2.594. rec_begin

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date				Y
Range				

5.2.595. rec_end

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date				Y
Range				

5.2.596. recipient_list

Table(s)	GARDS_QAT_NOTIFY (RMSMAN)			
Description	List of recipients for QAT comment notification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(128)			a128	Y
Range				

5.2.597. ref_id

Table(s)	RN_LOGS (RMSMAN)			
Description	Reference to related analyst comment			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.598. reference

Table(s)	GARDS_AUX_LINES_LIB (RMSMAN)			
Description	Reference for line properties			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y

Range	

5.2.599. reference_date

Table(s)	GARDS_RLR_RATIOS (RMSAUTO)			
Description	Reference date of the activity ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.600. refid

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Workflow field indicating the sample id			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	N
Range				

5.2.601. refidtype

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Workflow field indicating whether it is an input interval, PHD, SOH			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	N
Range				

5.2.602. refpeak_energy

Table(s)	GARDS_REFLINE_MASTER (RMSMAN), GARDS_UPDATE_REFLINES (RMSMAN), GARDS_XE_REFLINE_MASTER (RMSMAN)			
Description	Reference energy of the decay line			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.603. release_date

Table(s)	GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Release date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.604. report_mda

Table(s)	GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_ROI_CONCS (RMSAUTO), GARDS_ROI_CONCS (RMSMAN)			
Description	Indicator of whether MDA is to be reported			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.605. report_number

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Report number			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i2	Y
Range				

5.2.606. report_type

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Report type. FIN=Final PRE=Preliminary.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(3)			a3	Y
Range				

5.2.607. rer_intercept

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	Default RER intercept (typically 1.2)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.608. rer_mrp_used

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN)			
Description	Indicates what RER was chosen during analysis			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y

Range	

5.2.609. rer_slope

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	Default RER intercept to be used during calculations.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.3	Y
Range				

5.2.610. res_energy

Table(s)	GARDS_B_RESOLUTION_PAIRS (RMSAUTO), GARDS_B_RESOLUTION_PAIRS (RMSMAN), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSMAN)			
Description	Beta energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.611. res_energy

Table(s)	GARDS_RESOLUTION_PAIRS (RMSAUTO), GARDS_RESOLUTION_PAIRS (RMSMAN), GARDS_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_RESOLUTION_PAIRS_ORIG (RMSMAN)			
Description	Gamma energy (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	N
Range				

5.2.612. res_error

Table(s)	GARDS_B_RESOLUTION_PAIRS (RMSAUTO), GARDS_B_RESOLUTION_PAIRS (RMSMAN), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSMAN), GARDS_RESOLUTION_PAIRS (RMSAUTO), GARDS_RESOLUTION_PAIRS (RMSMAN), GARDS_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_RESOLUTION_PAIRS_ORIG (RMSMAN)			
Description	Uncertainty of the resolution (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y

Range	

5.2.613. resolution

Table(s)	GARDS_B_RESOLUTION_PAIRS (RMSAUTO), GARDS_B_RESOLUTION_PAIRS (RMSMAN), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSMAN)			
Description	Beta resolution (FWHM)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.614. resolution

Table(s)	GARDS_RESOLUTION_PAIRS (RMSAUTO), GARDS_RESOLUTION_PAIRS (RMSMAN), GARDS_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_RESOLUTION_PAIRS_ORIG (RMSMAN)			
Description	Resolution (keV)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	keV		g8.5	Y
Range				

5.2.615. result

Table(s)	GARDS_SAMPLE_FLAGS (RMSAUTO), GARDS_SAMPLE_FLAGS (RMSMAN)			
Description	Indicator of whether or not the test passed			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.616. result_comparison

Table(s)	GARDS_RLR_CONCLUSIONS (RMSAUTO)			
Description	Comparison of IDC and laboratory results			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.617. revid

Table(s)	GARDS_FPE (RMSAUTO), GARDS_FPE (RMSMAN)
Description	Revision number

Storage type	Unit	NA value	External	Nullable
number			i10	Y
Range				

5.2.618. review_cat

Table(s)	GARDS_QAT_QUERY_FILTER (RMSMAN)			
Description	Review category			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.619. review_date

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Date sample was last reviewed			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.620. review_time

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Amount of time to review sample (minutes) (no longer automatically updated)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.621. revision

Table(s)	GARDS_PRODUCT (RMSAUTO)			
Description	Revision identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.622. revision

Table(s)	GARDS_PRODUCT (RMSMAN)			
Description	Revision identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			i10	Y
Range				

5.2.623. right_fwhm_lim

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Right ROI FWHM limit			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.624. rlr_id

Table(s)	GARDS_RLR (RMSAUTO), GARDS_RLR_CONCLUSIONS (RMSAUTO), GARDS_RLR_OBJECTIVE (RMSAUTO), GARDS_RLR_RATIOS (RMSAUTO), GARDS_RLR_RESULTS (RMSAUTO), GARDS_RLR_SSREB (RMSAUTO)			
Description	Unique RLR identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.625. rms_id

Table(s)	GARDS_DATA_LOG (RMSAUTO), GARDS_PROCESSING_ERRORS (RMSMAN)			
Description	Internal identifier for message; either sample_id for PHDs, the rlr_id for RLRs, or the soh_id for SOHs			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.626. roi

Table(s)	GARDS_BG_EFFICIENCY_PAIRS (RMSAUTO), GARDS_BG_EFFICIENCY_PAIRS (RMSMAN), GARDS_BG_PROC_PARAMS_ROI (RMSAUTO), GARDS_BG_PROC_PARAMS_ROI (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN), GARDS_ROI_CHANNELS (RMSAUTO), GARDS_ROI_CHANNELS (RMSMAN), GARDS_ROI_CONCS (RMSAUTO), GARDS_ROI_CONCS (RMSMAN), GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN), GARDS_ROI_LIB (RMSMAN), GARDS_ROI_LIMITS (RMSAUTO), GARDS_ROI_LIMITS

	(RMSMAN), GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_XE_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	ROI identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.627. role_id

Table(s)	GARDS_ROLES (RMSMAN), GARDS_ROLES_PERMISSIONS (RMSMAN), GARDS_USERS_ROLES (RMSMAN)			
Description	Role identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.628. role_id

Table(s)	GARDS_DIST_SAMPLE_QUEUE (RMSMAN)			
Description	Role identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.629. role_name

Table(s)	GARDS_ROLES (RMSMAN)			
Description	Role name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	N
Range				

5.2.630. row_index

Table(s)	GARDS_ENERGY_CAL_COV (RMSAUTO), GARDS_ENERGY_CAL_COV (RMSMAN), GARDS_RESOLUTION_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSMAN)			
Description	Row index			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(10,0)			i10	N
Range				

5.2.631. rqst_id

Table(s)	GARDS_QUERY_RESULTS (RMSAUTO), GARDS_QUERY_RESULTS (RMSMAN)			
Description	Unique query identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.632. rqst_string

Table(s)	GARDS_QUERY_RESULTS (RMSAUTO), GARDS_QUERY_RESULTS (RMSMAN)			
Description	Unique query identifier.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(100)			a100	Y
Range				

5.2.633. sample_act

Table(s)	GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN)			
Description	Spectrum activity without decay			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.634. sample_diameter

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Diameter of the sample			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	cm		f5.1	N
Range				

5.2.635. sample_height

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Diameter of gas chamber.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	cm		f5.1	N
Range				

5.2.636. sample_id

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_BG_DETECTOR_STD_SPECTRA (RMSAUTO), GARDS_BG_EFFICIENCY_PAIRS (RMSAUTO), GARDS_BG_EFFICIENCY_PAIRS (RMSMAN), GARDS_BG_ENERGY_CAL (RMSAUTO), GARDS_BG_ENERGY_CAL (RMSMAN), GARDS_BG_ENERGY_CAL_DIRECTIVE (RMSMAN), GARDS_BG_ENERGY_CAL_ORIG (RMSAUTO), GARDS_BG_ENERGY_CAL_ORIG (RMSMAN), GARDS_BG_ISOTOPE_CONCS (RMSAUTO), GARDS_BG_ISOTOPE_CONCS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS_ROI (RMSAUTO), GARDS_BG_PROC_PARAMS_ROI (RMSMAN), GARDS_BG_QC_RESULT (RMSAUTO), GARDS_BG_QC_RESULT (RMSMAN), GARDS_BG_ROI_CONCS (RMSAUTO), GARDS_BG_ROI_CONCS (RMSMAN), GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN), GARDS_BG_STD_SPECTRA (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN), GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_B_RESOLUTION_PAIRS (RMSAUTO), GARDS_B_RESOLUTION_PAIRS (RMSMAN), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSMAN), GARDS_DIST_SAMPLE_QUEUE (RMSMAN), GARDS_EFFICIENCY_CAL (RMSAUTO), GARDS_EFFICIENCY_CAL (RMSMAN), GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN), GARDS_EFFICIENCY_PAIRS (RMSAUTO), GARDS_EFFICIENCY_PAIRS (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_ENERGY_PAIRS (RMSAUTO), GARDS_ENERGY_PAIRS (RMSMAN), GARDS_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_ENVIRONMENT (RMSAUTO), GARDS_HISTOGRAM (RMSAUTO), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN), GARDS_PRODUCT (RMSAUTO), GARDS_QAT_NOTIFY (RMSMAN), GARDS_QCHISTORY (RMSAUTO), GARDS_QCHISTORY (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN), GARDS_RESOLUTION_PAIRS (RMSAUTO), GARDS_RESOLUTION_PAIRS (RMSMAN), GARDS_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_RESOLUTION_PAIRS_ORIG (RMSMAN), GARDS_RLR_SSREB (RMSAUTO), GARDS_ROI_CHANNELS (RMSAUTO), GARDS_ROI_CHANNELS (RMSMAN), GARDS_ROI_CONCS

	(RMSAUTO), GARDS_ROI_CONCS (RMSMAN), GARDS_ROI_COUNTS (RMSAUTO), GARDS_ROI_COUNTS (RMSMAN), GARDS_ROI_LIMITS (RMSAUTO), GARDS_ROI_LIMITS (RMSMAN), GARDS_SAINT_PROCESS_PARAMS (RMSAUTO), GARDS_SAINT_PROCESS_PARAMS (RMSMAN), GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_AUX (RMSAUTO), GARDS_SAMPLE_CAT (RMSMAN), GARDS_SAMPLE_CERT (RMSAUTO), GARDS_SAMPLE_CERT_LINES (RMSAUTO), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN), GARDS_SAMPLE_DESCRIPTION (RMSAUTO), GARDS_SAMPLE_FLAGS (RMSAUTO), GARDS_SAMPLE_FLAGS (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN), GARDS_SAMPLE_RATIOS (RMSAUTO), GARDS_SAMPLE_RATIOS (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN), GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_SAMPLE_XE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_XE_PROC_PARAMS (RMSMAN), GARDS_SPECTRUM (RMSAUTO), GARDS_TOTAL_EFFIC (RMSAUTO), GARDS_TOTAL_EFFIC (RMSMAN), GARDS_XE_RESULTS (RMSAUTO), GARDS_XE_RESULTS (RMSMAN), GARDS_XE_SAMPLE_CATEGORY (RMSAUTO), GARDS_XE_SAMPLE_CATEGORY (RMSMAN), GARDS_XE_UNCORRECTED_RESULTS (RMSAUTO), GARDS_XE_UNCORRECTED_RESULTS (RMSMAN)			
Description	Unique sample identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.637. sample_id

Table(s)	GARDS_NIC (RMSAUTO)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				N
Range				

5.2.638. sample_id

Table(s)	GARDS_COMMENTS (RMSAUTO), GARDS_COMMENTS (RMSMAN), GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN), GARDS_FPE (RMSAUTO), GARDS_FPE (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_PRODUCT (RMSMAN), GARDS_STATUS_HISTORY (RMSMAN), GARDS_USER_COMMENTS (RMSAUTO), GARDS_USER_COMMENTS

	(RMSMAN), RN_LOGS (RMSMAN)			
Description	Unique sample identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.639. sample_id

Table(s)	GARDS_NIC (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.640. sample_id

Table(s)	GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN)			
Description	Unique sample identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(19,0)			i10	N
Range				

5.2.641. sample_id

Table(s)	GARDS_ENERGY_CAL_COV (RMSAUTO), GARDS_ENERGY_CAL_COV (RMSMAN), GARDS_QC_RESULTS (RMSAUTO), GARDS_QC_RESULTS (RMSMAN), GARDS_RESOLUTION_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSMAN)			
Description	Unique sample identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(22,0)			i10	N
Range				

5.2.642. sample_id

Table(s)	GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN)			
Description	Unique sample identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number(8,0)			i10	N
Range				

5.2.643. sample_ref_id

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Sample reference identification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	N
Range				

5.2.644. sample_ref_id

Table(s)	GARDS_PROCESSING_ERRORS (RMSMAN)			
Description	Sample reference identification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	Y
Range				

5.2.645. sample_status

Table(s)	GARDS_NIC (RMSAUTO)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)				N
Range				

5.2.646. sample_status

Table(s)	GARDS_NIC (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)				Y
Range				

5.2.647. sample_type

Table(s)	GARDS_BASELINE (RMSAUTO), GARDS_BASELINE (RMSMAN), GARDS_RELEVANT_NUCLIDES (RMSMAN), GARDS_SPECTRUM (RMSAUTO)			
Description	Sample type: P = particulate, G = HPGe based noble gas, or B = beta-gamma coincidence based noble gas			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	N
Range	sample_type IN {P, G, B}			

5.2.648. sample_type

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN), GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Sample type: P = particulate, G = HPGe based noble gas, or B = beta-gamma coincidence based noble gas			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range	sample_type IN {P, G, B}			

5.2.649. sample_type

Table(s)	GARDS_MDAS2REPORT (RMSMAN)			
Description	Sample type: P = particulate, G = HPGe based noble gas, or B = beta-gamma coincidence based noble gas			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)			a1	N
Range	sample_type IN {P, G, B}			

5.2.650. sample_type

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN)			
Description	Sample type: P = particulate, G = HPGe based noble gas, or B = beta-gamma coincidence based noble gas			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)			a1	Y
Range	sample_type IN {P, G, B}			

5.2.651. scan_msgtsd

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Y: scan ONCE the directory /$lan/data/shared/messages/rn_manual/msg_TSD for manually processing RN messages and after it finishes scanning it converts the value to N. N: Do not scan the directory			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.652. score

Table(s)	GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN)
Description	Score used when comparing this ECR to other ECRs, a measure of both the ape

	and determinant			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.653. sdmult

Table(s)	GARDS_QCPARAMS (RMSMAN)			
Description	Location test standard deviation multiplier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.654. sensor_name

Table(s)	GARDS_SOH_SENSOR_DATA (RMSAUTO)			
Description	Unique sensor name to distinguish between sensors of the same type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)			a20	N
Range				

5.2.655. sensor_type

Table(s)	GARDS_SOH_SENSOR_DATA (RMSAUTO)			
Description	Sensor type (TEMP, PRESSURE, PROCESSFLOW, VOLTAGE, COUNTRATES, DEWPOINT, CO2VOLUME)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)			a20	N
Range	sensor_type IN { TEMP, PRESSURE, PROCESSFLOW, VOLTAGE, COUNTRATES, DEWPOINT, CO2VOLUME }			

5.2.656. session_id

Table(s)	GARDS_TRENDVUE (RMSAUTO), GARDS_TRENDVUE (RMSMAN)			
Description	Unique session identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.657. severity

Table(s)	RN_LOGS (RMSMAN)
Description	Severity level null, or between 1 and 4 for this analyst comment

Storage type	Unit	NA value	External	Nullable
number			i10	Y
Range				

5.2.658. shortterm_moving_avg_days

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	period in which the average activity concentration is determined for a short term abnormal threshold determination (30 days as default)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.659. site_det_code

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Station code concatenated with detector code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(15)			a15	Y
Range				

5.2.660. sleep_time

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Determines the sleep time in seconds before processing RECEIVED messages			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.661. slope

Table(s)	GARDS_BG_QC_RESULT (RMSAUTO), GARDS_BG_QC_RESULT (RMSMAN)			
Description	Noise slope in QC spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)			g8.5	Y
Range				

5.2.662. soh_char_id

Table(s)	GARDS_SOH_CHAR_DATA (RMSAUTO)
Description	Unique state of health character data identifier

Storage type	Unit	NA value	External	Nullable
number			i10	N
Range				

5.2.663. soh_flag

Table(s)	GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN)			
Description	SOH flag: B, F, G			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	Y
Range				

5.2.664. soh_id

Table(s)	GARDS_SOH_CHAR_DATA (RMSAUTO), GARDS_SOH_HEADER (RMSAUTO), GARDS_SOH_NUM_DATA (RMSAUTO), GARDS_SOH_SENSOR_DATA (RMSAUTO)			
Description	SOH identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.665. soh_num_id

Table(s)	GARDS_SOH_NUM_DATA (RMSAUTO)			
Description	State of health identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.666. soh_sensor_id

Table(s)	GARDS_SOH_SENSOR_DATA (RMSAUTO)			
Description	State of health sensor id			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.667. soh_status

Table(s)	GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN)

Description	Processing status: B, P, V			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1)			a1	Y
Range				

5.2.668. special_instructions

Table(s)	GARDS_RLR_OBJECTIVE (RMSAUTO)			
Description	Special instructions			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.669. spectral_qualifier

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_REFLINE_MASTER (RMSMAN), GARDS_UPDATE_REFLINES (RMSMAN), GARDS_XE_REFLINE_MASTER (RMSMAN)			
Description	Indicator of full or preliminary spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(5)			a5	N
Range				

5.2.670. spectral_qualifier

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Indicator of full or preliminary spectrum			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(5)			a5	Y
Range				

5.2.671. split_sample_ref_id

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Split sample reference identification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	Y
Range				

5.2.672. split_station

Table(s)	GARDS_STATIONS (RMSMAN)

Description	Flag indicating whether or not station is split			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.673. squant_err

Table(s)	GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_SAMPLE_PROC_PARAMS (RMSAUTO), GARDS_SAMPLE_PROC_PARAMS (RMSMAN)			
Description	Uncertainty of sample quantity (m3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	m3		g8.5	Y
Range				

5.2.674. srid

Table(s)	GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN)			
Description	Sample reference identification			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	N
Range				

5.2.675. sta_msgid

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Corresponding station message ID from the RN message			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)			i10	Y
Range				

5.2.676. start_channel

Table(s)	GARDS_SPECTRUM (RMSAUTO)			
Description	Start channel			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.677. start_date

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)

Description	Initialization date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.678. start_time

Table(s)	GARDS_MET_DATA (RMSAUTO), GARDS_MET_HEADER (RMSAUTO)			
Description	Start time of the period			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.679. start_time

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Start time of the period			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.680. start_time

Table(s)	GARDS_INTERVAL (RMSAUTO)			
Description	Start time of the period			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(53)	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.681. state

Table(s)	GARDS_QCHISTORY (RMSAUTO), GARDS_QCHISTORY (RMSMAN)			
Description	Text results of quality control hypothesis tests			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(128)			a128	N
Range				

5.2.682. state

Table(s)	GARDS_RN_INTERVAL (RMSAUTO)			
Description	Workflow field indicating the status			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(16)			a16	N
Range				

5.2.683. state

Table(s)	GARDS_INTERVAL (RMSAUTO)			
Description	Date of data sampling commencement.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	Y
Range				

5.2.684. station_code

Table(s)	GARDS_ALERTS (RMSAUTO)			
Description	Station code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(5)			a5	Y
Range				

5.2.685. station_code

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Station code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(10)			a5	Y
Range				

5.2.686. station_code

Table(s)	RN_LOGS (RMSMAN)			
Description	Station code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)			a5	Y
Range				

5.2.687. station_code

Table(s)	GARDS_PROCESSING_ERRORS (RMSMAN), GARDS_RN_INTERVAL (RMSAUTO), GARDS_SAMPLE (RMSMAN), GARDS_STATIONS (RMSMAN)			
Description	Station code			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(5)			a5	Y

Range	

5.2.688. station_id

Table(s)	GARDS_ALERTS (RMSAUTO), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_EFFICIENCY_IDC_PAIRS (RMSAUTO), GARDS_EFFICIENCY_IDC_PAIRS (RMSMAN), GARDS_PROC_PARAMS_TEMPLATE (RMSMAN), GARDS_QCTARGETS (RMSMAN), GARDS_SOH_CHAR_DATA (RMSAUTO), GARDS_SOH_HEADER (RMSAUTO), GARDS_SOH_NUM_DATA (RMSAUTO), GARDS_SOH_SENSOR_DATA (RMSAUTO), GARDS_STADET (RMSMAN), GARDS_STATIONS (RMSMAN), GARDS_STATIONS_SCHEDULE (RMSMAN), GARDS_STATION_ASSIGNMENTS (RMSMAN), GARDS_UPDATE_REFLINES (RMSMAN)			
Description	Unique station identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.689. station_id

Table(s)	GARDS_DATA_LOG (RMSAUTO), GARDS_MET_DATA (RMSAUTO), GARDS_MET_HEADER (RMSAUTO), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LINES_IDED (RMSAUTO), GARDS_NUCL_LINES_IDED (RMSMAN), GARDS_NUCL_LINES_IDED_ORIG (RMSAUTO), GARDS_NUCL_LINES_IDED_ORIG (RMSMAN), GARDS_QCHISTORY (RMSAUTO), GARDS_QCHISTORY (RMSMAN), GARDS_RLR (RMSAUTO), GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN), GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Unique station identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.690. station_id

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN), GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.691. station_list

Table(s)	GARDS_QAT_QUERY_FILTER (RMSMAN)			
Description	Comma-separated list of stations			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(1024)			a128	Y
Range				

5.2.692. status

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_STATUS (RMSAUTO), GARDS_SAMPLE_STATUS (RMSMAN)			
Description	Current processing status of sample. Some possible values are: A- analyzed, P- processed, R- reviewed, U- unprocessed, V- viewed, F xxx, D xxx, B xxx, Z - xxx			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range	status IN {A, P, R, U, V, F, D, B, Z}			

5.2.693. status

Table(s)	GARDS_DATA_LOG (RMSAUTO)			
Description	Status			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.694. status

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Status of the RN message. Values: SKIPPED, RECEIVED, QUEUED, RUNNING, L-RUNNING, DONE, PARTIAL_DONE or FAILED.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)			a15	Y
Range				

5.2.695. status

Table(s)	GARDS_DETECTORS (RMSMAN)			
Description	Status of detector: if Y, then data from this detector are automatically processed; if N, then they are not processed automatically.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)			a2	Y

Range	status IN {N, Y}

5.2.696. status

Table(s)	GARDS_STATIONS (RMSMAN)			
Description	Status of station: if NULL, then station is fully operational, otherwise status is I.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2)			a2	Y
Range	status IN {I}			

5.2.697. status_list

Table(s)	GARDS_QAT_QUERY_FILTER (RMSMAN)			
Description	Comma-separated list of statuses			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(128)			a128	Y
Range				

5.2.698. std_spectra_id

Table(s)	GARDS_BG_STD_SPECTRA_RESULT (RMSAUTO), GARDS_BG_STD_SPECTRA_RESULT (RMSMAN)			
Description	Unique identifier for the standard spectra. This is a sample id for the std spectra)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.699. stop_acquisition

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Y to stop/exit program RnMessagesProcessor.pl			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.700. stop_pipeline

Table(s)	PIPELINE_CONTROL (RMSMAN)			
Description	Y to stop calling rms_pipeline but the program will continue read messages from acq.acq_msg table			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.701. subid

Table(s)	GARDS_RECEIPT_LOG (RMSAUTO)			
Description	Subscription id of obsolete table?			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.702. subject

Table(s)	RN_LOGS (RMSMAN)			
Description	Subject of the analyst comment			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(50)			a50	Y
Range				

5.2.703. subject

Table(s)	RMS_MSGDISC (RMSAUTO)			
Description	Subject of the RN message extracted of the mail header of the message			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(64)			a64	Y
Range				

5.2.704. sum_corr

Table(s)	GARDS_IRF (RMSMAN)			
Description	Summing cascade correction factor			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number		NOT ALLOWED	f10.8	Y
Range				

5.2.705. t_time

Table(s)	GARDS_QAT_CONFIG (RMSMAN)			
Description	Number of minutes after gards_sample_status.entry_date that rms_QAT_auto waits before releasing a sample			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	Minutes		g8.5	Y
Range				

5.2.706. telephone

Table(s)	GARDS_POC (RMSMAN)			
Description	Telephone number of point of contact			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(20)			a20	Y
Range				

5.2.707. temp_count

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Net number of counts in the ROI by subtracting the detector background and interference corrections			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.708. temp_count_err

Table(s)	GARDS_BG_ROI_COUNTS (RMSAUTO), GARDS_BG_ROI_COUNTS (RMSMAN)			
Description	Uncertainty of the net number of counts in the ROI by subtracting the detector background and interference corrections			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	counts		g8.5	Y
Range				

5.2.709. temperature

Table(s)	GARDS_ENVIRONMENT (RMSAUTO)			
Description	Outside temperature (degrees Celsius)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f5.1	Y
Range				

5.2.710. temporal_unit

Table(s)	GARDS_STATIONS_SCHEDULE (RMSMAN)			
Description	Units for schedule period - currently only the value DAYS is supported			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(5)	days		a5	Y
Range	temporal_unit IN {DAYS}			

5.2.711. temporal_value

Table(s)	GARDS_STATIONS_SCHEDULE (RMSMAN)			
Description	Units for schedule period - currently only the value DAYS is supported.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	days		i10	Y
Range				

5.2.712. test

Table(s)	GARDS_FLAGS (RMSAUTO), GARDS_FLAGS (RMSMAN)			
Description	Calculated result of the test; this value is compared to the threshold value (examples are <, = , >, >=)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(8)			a8	Y
Range				

5.2.713. test_code

Table(s)	GARDS_CAT_CRITERIA_TESTS (RMSMAN), GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Unique test code identifier string			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(24)			a24	Y
Range				

5.2.714. test_date

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Test completion date at lab			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.715. test_id

Table(s)	GARDS_CAT_CRITERIA_TESTS (RMSMAN), GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Unique identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.716. test_name

Table(s)	GARDS_QC_RESULTS (RMSAUTO), GARDS_QC_RESULTS (RMSMAN)			
Description	Quality control test name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	N
Range				

5.2.717. test_name

Table(s)	GARDS_CAT_CRITERIA_TESTS (RMSMAN), GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Test name label for display			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(48)			a48	Y
Range				

5.2.718. test_type

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Type of test performed			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a40	Y
Range				

5.2.719. tests_authorized

Table(s)	GARDS_RLR_OBJECTIVE (RMSAUTO)			
Description	Tests authorized			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.720. threshold

Table(s)	GARDS_FLAGS (RMSAUTO), GARDS_FLAGS (RMSMAN)			
Description	Value against which the results in the test column are compared			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.721. *total_rainfall*

Table(s)	GARDS_MET_DATA (RMSAUTO)			
Description	Total rainfall (mm)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	mm		f5.1	Y
Range				

5.2.722. *transmit_date*

Table(s)	GARDS_RLR (RMSAUTO)			
Description	Transmit date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.723. *transmit_dtg*

Table(s)	GARDS_SOH_HEADER (RMSAUTO)			
Description	Transmit date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	N
Range	any valid ORACLE date			

5.2.724. *transmit_dtg*

Table(s)	GARDS_SAMPLE (RMSMAN), GARDS_SAMPLE_DATA (RMSAUTO), GARDS_SAMPLE_DATA (RMSMAN)			
Description	Transmit date and time			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.725. *tstat*

Table(s)	GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN)			
Description	Percentage point of the t-statistic for Type I error with a probability set to 99%			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.726. tstat

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Value of a variable used in bounds estimation			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.727. tstat

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.728. type

Table(s)	GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_B_RESOLUTION_PAIRS (RMSAUTO), GARDS_B_RESOLUTION_PAIRS (RMSMAN), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_COV (RMSAUTO), GARDS_ENERGY_CAL_COV (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_ENERGY_PAIRS (RMSAUTO), GARDS_ENERGY_PAIRS (RMSMAN), GARDS_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN), GARDS_RESOLUTION_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSMAN), GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_PAIRS (RMSAUTO), GARDS_RESOLUTION_PAIRS (RMSMAN), GARDS_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_RESOLUTION_PAIRS_ORIG (RMSMAN)			
Description	Calibration type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(8)			a8	N
Range				

5.2.729. type

Table(s)	GARDS_EFFICIENCY_ERROR_CAL (RMSAUTO), GARDS_EFFICIENCY_ERROR_CAL (RMSMAN)			
Description	Uncertainty type: up_err or low_err			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(8)			a8	N
Range				

5.2.730. type

Table(s)	GARDS_CODES (RMSMAN)			
Description	Category of code.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(15)			a15	N
Range				

5.2.731. type

Table(s)	GARDS_MDAS2REPORT (RMSMAN), GARDS_NUCL_IDED (RMSAUTO), GARDS_NUCL_IDED (RMSMAN), GARDS_NUCL_IDED_ORIG (RMSAUTO), GARDS_NUCL_IDED_ORIG (RMSMAN), GARDS_NUCL_LIB (RMSMAN), GARDS_NUCL_LIB_ARCHIVE (RMSMAN), GARDS_RELEVANT_NUCLIDES (RMSMAN), GARDS_XE_NUCL_LIB (RMSMAN)			
Description	Nuclide type of ACTIVATION, COSMIC, FISSION(G), FISSION(P), or NATURAL.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	Y
Range				

5.2.732. type

Table(s)	GARDS_RECEIPT_LOG (RMSAUTO)			
Description	Acknowledgment type of obsolete table?			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(5)			a5	Y
Range				

5.2.733. type

Table(s)	GARDS_DETECTORS (RMSMAN)			
Description	Detector type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

varchar2(6)			a6	Y
Range				

5.2.734. type

Table(s)	GARDS_STATIONS (RMSMAN)			
Description	Station type			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(6)			a6	Y
Range				

5.2.735. typeid

Table(s)	GARDS_PRODUCT (RMSAUTO)			
Description	Unique type identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N
Range				

5.2.736. typeid

Table(s)	GARDS_PRODUCT (RMSMAN)			
Description	Unique type identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.737. unit

Table(s)	GARDS_SOH_CODE (RMSMAN)			
Description	Unit in which the parameter is stored			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(32)			a32	Y
Range				

5.2.738. units

Table(s)	GARDS_CAT_CRITERIA_TESTS (RMSMAN), GARDS_FLAGS (RMSAUTO), GARDS_FLAGS (RMSMAN), GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Units of test bounds			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(16)			a16	Y

Range	

5.2.739. upper_bound

Table(s)	GARDS_AUTO_SAMPLE_CAT (RMSAUTO), GARDS_CAT_TEMPLATE (RMSMAN), GARDS_SAMPLE_CAT (RMSMAN)			
Description	Upper limit of the amount of a nuclide that can be acceptably found			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.740. upper_lim

Table(s)	GARDS_NIC (RMSAUTO), GARDS_NIC (RMSMAN), GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y
Range				

5.2.741. upper_limit

Table(s)	GARDS_CAT_CRITERIA_TESTS (RMSMAN)			
Description	Upper limit bound of the test			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.742. upper_limit_green

Table(s)	GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Green flag upper limit			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.743. upper_limit_yellow

Table(s)	GARDS_SAMPLE_TESTS (RMSMAN)			
Description	Yellow flag upper limit			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.744. upper_roi_number

Table(s)	GARDS_SAMPLE_RATIOS (RMSAUTO), GARDS_SAMPLE_RATIOS (RMSMAN)			
Description	Upper ROI number			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.745. use_mult

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	0/1 flag for use of multiplets in ECR updating (typically 1)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.746. use_weight

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	0/1 flag for use of weights in ECR updating (typically 1)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i1	Y
Range				

5.2.747. user_comment

Table(s)	GARDS_SOH_EVALUATION (RMSAUTO), GARDS_SOH_EVALUATION (RMSMAN)			
Description	Comment			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(2048)			a2048	Y
Range				

5.2.748. user_id

Table(s)	GARDS_USERS (RMSMAN), GARDS_USERS_ROLES (RMSMAN)			
Description	User identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	N

Range	

5.2.749. user_id

Table(s)	GARDS_DIST_SAMPLE_QUEUE (RMSMAN), GARDS_QAT_QUERY_FILTER (RMSMAN), GARDS_STATION_ASSIGNMENTS (RMSMAN)			
Description	User identifier			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.2.750. user_name

Table(s)	GARDS_USERS (RMSMAN)			
Description	User name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	N
Range				

5.2.751. user_name

Table(s)	GARDS_STATUS_HISTORY (RMSMAN)			
Description	User name			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			a30	Y
Range				

5.2.752. value

Table(s)	GARDS_SOH_NUM_DATA (RMSAUTO), GARDS_SOH_SENSOR_DATA (RMSAUTO)			
Description	Value of the specified SOH parameter			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			f8.5	N
Range				

5.2.753. value

Table(s)	GARDS_SAMPLE_FLAGS (RMSAUTO), GARDS_SAMPLE_FLAGS (RMSMAN)			
Description	Threshold value for the test.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.754. value

Table(s)	GARDS_SAINT_DEFAULT_PARAMS (RMSAUTO), GARDS_SAINT_DEFAULT_PARAMS (RMSMAN)			
Description	Parameter default setting			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(255)			a256	N
Range				

5.2.755. value

Table(s)	GARDS_SAINT_PROCESS_PARAMS (RMSAUTO), GARDS_SAINT_PROCESS_PARAMS (RMSMAN)			
Description	Parameter setting			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(255)			a256	Y
Range				

5.2.756. value

Table(s)	GARDS_USERENV (RMSMAN)			
Description	Variable value			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(256)			a256	N
Range				

5.2.757. value

Table(s)	GARDS_SOH_CHAR_DATA (RMSAUTO)			
Description	Value of the specified SOH parameter			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
varchar2(30)			f8.5	N
Range				

5.2.758. width

Table(s)	GARDS_PEAKS (RMSAUTO), GARDS_PEAKS (RMSMAN), GARDS_PEAKS_ORIG (RMSAUTO), GARDS_PEAKS_ORIG (RMSMAN)			
Description	Width of the peak (channels)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.759. width_adj

Table(s)	GARDS_QCPARAMS (RMSMAN)			
Description	Peak width standard deviation inflation factor			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.760. winner

Table(s)	GARDS_B_ENERGY_PAIRS (RMSAUTO), GARDS_B_ENERGY_PAIRS (RMSMAN), GARDS_B_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_B_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_B_RESOLUTION_CAL (RMSAUTO), GARDS_B_RESOLUTION_CAL (RMSMAN), GARDS_B_RESOLUTION_PAIRS (RMSAUTO), GARDS_B_RESOLUTION_PAIRS (RMSMAN), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_B_RESOLUTION_PAIRS_ORIG (RMSMAN), GARDS_ENERGY_CAL (RMSAUTO), GARDS_ENERGY_CAL (RMSMAN), GARDS_ENERGY_CAL_COV (RMSAUTO), GARDS_ENERGY_CAL_COV (RMSMAN), GARDS_ENERGY_CAL_ORIG (RMSAUTO), GARDS_ENERGY_CAL_ORIG (RMSMAN), GARDS_ENERGY_PAIRS (RMSAUTO), GARDS_ENERGY_PAIRS (RMSMAN), GARDS_ENERGY_PAIRS_ORIG (RMSAUTO), GARDS_ENERGY_PAIRS_ORIG (RMSMAN), GARDS_RESOLUTION_CAL (RMSAUTO), GARDS_RESOLUTION_CAL (RMSMAN), GARDS_RESOLUTION_CAL_COV (RMSAUTO), GARDS_RESOLUTION_CAL_COV (RMSMAN), GARDS_RESOLUTION_CAL_ORIG (RMSAUTO), GARDS_RESOLUTION_CAL_ORIG (RMSMAN), GARDS_RESOLUTION_PAIRS (RMSAUTO), GARDS_RESOLUTION_PAIRS (RMSMAN), GARDS_RESOLUTION_PAIRS_ORIG (RMSAUTO), GARDS_RESOLUTION_PAIRS_ORIG (RMSMAN)			
Description	Flag indicating whether the calibration type is used (Y) or not (N)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
char(1)			a1	Y
Range				

5.2.761. xe133_xe131m_limit

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Ratio limit of Xe-133 to Xe-131m, related to the categorization flag using the Bayesian ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	Y
Range				

5.2.762. xe133m_xe131m_limit

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Ratio limit of Xe-133m to Xe-131m, related to the categorization flag using the Bayesian ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.763. xe133m_xe133_limit

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Ratio limit of Xe-133m to Xe-133, related to the categorization flag using the Bayesian ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.764. xe135_xe133_limit

Table(s)	GARDS_XE_CATEGORY_PARAMS (RMSMAN)			
Description	Ratio limit of Xe-135 to Xe-133, related to the categorization flag using the Bayesian ratio			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.765. xe_collect_yield

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Xenon collection yield of the system			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	N
Range				

5.2.766. xe_collect_yield_err

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Uncertainty of the xenon collection yield			
Storage type	**Unit**	**NA value**	**External**	**Nullable**

number			g8.5	N
Range				

5.2.767. xe_in_air

Table(s)	GARDS_BG_CONFIG_PARAMS (RMSMAN), GARDS_BG_PROC_PARAMS (RMSAUTO), GARDS_BG_PROC_PARAMS (RMSMAN)			
Description	Volume of stable xenon (ml) in the 1 m3 air			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
float(126)	cm3/m3		g8.5	Y
Range				

5.2.768. xe_volume

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Xenon volume (cm3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	cm3		g8.5	N
Range				

5.2.769. xe_volume_err

Table(s)	GARDS_SAMPLE_AUX (RMSAUTO)			
Description	Uncertainty of the xenon volume (cm3)			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number	cm3		g8.5	N
Range				

5.2.770. xform

Table(s)	GARDS_CAT_TEMPLATE (RMSMAN)			
Description	Code for transform function applied to measurement data			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			g8.5	Y
Range				

5.2.771. xform

Table(s)	GARDS_NIC_INIT (RMSMAN)			
Description				
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number				Y

Range	

5.2.772. zero_date

Table(s)	GARDS_RLR_RATIOS (RMSAUTO)			
Description	Zero date			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
date	Date	NOT ALLOWED	a21	Y
Range	any valid ORACLE date			

5.2.773. zeroshift

Table(s)	GARDS_SAMPLE_UPDATE_PARAMS (RMSAUTO), GARDS_SAMPLE_UPDATE_PARAMS (RMSMAN), GARDS_UPDATE_PARAMS_TEMPLATE (RMSMAN)			
Description	Change in channel number at 0 keV between one spectrum and the previous spectrum.			
Storage type	**Unit**	**NA value**	**External**	**Nullable**
number			i10	Y
Range				

5.3. Radionuclide Trigger Descriptions

5.3.1. ALERTS_B_I_U

Table	GARDS_ALERTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.2. ALERTS_S_I

Table	GARDS_ALERTS
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`declare`
	`new_id number;`
	`begin`
	`select alert_seq.nextval into new_id from dual;`
	`:new.alert_id := new_id;`
	`end;`

5.3.3. AUTO_SAMPLE_CAT_B_I_U

Table	GARDS_AUTO_SAMPLE_CAT
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.4. AUX_LIB_B_I_U

Table	GARDS_AUX_LIB
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.5. AUX_LINES_LIB_B_I_U

Table	GARDS_AUX_LINES_LIB

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.6. BASELINE_B_I_U

Table	GARDS_BASELINE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.7. BASELINE_B_I_U

Table	GARDS_BASELINE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.8. BG_CONFIG_PARAMS_B_I_U

Table	GARDS_BG_CONFIG_PARAMS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.9. BG_DET_STD_SPECTRA_B_I_U

Table	GARDS_BG_DETECTOR_STD_SPECTRA
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.10. BG_EFFICIENCY_PAIRS_B_I_U

Table	GARDS_BG_EFFICIENCY_PAIRS

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.11. BG_ENERGY_CAL_B_I_U

Table	GARDS_BG_ENERGY_CAL
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.12. BG_ENERGY_CAL_DIR_B_I_U

Table	GARDS_BG_ENERGY_CAL_DIRECTIVE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.13. BG_ENERGY_CAL_ORIG_B_I_U

Table	GARDS_BG_ENERGY_CAL_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.14. BG_ISOTOPE_CONCS_B_I_U

Table	GARDS_BG_ISOTOPE_CONCS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.15. BG_PROC_PARAMS_B_I_U

Table	GARDS_BG_PROC_PARAMS

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.16. BG_PROC_PARAMS_ROI_B_I_U

Table	GARDS_BG_PROC_PARAMS_ROI
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.17. BG_QC_RESULT_B_I_U

Table	GARDS_BG_QC_RESULT
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.18. BG_ROI_CONCS_B_I_U

Table	GARDS_BG_ROI_CONCS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.19. BG_ROI_COUNTS_B_I_U

Table	GARDS_BG_ROI_COUNTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.20. BG_STD_SPECTRA_B_I_U

Table	GARDS_BG_STD_SPECTRA

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.21. BG_STD_SPECTRA_RESULT_B_I_U

Table	GARDS_BG_STD_SPECTRA_RESULT
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.3.22. BG_STD_SPECTRA_RESULT_I_U_D

Table	GARDS_BG_STD_SPECTRA_RESULT
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	`begin`
	`      IF INSERTING THEN`
	`      INSERT into rmsman.gards_bg_std_spectra_result (`
	`            SAMPLE_ID              ,`
	`            STD_SPECTRA_ID         ,`
	`            GAMMA_COEFF1           ,`
	`            GAMMA_COEFF2           ,`
	`            GAMMA_COEFF3           ,`
	`            BETA_COEFF1            ,`
	`            BETA_COEFF2            ,`
	`            BETA_COEFF3            ,`
	`            ESTIMATE               ,`
	`            ERROR`
	`            )`
	`      VALUES (`
	`            :new.SAMPLE_ID              ,`
	`            :new.STD_SPECTRA_ID         ,`
	`            :new.GAMMA_COEFF1           ,`
	`            :new.GAMMA_COEFF2           ,`
	`            :new.GAMMA_COEFF3           ,`
	`            :new.BETA_COEFF1            ,`
	`            :new.BETA_COEFF2            ,`
	`            :new.BETA_COEFF3            ,`
	`            :new.ESTIMATE               ,`
	`            :new.ERROR`
	`            );`
	`      ELSIF UPDATING THEN`
	`      UPDATE rmsman.gards_bg_std_spectra_result set`
	`            SAMPLE_ID = :new.SAMPLE_ID              ,`

	STD_SPECTRA_ID = :new.STD_SPECTRA_ID ,
	GAMMA_COEFF1 = :new.GAMMA_COEFF1 ,
	GAMMA_COEFF2 = :new.GAMMA_COEFF2 ,
	GAMMA_COEFF3 = :new.GAMMA_COEFF3 ,
	BETA_COEFF1 = :new.BETA_COEFF1 ,
	BETA_COEFF2 = :new.BETA_COEFF2 ,
	BETA_COEFF3 = :new.BETA_COEFF3 ,
	ESTIMATE = :new.ESTIMATE ,
	ERROR = :new.ERROR
	where sample_id = :old.sample_id;
	ELSE
	delete from rmsman.gards_bg_std_spectra_result
	where sample_id = :old.sample_id;
	END IF;
	end;

5.3.23. B_ENERGY_PAIRS_B_I_U

Table	GARDS_B_ENERGY_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.24. B_ENERGY_PAIRS_ORIG_B_I_U

Table	GARDS_B_ENERGY_PAIRS_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.25. B_RESOLUTION_PAIRS_B_I_U

Table	GARDS_B_RESOLUTION_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.26. B_RESOLUTION_PAIRS_ORIG_B_I_U

Table	GARDS_B_RESOLUTION_PAIRS_ORIG
Type	BEFORE EACH ROW

Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.27. CAT_CRITERIA_B_I_U

Table	GARDS_CAT_CRITERIA_TESTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.28. CAT_TEMPLATE_B_I_U

Table	GARDS_CAT_TEMPLATE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.29. CODES_B_I_U

Table	GARDS_CODES
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.30. COMMENTS_B_I_U

Table	GARDS_COMMENTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.31. COMMENTS_DEFS_B_I_U

Table	GARDS_COMMENTS_DEFS
Type	BEFORE EACH ROW

Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.32. CSC_MODCOEFF_LIB_B_I_U

Table	GARDS_CSC_MODCOEFF_LIB
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.33. DATA_LOG_B_I_U

Table	GARDS_DATA_LOG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.34. DBROLE_OWNER_B_I_U

Table	GARDS_DBROLE_OWNER
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.35. DEFAULT_PARAMS_B_I_U

Table	GARDS_SAINT_DEFAULT_PARAMS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.36. DETECTORS_B_I_U

Table	GARDS_DETECTORS
Type	BEFORE EACH ROW

Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.37. DIST_SAMPLE_QUEUE_B_I_U

Table	GARDS_DIST_SAMPLE_QUEUE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.38. EFFICIENCY_CAL_B_I_U

Table	GARDS_EFFICIENCY_CAL
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.39. EFFICIENCY_ERROR_CAL_B_I_U

Table	GARDS_EFFICIENCY_ERROR_CAL
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	
	:new.moddate := sysdate;
	
	end;

5.3.40. EFFICIENCY_ERR_CAL_B_I_U

Table	GARDS_EFFICIENCY_ERROR_CAL
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	
	:new.moddate := sysdate;
	
	end;

5.3.41. EFFICIENCY_IDC_PAIRS_B_I_U

Table	GARDS_EFFICIENCY_IDC_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.42. EFFICIENCY_IDC_PAIRS_I_U_D

Table	GARDS_EFFICIENCY_IDC_PAIRS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	`begin`
	`  IF INSERTING THEN`
	`  INSERT into rmsman.gards_efficiency_idc_pairs (`
	`  detector_id,`
	`  station_id,`
	`  effic_energy,`
	`  efficiency,`
	`  effic_error,`
	`  begin_date,`
	`  end_date,`
	`  sample_id`
	`) values (`
	`  :new.detector_id,`
	`  :new.station_id,`
	`  :new.effic_energy,`
	`  :new.efficiency,`
	`  :new.effic_error,`
	`  :new.begin_date,`
	`  :new.end_date,`
	`  :new.sample_id`
	`);`
	`ELSIF UPDATING THEN`
	`update rmsman.gards_efficiency_idc_pairs set`
	`  detector_id=:new.detector_id,`
	`  station_id=:new.station_id,`
	`  effic_energy=:new.effic_energy,`
	`  efficiency=:new.efficiency,`
	`  effic_error=:new.effic_error,`
	`  begin_date=:new.begin_date,`
	`  end_date=:new.end_date,`
	`  sample_id=:new.sample_id`
	`where detector_id=:old.detector_id and station_id=:old.sta-`
	`tion_id and begin_date=:old.begin_date`
	`  and effic_energy=:old.effic_energy;`
	`ELSE`
	`  delete from rmsman.gards_efficiency_idc_pairs where de-`

	tector_id=:old.detector_id
	  and station_id=:old.station_id and
	begin_date=:old.begin_date and effic_energy=:old.effic_energy;
	END IF;
	end;

5.3.43. EFFICIENCY_PAIRS_B_I_U

Table	GARDS_EFFICIENCY_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	 :new.moddate := sysdate;
	end;

5.3.44. EFF_VGSL_PAIRS_B_I_U

Table	GARDS_EFFICIENCY_VGSL_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	 :new.moddate := sysdate;
	end;

5.3.45. ENERGY_CAL_B_I_U

Table	GARDS_ENERGY_CAL
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	  :new.moddate := sysdate;
	end;

5.3.46. ENERGY_CAL_COV_B_I_U

Table	GARDS_ENERGY_CAL_COV
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	  :new.moddate := sysdate;
	end;

5.3.47. ENERGY_CAL_ORIG_B_I_U

Table	GARDS_ENERGY_CAL_ORIG
Type	BEFORE EACH ROW

Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.48. ENERGY_PAIRS_B_I_U

Table	GARDS_ENERGY_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.49. ENERGY_PAIRS_B_I_U

Table	GARDS_ENERGY_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`if user !='VPACI' then`
	`:new.moddate := sysdate;`
	`end if;`
	`end;`

5.3.50. ENERGY_PAIRS_ORIG_B_I_U

Table	GARDS_ENERGY_PAIRS_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.51. ENVIRONMENT_B_I_U

Table	GARDS_ENVIRONMENT
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.52. FLAGS_B_I_U

Table	GARDS_FLAGS

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.53. FPE_B_I_U

Table	GARDS_FPE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.54. GARDS_BG_EFFIC_PRS_I_U_D

Table	GARDS_BG_EFFICIENCY_PAIRS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	`begin`
	`  IF INSERTING THEN`
	`  insert into rmsman.gards_bg_efficiency_pairs`
	`  (sample_id, bg_efficiency, bg_effic_error, roi)`
	`  values`
	`(:new.sample_id, :new.bg_efficiency, :new.bg_effic_error, :new`
	`.roi);`
	`  ELSIF UPDATING THEN`
	`  update rmsman.gards_bg_efficiency_pairs set`
	`  sample_id = :new.sample_id,`
	`  bg_efficiency = :new.bg_efficiency ,`
	`  bg_effic_error = :new.bg_effic_error ,`
	`  roi = :new.roi`
	`  where sample_id= :old.sample_id and roi = :old.roi;`
	`  ELSE`
	`  delete from rmsman.gards_bg_efficiency_pairs where`
	`  sample_id = :old.sample_id and roi = :old.roi;`
	`  END IF;`
	`end;`

5.3.55. GARDS_BG_ENERGY_CAL_I_U_D

Table	GARDS_BG_ENERGY_CAL
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	`begin` `  IF INSERTING THEN`

```
                        insert into rmsman.gards_bg_energy_cal
                        (
                         sample_id,
                         beta_coeff1,
                         beta_coeff2,
                         beta_coeff3,
                         gamma_coeff1,
                         gamma_coeff2,
                         gamma_coeff3
                         )
                        values (
                        :new.sample_id,
                        :new.beta_coeff1,
                        :new.beta_coeff2,
                        :new.beta_coeff3,
                        :new.gamma_coeff1,
                        :new.gamma_coeff2,
                        :new.gamma_coeff3
                        );
                ELSIF UPDATING THEN
                        update rmsman.gards_bg_energy_cal set
                        sample_id = :new.sample_id,
                        beta_coeff1 = :new.beta_coeff1,
                        beta_coeff2 = :new.beta_coeff2,
                        beta_coeff3 = :new.beta_coeff3,
                        gamma_coeff1 = :new.gamma_coeff1,
                        gamma_coeff2 = :new.gamma_coeff2,
                        gamma_coeff3 = :new.gamma_coeff3
                     where sample_id= :old.sample_id;
                ELSE
                        delete from rmsman.gards_bg_energy_cal
where
                                sample_id = :old.sample_id;
                END IF;
end;
```

5.3.56. GARDS_BG_ENERGY_CAL_ORIG_I_U_D

Table	GARDS_BG_ENERGY_CAL_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	

```
begin
                IF INSERTING THEN
                        insert into
rmsman.gards_bg_energy_cal_orig
                        (
                         sample_id,
                         beta_coeff1,
                         beta_coeff2,
                         beta_coeff3,
                         gamma_coeff1,
```

```
                        gamma_coeff2,
                        gamma_coeff3
                        )
                      values (
                      :new.sample_id,
                      :new.beta_coeff1,
                      :new.beta_coeff2,
                      :new.beta_coeff3,
                      :new.gamma_coeff1,
                      :new.gamma_coeff2,
                      :new.gamma_coeff3
                      );
               ELSIF UPDATING THEN
                      update rmsman.gards_bg_energy_cal_orig
set
                      sample_id = :new.sample_id,
                      beta_coeff1 = :new.beta_coeff1,
                      beta_coeff2 = :new.beta_coeff2,
                      beta_coeff3 = :new.beta_coeff3,
                      gamma_coeff1 = :new.gamma_coeff1,
                      gamma_coeff2 = :new.gamma_coeff2,
                      gamma_coeff3 = :new.gamma_coeff3
                    where sample_id= :old.sample_id;
               ELSE
                      delete from
rmsman.gards_bg_energy_cal_orig where
                                  sample_id = :old.sample_id;
               END IF;
end;
```

5.3.57. GARDS_BG_ISOTOPE_CONCS_I_U_D

Table	GARDS_BG_ISOTOPE_CONCS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	

```
begin
        IF INSERTING THEN
        INSERT into rmsman.gards_bg_isotope_concs (
                SAMPLE_ID       ,
                NUCLIDE_ID      ,
                CONC            ,
                CONC_ERR        ,
                MDC             ,
            ACTIVITY_ACQ_START ,
            ACTIVITY_ERR_ACQ_START ,
            MDA_ACQ_START     ,
                NID_FLAG        ,
                LC              ,
                LD            ,
            METHOD_ID
                )
```

	VALUES (
	:new.SAMPLE_ID ,
	:new.NUCLIDE_ID ,
	:new.CONC ,
	:new.CONC_ERR ,
	:new.MDC ,
	:new.ACTIVITY_ACQ_START ,
	:new.ACTIVITY_ERR_ACQ_START ,
	:new.MDA_ACQ_START ,
	:new.NID_FLAG ,
	:new.LC ,
	:new.LD ,
	:new.METHOD_ID
	);
	ELSIF UPDATING THEN
	UPDATE rmsman.gards_bg_isotope_concs set
	SAMPLE_ID = :new.SAMPLE_ID ,
	NUCLIDE_ID = :new.NUCLIDE_ID ,
	CONC = :new.CONC ,
	CONC_ERR = :new.CONC_ERR ,
	MDC = :new.MDC ,
	ACTIVITY_ACQ_START = :new.ACTIVITY_ACQ_START ,
	ACTIVITY_ERR_ACQ_START
	= :new.ACTIVITY_ERR_ACQ_START ,
	MDA_ACQ_START = :new.MDA_ACQ_START ,
	NID_FLAG = :new.NID_FLAG ,
	LC = :new.LC ,
	LD = :new.LD ,
	METHOD_ID = :new.METHOD_ID
	where sample_id = :old.sample_id;
	ELSE
	delete from rmsman.gards_bg_isotope_concs
	where sample_id = :old.sample_id;
	END IF;
	end;

5.3.58. GARDS_BG_PROC_PARAMS_I_U_D

Table	GARDS_BG_PROC_PARAMS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	INSERT into rmsman.gards_bg_proc_params (
	SAMPLE_ID,
	LC_ABSCISSA,
	METHOD,
	DET_BKGND_USED,
	GAS_BKGND_USED,
	BETA_ECR_ORDER,
	GAMMA_ECR_ORDER,

```
                MAX_QC_DEV,
                QC_ID,
                XE_IN_AIR,
                DET_BKGND_ID,
                GAS_BKGND_ID,
                QC_B_THRESHOLD,
                BIN_ROWS,
                BIN_MIN_COUNT,
                BIN_GAMMA_START,
                BIN_BETA_START,
                BIN_MAX_VECTOR_SIZE,
                INTERFERENCE_USED
                )
        VALUES (
                :new.SAMPLE_ID,
                :new.LC_ABSCISSA,
                :new.METHOD,
                :new.DET_BKGND_USED,
                :new.GAS_BKGND_USED,
                :new.BETA_ECR_ORDER,
                :new.GAMMA_ECR_ORDER,
                :new.MAX_QC_DEV,
                :new.QC_ID,
                :new.XE_IN_AIR,
                :new.DET_BKGND_ID,
                :new.GAS_BKGND_ID,
                :new.QC_B_THRESHOLD,
                :new.BIN_ROWS,
                :new.BIN_MIN_COUNT,
                :new.BIN_GAMMA_START,
                :new.BIN_BETA_START,
                :new.BIN_MAX_VECTOR_SIZE,
                :new.INTERFERENCE_USED
                );
        ELSIF UPDATING THEN
        UPDATE rmsman.gards_bg_proc_params set
                SAMPLE_ID = :new.SAMPLE_ID,
                LC_ABSCISSA = :new.LC_ABSCISSA,
                METHOD = :new.METHOD,
                DET_BKGND_USED = :new.DET_BKGND_USED,
                GAS_BKGND_USED = :new.GAS_BKGND_USED,
                BETA_ECR_ORDER = :new.BETA_ECR_ORDER,
                GAMMA_ECR_ORDER = :new.GAMMA_ECR_ORDER,
                MAX_QC_DEV = :new.MAX_QC_DEV,
                QC_ID = :new.QC_ID,
                XE_IN_AIR = :new.XE_IN_AIR,
                DET_BKGND_ID = :new.DET_BKGND_ID,
                GAS_BKGND_ID = :new.GAS_BKGND_ID,
                QC_B_THRESHOLD = :new.QC_B_THRESHOLD,
                BIN_ROWS = :new.BIN_ROWS,
                BIN_MIN_COUNT = :new.BIN_MIN_COUNT,
                BIN_GAMMA_START = :new.BIN_GAMMA_START,
                BIN_BETA_START = :new.BIN_BETA_START,
```

	BIN_MAX_VECTOR_SIZE
	= :new.BIN_MAX_VECTOR_SIZE,
	INTERFERENCE_USED = :new.INTERFERENCE_USED
	where sample_id= :old.sample_id;
	ELSE
	delete from rmsman.gards_bg_proc_params where
	sample_id = :old.sample_id;
	END IF;
	end;

5.3.59. GARDS_BG_PROC_PARAMS_ROI_I_U_D

Table	GARDS_BG_PROC_PARAMS_ROI
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	INSERT into rmsman.gards_bg_proc_params_roi (
	SAMPLE_ID ,
	ROI ,
	HALFLIFE_SEC ,
	ABUNDANCE ,
	NUCLIDE_ID)
	VALUES (
	:new.SAMPLE_ID ,
	:new.ROI ,
	:new.HALFLIFE_SEC ,
	:new.ABUNDANCE ,
	:new.NUCLIDE_ID);
	ELSIF UPDATING THEN
	UPDATE rmsman.gards_bg_proc_params_roi set
	SAMPLE_ID = :new.SAMPLE_ID ,
	ROI = :new.ROI ,
	HALFLIFE_SEC = :new.HALFLIFE_SEC ,
	ABUNDANCE = :new.ABUNDANCE ,
	NUCLIDE_ID = :new.NUCLIDE_ID
	where sample_id= :old.sample_id;
	ELSE
	delete from rmsman.gards_bg_proc_params_roi where
	sample_id = :old.sample_id;
	END IF;
	end;

5.3.60. GARDS_BG_QC_RESULT_I_U_D

Table	GARDS_BG_QC_RESULT
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	begin IF INSERTING THEN

	INSERT into rmsman.gards_bg_qc_result (
	SAMPLE_ID ,
	AMPLITUDE ,
	FWHM ,
	CENTROID ,
	OFFSET ,
	SLOPE
	)
	VALUES (
	:new.SAMPLE_ID ,
	:new.AMPLITUDE ,
	:new.FWHM ,
	:new.CENTROID ,
	:new.OFFSET ,
	:new.SLOPE
	);
	ELSIF UPDATING THEN
	UPDATE rmsman.gards_bg_qc_result set
	SAMPLE_ID = :new.SAMPLE_ID ,
	AMPLITUDE = :new.AMPLITUDE ,
	FWHM = :new.FWHM ,
	CENTROID = :new.CENTROID ,
	OFFSET = :new.OFFSET ,
	SLOPE = :new.SLOPE
	where sample_id = :old.sample_id;
	ELSE
	delete from rmsman.gards_bg_qc_result where sample_id = :old.sample_id;
	END IF;
	end;

5.3.61. GARDS_BG_ROI_CONCS_I_U_D

Table	GARDS_BG_ROI_CONCS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	INSERT into rmsman.gards_bg_roi_concs (
	SAMPLE_ID ,
	ROI ,
	CONC ,
	CONC_ERR ,
	MDC ,
	NID_FLAG ,
	LC ,
	LD ,
	METHOD_ID
	)
	VALUES (
	:new.SAMPLE_ID ,

```
            :new.ROI    ,
            :new.CONC   ,
            :new.CONC_ERR     ,
            :new.MDC    ,
            :new.NID_FLAG     ,
            :new.LC           ,
            :new.LD   ,
    :new.METHOD_ID
            );
      ELSIF UPDATING THEN
      UPDATE rmsman.gards_bg_roi_concs set
            SAMPLE_ID = :new.SAMPLE_ID    ,
            ROI = :new.ROI                ,
            CONC = :new.CONC        ,
            CONC_ERR = :new.CONC_ERR      ,
            MDC = :new.MDC                ,
            NID_FLAG = :new.NID_FLAG      ,
            LC = :new.LC                  ,
            LD = :new.LD      ,
    METHOD_ID = :new.METHOD_ID
      where
    SAMPLE_ID = :old.SAMPLE_ID and
    ROI = :old.ROI and
    METHOD_ID = :old.METHOD_ID;
      ELSE
            delete from
      rmsman.gards_bg_roi_concs
    where
      SAMPLE_ID = :old.SAMPLE_ID and
      ROI = :old.ROI and
      METHOD_ID = :old.METHOD_ID;
      END IF;
end;
```

5.3.62. GARDS_BG_ROI_COUNTS_I_U_D

Table	GARDS_BG_ROI_COUNTS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	

```
begin
   if inserting then
      insert into RMSMAN.GARDS_BG_ROI_COUNTS
      (SAMPLE_ID,
       ROI,
       GROSS,
       GROSS_ERR,
       GAS_BKGND_GROSS,
       TEMP_COUNT,
       TEMP_COUNT_ERR,
       GAS_BKGND_COUNT,
       GAS_BKGND_COUNT_ERR,
```

```
        DET_BKGND_COUNT,
        DET_BKGND_COUNT_ERR,
        NET_COUNT,
        NET_COUNT_ERR,
        CRITICAL_LEV_SAMP,
        CRITICAL_LEV_GAS,
        MODDATE,
        METHOD_ID)
      values
      (:new.SAMPLE_ID,
      :new.ROI,
      :new.GROSS,
      :new.GROSS_ERR,
      :new.GAS_BKGND_GROSS,
      :new.TEMP_COUNT,
      :new.TEMP_COUNT_ERR,
      :new.GAS_BKGND_COUNT,
      :new.GAS_BKGND_COUNT_ERR,
      :new.DET_BKGND_COUNT,
      :new.DET_BKGND_COUNT_ERR,
          :new.NET_COUNT,
      :new.NET_COUNT_ERR,
      :new.CRITICAL_LEV_SAMP,
      :new.CRITICAL_LEV_GAS,
      :new.MODDATE,
      :new.METHOD_ID);
   elsif updating then
      update RMSMAN.GARDS_BG_ROI_COUNTS set
       GROSS = :new.GROSS,
       GROSS_ERR = :new.GROSS_ERR,
       GAS_BKGND_GROSS = :new.GAS_BKGND_GROSS,
       TEMP_COUNT = :new.TEMP_COUNT,
       TEMP_COUNT_ERR = :new.TEMP_COUNT_ERR,
       GAS_BKGND_COUNT = :new.GAS_BKGND_COUNT,
       GAS_BKGND_COUNT_ERR = :new.GAS_BKGND_COUNT_ERR,
       DET_BKGND_COUNT = :new.DET_BKGND_COUNT,
       DET_BKGND_COUNT_ERR = :new.DET_BKGND_COUNT_ERR,
       NET_COUNT = :new.NET_COUNT,
       NET_COUNT_ERR = :new.NET_COUNT_ERR,
       CRITICAL_LEV_SAMP = :new.CRITICAL_LEV_SAMP,
       CRITICAL_LEV_GAS = :new.CRITICAL_LEV_GAS,
       MODDATE = :new.MODDATE,
       METHOD_ID = :new.METHOD_ID
      where
       SAMPLE_ID = :old.SAMPLE_ID and
       ROI = :old.ROI and
       METHOD_ID = :old.METHOD_ID;
   else
      delete from
       RMSMAN.GARDS_BG_ROI_COUNTS
          where
       SAMPLE_ID = :old.SAMPLE_ID and
       ROI = :old.ROI and
```

	METHOD_ID = :old.METHOD_ID;
	end if;
	end;

5.3.63. GARDS_B_ENRGY_PRS_INSERT

Table	GARDS_B_ENERGY_PAIRS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	if inserting then
	insert into rmsman.gards_b_energy_pairs (sample_id, decay_mode, cal_energy, cal_error, channel, type, winner)
	values (:new.sample_id, :new.decay_mode, :new.cal_energy, :new.cal_error, :new.channel, :new.type, :new.winner);
	elsif updating then
	update rmsman.gards_b_energy_pairs set
	sample_id = :new.sample_id,
	decay_mode = :new.decay_mode ,
	cal_energy = :new.cal_energy ,
	cal_error = :new.cal_error ,
	channel = :new.channel ,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and nvl(type,'NULL') = nvl(:old.type,'NULL') and cal_energy = :old.cal_energy;
	else
	delete from rmsman.gards_b_energy_pairs where sample_id = :old.sample_id and cal_energy = :old.cal_energy and nvl(type,'NULL') = nvl(:old.type,'NULL');
	end if;
	end;

5.3.64. GARDS_B_ENRGY_PRS_ORG_I_U_D

Table	GARDS_B_ENERGY_PAIRS_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	if inserting then
	insert into rmsman.gards_b_energy_pairs_orig
	(sample_id, decay_mode, cal_energy, cal_error, channel, type, winner)
	values (:new.sample_id, :new.decay_mode, :new.cal_energy, :new.cal_error, :new.channel, :new.type, :new.winner);
	elsif updating then
	update rmsman.gards_b_energy_pairs_orig set
	sample_id = :new.sample_id,
	decay_mode = :new.decay_mode ,
	cal_energy = :new.cal_energy ,

	cal_error = :new.cal_error ,
	channel = :new.channel,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and cal_energy
	= :old.cal_energy and nvl(type,'NULL') =
	nvl(:old.type,'NULL');
	else
	delete from rmsman.gards_b_energy_pairs_orig where
	sample_id = :old.sample_id and cal_energy
	= :old.cal_energy and nvl(type,'NULL') =
	nvl(:old.type,'NULL');
	end if;
	end;

5.3.65. GARDS_B_RESOLUTION_CAL_B_I_U

Table	GARDS_B_RESOLUTION_CAL
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.66. GARDS_B_RESOLUTION_CAL_I_U_D

Table	GARDS_B_RESOLUTION_CAL
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into RMSMAN.GARDS_B_RESOLUTION_CAL (
	sample_id,
	coeff1,
	coeff2,
	coeff3,
	coeff4,
	coeff5,
	coeff6,
	coeff7,
	coeff8,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.coeff1,
	:new.coeff2,
	:new.coeff3,
	:new.coeff4,

	:new.coeff5,
	:new.coeff6,
	:new.coeff7,
	:new.coeff8,
	:new.type,
	:new.winner
	);
	ELSIF UPDATING THEN
	update RMSMAN.GARDS_B_RESOLUTION_CAL set
	sample_id = :new.sample_id,
	coeff1 = :new.coeff1,
	coeff2 = :new.coeff2,
	coeff3 = :new.coeff3,
	coeff4 = :new.coeff4,
	coeff5 = :new.coeff5,
	coeff6 = :new.coeff6,
	coeff7 = :new.coeff7,
	coeff8 = :new.coeff8,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and NVL(type,'NULL') =
	NVL(:old.type,'NULL');
	ELSE
	delete from RMSMAN.GARDS_B_RESOLUTION_CAL where
	sample_id = :old.sample_id and NVL(type,'NULL') = NVL(:old.-
	type,'NULL');
	END IF;
	end;

5.3.67. GARDS_B_RES_PAIRS_I_U_D

Table	GARDS_B_RESOLUTION_PAIRS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	if inserting then
	insert into rmsman.gards_b_resolution_pairs (sample_id,
	res_energy, resolution, res_error, type, winner)
	values (:new.sample_id, :new.res_energy, :new.resolu-
	tion, :new.res_error, :new.type, :new.winner);
	elsif updating then
	update rmsman.gards_b_resolution_pairs set sample_id
	= :new.sample_id, res_energy = :new.res_energy, resolution
	= :new.resolution,
	res_error = :new.res_error, type = :new.type, winner =
	:new.winner
	where sample_id= :old.sample_id and nvl(type,'NULL') =
	nvl(:old.type,'NULL') and res_energy = :old.res_energy;
	else
	delete from rmsman.gards_b_resolution_pairs where
	sample_id = :old.sample_id and res_energy = :old.res_energy

	and nvl(type,'NULL') = nvl(:old.type,'NULL');
	end if;
	end;

5.3.68. GARDS_B_RES_PRS_ORIG_I_U_D

Table	GARDS_B_RESOLUTION_PAIRS_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	if inserting then
	insert into rmsman.gards_b_resolution_pairs_orig (sample_id, res_energy, resolution, res_error, type, winner)
	values (:new.sample_id, :new.res_energy, :new.resolution, :new.res_error, :new.type, :new.winner);
	elsif updating then
	update rmsman.gards_b_resolution_pairs_orig set sample_id = :new.sample_id, res_energy = :new.res_energy, resolution = :new.resolution,
	res_error = :new.res_error, type = :new.type, winner = :new.winner
	where sample_id= :old.sample_id and nvl(type,'NULL') = nvl(:old.type,'NULL') and res_energy = :old.res_energy;
	else
	delete from rmsman.gards_b_resolution_pairs_orig where sample_id = :old.sample_id and res_energy = :old.res_energy and nvl(type,'NULL') = nvl(:old.type,'NULL');
	end if;
	end;

5.3.69. GARDS_COMMENTS_I_U_D

Table	GARDS_COMMENTS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_comments (
	sample_id,
	peak_id,
	nucl_name,
	comment_id,
	analyst,
	dtg,
	comment_type
	)
	values (
	:new.sample_id,
	:new.peak_id,
	:new.nucl_name,

```
              :new.comment_id,
              :new.analyst,
              :new.dtg,
              :new.comment_type
            );
      ELSIF UPDATING THEN
            update rmsman.gards_comments set
              sample_id = :new.sample_id,
              peak_id = :new.peak_id,
              nucl_name = :new.nucl_name,
              comment_id = :new.comment_id,
              analyst = :new.analyst,
              dtg = :new.dtg,
              comment_type = :new.comment_type
            where comment_id= :old.comment_id;
      ELSE
            delete from rmsman.gards_comments where comment_id
= :old.comment_id;
      END IF;
end;
```

5.3.70. GARDS_EFFICIENCY_CAL_I_U_D

Table	GARDS_EFFICIENCY_CAL
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	

```
begin
      IF INSERTING THEN
            insert into rmsman.gards_efficiency_cal (
              sample_id,
              degree,
              efftype,
              coeff1,
              coeff2,
              coeff3,
              coeff4,
              coeff5,
              coeff6,
              coeff7,
              coeff8
            )
            values (
              :new.sample_id,
              :new.degree,
              :new.efftype,
              :new.coeff1,
              :new.coeff2,
              :new.coeff3,
              :new.coeff4,
              :new.coeff5,
              :new.coeff6,
```

	:new.coeff7,
	:new.coeff8
	);
	ELSIF UPDATING THEN
	update rmsman.gards_efficiency_cal set
	sample_id = :new.sample_id,
	degree = :new.degree,
	efftype = :new.efftype,
	coeff1 = :new.coeff1,
	coeff2 = :new.coeff2,
	coeff3 = :new.coeff3,
	coeff4 = :new.coeff4,
	coeff5 = :new.coeff5,
	coeff6 = :new.coeff6,
	coeff7 = :new.coeff7,
	coeff8 = :new.coeff8
	where sample_id= :old.sample_id;
	ELSE
	delete from rmsman.gards_efficiency_cal where sample_id = :old.sample_id;
	END IF;
	end;

5.3.71. GARDS_EFFICIENCY_ERR_CAL_I_U_D

Table	GARDS_EFFICIENCY_ERROR_CAL
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	
	IF INSERTING THEN
	
	insert into rmsman.gards_efficiency_error_cal (
	
	sample_id,
	
	degree,
	
	type,
	
	coeff1,
	
	coeff2,
	
	coeff3,
	
	coeff4,
	
	coeff5,

```
                coeff6,
                coeff7,
                coeff8
              )
              values (
                :new.sample_id,
                :new.degree,
                :new.type,
                :new.coeff1,
                :new.coeff2,
                :new.coeff3,
                :new.coeff4,
                :new.coeff5,
                :new.coeff6,
                :new.coeff7,
                :new.coeff8
              );
        ELSIF UPDATING THEN
              update rmsman.gards_efficiency_error_cal set
                sample_id = :new.sample_id,
                degree = :new.degree,
                type = :new.type,
                coeff1 = :new.coeff1,
                coeff2 = :new.coeff2,
                coeff3 = :new.coeff3,
                coeff4 = :new.coeff4,
                coeff5 = :new.coeff5,
```

	coeff6 = :new.coeff6,
	
	coeff7 = :new.coeff7,
	
	coeff8 = :new.coeff8
	
	where sample_id= :old.sample_id;
	
	ELSE
	
	delete from rmsman.gards_efficiency_error_cal where sample_id = :old.sample_id;
	
	END IF;
	
	end;

5.3.72. GARDS_EFFICIENCY_PAIRS_I_U_D

Table	GARDS_EFFICIENCY_PAIRS
Type	AFTER EACH ROW
Event	INSERT

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_efficiency_pairs (
	sample_id,
	effic_energy,
	efficiency,
	effic_error
	)
	values (
	:new.sample_id,
	:new.effic_energy,
	:new.efficiency,
	:new.effic_error
	);
	ELSIF UPDATING THEN
	update rmsman.gards_efficiency_pairs set
	sample_id = :new.sample_id,
	effic_energy = :new.effic_energy,
	efficiency = :new.efficiency ,
	effic_error = :new.effic_error
	where sample_id= :old.sample_id and effic_energy = :old.effic_energy;
	ELSE
	delete from rmsman.gards_efficiency_pairs where sample_id = :old.sample_id and effic_energy = :old.effic_energy ;
	END IF;
	end;

5.3.73. GARDS_ENERGY_CAL_COV_I_U_D

Table	GARDS_ENERGY_CAL_COV
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_energy_cal_cov (
	sample_id,
	row_index,
	col_index,
	coeff,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.row_index,
	:new.col_index,
	:new.coeff,
	:new.type,
	:new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_energy_cal_cov set
	sample_id = :new.sample_id,
	row_index = :new.row_index,
	col_index = :new.col_index,
	coeff = :new.coeff,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and NVL(type,'NULL') = NVL(:old.type,'NULL') and row_index= :old.row_index and col_index= :old.col_index;
	ELSE
	delete from rmsman.gards_energy_cal_cov where sample_id = :old.sample_id and NVL(type,'NULL') = NVL(:old.-type,'NULL')
	and row_index= :old.row_index and col_index= :old.col_index;
	END IF;
	end;

5.3.74. GARDS_ENERGY_CAL_I_U_D

Table	GARDS_ENERGY_CAL
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	begin IF INSERTING THEN

```
                insert into rmsman.gards_energy_cal (
                  sample_id,
                  coeff1,
                  coeff2,
                  coeff3,
                  coeff4,
                  coeff5,
                  coeff6,
                  coeff7,
                  coeff8,
                  energy_units,
                  cnv_factor,
                  ape,
                  det,
                  mse,
                  tstat,
                  score,
                  type,
                  winner
                )
                values (
                  :new.sample_id,
                  :new.coeff1,
                  :new.coeff2,
                  :new.coeff3,
                  :new.coeff4,
                  :new.coeff5,
                  :new.coeff6,
                  :new.coeff7,
                  :new.coeff8,
                  :new.energy_units,
                  :new.cnv_factor,
                  :new.ape,
                  :new.det,
                  :new.mse,
                  :new.tstat,
                  :new.score,
                  :new.type,
                  :new.winner
                );
        ELSIF UPDATING THEN
                update rmsman.gards_energy_cal set
                  sample_id = :new.sample_id,
                  coeff1 = :new.coeff1,
                  coeff2 = :new.coeff2,
                  coeff3 = :new.coeff3,
                  coeff4 = :new.coeff4,
                  coeff5 = :new.coeff5,
                  coeff6 = :new.coeff6,
                  coeff7 = :new.coeff7,
                  coeff8 = :new.coeff8,
                  energy_units = :new.energy_units,
                  cnv_factor = :new.cnv_factor,
```

	ape = :new.ape,
	det = :new.det,
	mse = :new.mse,
	tstat = :new.tstat,
	score = :new.score,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and NVL(type,'NULL') = NVL(:old.type,'NULL');
	ELSE
	delete from rmsman.gards_energy_cal where sample_id = :old.sample_id and NVL(type,'NULL') = NVL(:old.-type,'NULL');
	END IF;
	end;

5.3.75. GARDS_ENERGY_CAL_ORIG_I_U_D

Table	GARDS_ENERGY_CAL_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_energy_cal_orig (
	sample_id,
	coeff1,
	coeff2,
	coeff3,
	coeff4,
	coeff5,
	coeff6,
	coeff7,
	coeff8,
	energy_units,
	cnv_factor,
	ape,
	det,
	mse,
	tstat,
	score,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.coeff1,
	:new.coeff2,
	:new.coeff3,
	:new.coeff4,
	:new.coeff5,
	:new.coeff6,

```
  :new.coeff7,
  :new.coeff8,
  :new.energy_units,
  :new.cnv_factor,
  :new.ape,
  :new.det,
  :new.mse,
  :new.tstat,
  :new.score,
                  :new.type,
                  :new.winner
);
ELSIF UPDATING THEN
update rmsman.gards_energy_cal_orig set
                  sample_id = :new.sample_id,
                  coeff1 = :new.coeff1,
                  coeff2 = :new.coeff2,
                  coeff3 = :new.coeff3,
                  coeff4 = :new.coeff4,
                  coeff5 = :new.coeff5,
                  coeff6 = :new.coeff6,
                  coeff7 = :new.coeff7,
                  coeff8 = :new.coeff8,
                  energy_units = :new.energy_units,
                  cnv_factor = :new.cnv_factor,
                  ape = :new.ape,
                  det = :new.det,
                  mse = :new.mse,
                  tstat = :new.tstat,
                  score = :new.score,
                  type = :new.type,
                  winner = :new.winner
where sample_id= :old.sample_id and NVL(type,'NULL') =
NVL(:old.type,'NULL');
ELSE
                delete from rmsman.gards_energy_cal_orig where
sample_id = :old.sample_id and NVL(type,'NULL') = NVL(:old.-
type,'NULL');
END IF;
end;
```

5.3.76. GARDS_ENERGY_PAIRS_I_U_D

Table	GARDS_ENERGY_PAIRS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	`begin`
	`IF INSERTING THEN`
	`insert into rmsman.gards_energy_pairs (`
	`  sample_id,`
	`  cal_energy,`

	channel,
	cal_error,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.cal_energy,
	:new.channel,
	:new.cal_error,
	:new.type,
	:new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_energy_pairs set
	sample_id = :new.sample_id,
	cal_energy = :new.cal_energy,
	channel = :new.channel,
	cal_error = :new.cal_error,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and
	NVL(type,'NULL') = NVL(:old.type,'NULL') and cal_energy
	= :old.cal_energy;
	ELSE
	delete from rmsman.gards_energy_pairs where
	sample_id = :old.sample_id and
	cal_energy = :old.cal_energy and
	NVL(type,'NULL') = NVL(:old.type,'NULL');
	END IF;
	end;

5.3.77. GARDS_ENERGY_PAIRS_ORIG_I_U_D

Table	GARDS_ENERGY_PAIRS_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_energy_pairs_orig (
	sample_id,
	cal_energy,
	channel,
	cal_error,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.cal_energy,
	:new.channel,

	:new.cal_error,
	:new.type,
	:new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_energy_pairs_orig set
	sample_id = :new.sample_id,
	cal_energy = :new.cal_energy,
	channel = :new.channel,
	cal_error = :new.cal_error,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and cal_energy = :old.cal_energy and NVL(type,'NULL') = NVL(:old.type,'NULL');
	ELSE
	delete from rmsman.gards_energy_pairs_orig where sample_id = :old.sample_id and cal_energy = :old.cal_energy and NVL(type,'NULL') = NVL(:old.type,'NULL');
	END IF;
	end;

5.3.78. GARDS_NCL_LNS_IDD_ORG_I_U_D

Table	GARDS_NUCL_LINES_IDED_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_nucl_lines_ided_orig (
	sample_id,
	station_id,
	detector_id,
	name, energy,
	energy_err,
	abundance,
	abundance_err,
	peak,
	activity,
	activ_err,
	effic,
	effic_err,
	mda,
	key_flag,
	nuclide_id,
	csc_ratio,
	csc_ratio_err,
	csc_mod_flag,
	id_percent
	)
	values (

```
    :new.sample_id,
    :new.station_id,
    :new.detector_id,
    :new.name,
    :new.energy,
    :new.energy_err,
    :new.abundance,
    :new.abundance_err,
    :new.peak,
    :new.activity,
    :new.activ_err,
    :new.effic,
    :new.effic_err,
    :new.mda,
    :new.key_flag,
    :new.nuclide_id,
                  :new.csc_ratio,
                  :new.csc_ratio_err,
                  :new.csc_mod_flag,
                  :new.id_percent
  );
 ELSIF UPDATING THEN
  update rmsman.gards_nucl_lines_ided_orig set
     sample_id = :new.sample_id,
     station_id = :new.station_id,
     detector_id = :new.detector_id,
     name = :new.name,
     energy = :new.energy,
     energy_err = :new.energy_err,
     abundance = :new.abundance,
     abundance_err = :new.abundance_err,
     peak = :new.peak,
     activity = :new.activity,
     activ_err = :new.activ_err,
     effic = :new.effic,
     effic_err = :new.effic_err,
     mda = :new.mda,
     key_flag = :new.key_flag,
     nuclide_id = :new.nuclide_id,
                  csc_ratio = :new.csc_ratio,
                  csc_ratio_err = :new.csc_ratio_err,
                  csc_mod_flag = :new.csc_mod_flag,
                  id_percent = :new.id_percent
  where sample_id= :old.sample_id and name = :old.name and en-
ergy = :old.energy;
 ELSE
  delete from rmsman.gards_nucl_lines_ided_orig where
sample_id = :old.sample_id
           and name = :old.name and energy = :old.energy;
 END IF;
end;
```

5.3.79. GARDS_NUCL_IDED_I_U_D

Table	GARDS_NUCL_IDED
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_nucl_ided (
	sample_id,
	station_id,
	detector_id,
	nuclide_id,
	name,
	type,
	halflife,
	ave_activ,
	ave_activ_err,
	activ_key,
	activ_key_err,
	mda,
	mda_err,
	ave_activ_acq_start,
	ave_activ_err_acq_start,
	activ_key_acq_start,
	activ_key_err_acq_start,
	mda_acq_start,
	mda_err_acq_start,
	nid_flag,
	activ_decay,
	activ_decay_err,
	comp_confid,
	report_mda,
	pd_mod_flag,
	csc_ratio,
	csc_ratio_err,
	csc_mod_flag
	)
	values (
	:new.sample_id,
	:new.station_id,
	:new.detector_id,
	:new.nuclide_id,
	:new.name, :new.type,
	:new.halflife ,
	:new.ave_activ,
	:new.ave_activ_err,
	:new.activ_key,
	:new.activ_key_err,
	:new.mda,
	:new.mda_err,
	:new.ave_activ_acq_start,
	:new.ave_activ_err_acq_start,

```
                    :new.activ_key_acq_start,
                    :new.activ_key_err_acq_start,
                    :new.mda_acq_start,
                    :new.mda_err_acq_start,
                    :new.nid_flag,
                    :new.activ_decay,
                    :new.activ_decay_err,
                    :new.comp_confid,
                    :new.report_mda,
                    :new.pd_mod_flag,
                    :new.csc_ratio,
                    :new.csc_ratio_err,
                    :new.csc_mod_flag
                  );
        ELSIF UPDATING THEN
                  update rmsman.gards_nucl_ided set
                    sample_id = :new.sample_id,
                    station_id = :new.station_id,
                    detector_id = :new.detector_id,
                    nuclide_id = :new.nuclide_id,
                    name = :new.name,
                    type = :new.type,
                    halflife = :new.halflife ,
                    ave_activ = :new.ave_activ,
                    ave_activ_err = :new.ave_activ_err ,
                    activ_key = :new.activ_key,
                    activ_key_err = :new.activ_key_err,
                    mda = :new.mda,
                    mda_err = :new.mda_err,
                    ave_activ_acq_start
= :new.ave_activ_acq_start,
                    ave_activ_err_acq_start
= :new.ave_activ_err_acq_start ,
                    activ_key_acq_start
= :new.activ_key_acq_start,
                    activ_key_err_acq_start
= :new.activ_key_err_acq_start,
                    mda_acq_start = :new.mda_acq_start,
                    mda_err_acq_start = :new.mda_err_acq_start,
                    nid_flag = :new.nid_flag,
                    activ_decay = :new.activ_decay,
                    activ_decay_err = :new.activ_decay_err,
                    comp_confid = :new.comp_confid,
                    report_mda = :new.report_mda,
                    pd_mod_flag = :new.pd_mod_flag,
                    csc_ratio = :new.csc_ratio,
                    csc_ratio_err = :new.csc_ratio_err,
                    csc_mod_flag = :new.csc_mod_flag
                  where sample_id= :old.sample_id and name
= :old.name;
        ELSE
                  delete from rmsman.gards_nucl_ided where
sample_id = :old.sample_id and name = :old.name;
```

	END IF; end;

5.3.80. GARDS_NUCL_IDED_ORIG_I_U_D

Table	GARDS_NUCL_IDED_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_nucl_ided_orig (
	sample_id,
	station_id,
	detector_id,
	nuclide_id,
	name,
	type,
	halflife,
	ave_activ,
	ave_activ_err,
	activ_key,
	activ_key_err,
	mda,
	mda_err,
	ave_activ_acq_start,
	ave_activ_err_acq_start,
	activ_key_acq_start,
	activ_key_err_acq_start,
	mda_acq_start,
	mda_err_acq_start,
	nid_flag,
	activ_decay,
	activ_decay_err,
	comp_confid,
	report_mda,
	pd_mod_flag,
	csc_ratio,
	csc_ratio_err,
	csc_mod_flag
	)
	values (
	:new.sample_id,
	:new.station_id,
	:new.detector_id,
	:new.nuclide_id,
	:new.name,
	:new.type,
	:new.halflife,
	:new.ave_activ,
	:new.ave_activ_err,
	:new.activ_key,

```
                    :new.activ_key_err,
                    :new.mda,
                    :new.mda_err,
                    :new.ave_activ_acq_start,
                    :new.ave_activ_err_acq_start,
                    :new.activ_key_acq_start,
                    :new.activ_key_err_acq_start,
                    :new.mda_acq_start,
                    :new.mda_err_acq_start,
                    :new.nid_flag,
                    :new.activ_decay,
                    :new.activ_decay_err,
                    :new.comp_confid,
                    :new.report_mda,
                    :new.pd_mod_flag,
                    :new.csc_ratio,
                    :new.csc_ratio_err,
                    :new.csc_mod_flag
                  );
          ELSIF UPDATING THEN
                  update rmsman.gards_nucl_ided_orig set
                    sample_id = :new.sample_id,
                    station_id = :new.station_id ,
                    detector_id = :new.detector_id,
                    nuclide_id = :new.nuclide_id,
                    name = :new.name,
                    type = :new.type,
                    halflife = :new.halflife,
                    ave_activ = :new.ave_activ,
                    ave_activ_err = :new.ave_activ_err,
                    activ_key = :new.activ_key,
                    activ_key_err = :new.activ_key_err,
                    mda = :new.mda,
                    mda_err = :new.mda_err,
                    ave_activ_acq_start
= :new.ave_activ_acq_start,
                    ave_activ_err_acq_start
= :new.ave_activ_err_acq_start,
                    activ_key_acq_start
= :new.activ_key_acq_start,
                    activ_key_err_acq_start
= :new.activ_key_err_acq_start,
                    mda_acq_start = :new.mda_acq_start,
                    mda_err_acq_start = :new.mda_err_acq_start,
                    nid_flag = :new.nid_flag,
                    activ_decay = :new.activ_decay,
                    activ_decay_err = :new.activ_decay_err,
                    comp_confid = :new.comp_confid ,
                    report_mda = :new.report_mda,
                    pd_mod_flag = :new.pd_mod_flag,
                    csc_ratio = :new.csc_ratio,
                    csc_ratio_err = :new.csc_ratio_err,
                    csc_mod_flag = :new.csc_mod_flag
```

	where sample_id= :old.sample_id and name = :old.name;
	ELSE
	delete from rmsman.gards_nucl_ided_orig where sample_id = :old.sample_id and name = :old.name;
	END IF;
	
	end;

5.3.81. GARDS_NUCL_LINES_IDED_I_U_D

Table	GARDS_NUCL_LINES_IDED
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_nucl_lines_ided (
	sample_id,
	station_id,
	detector_id,
	name,
	energy,
	energy_err,
	abundance,
	abundance_err,
	peak,
	activity,
	activ_err,
	effic,
	effic_err,
	mda,
	key_flag,
	nuclide_id,
	csc_ratio,
	csc_ratio_err,
	csc_mod_flag,
	id_percent
	)
	values (
	:new.sample_id,
	:new.station_id,
	:new.detector_id,
	:new.name,
	:new.energy,
	:new.energy_err,
	:new.abundance,
	:new.abundance_err,
	:new.peak,
	:new.activity,
	:new.activ_err,
	:new.effic,

	:new.effic_err,
	:new.mda,
	:new.key_flag,
	:new.nuclide_id,
	:new.csc_ratio,
	:new.csc_ratio_err,
	:new.csc_mod_flag,
	:new.id_percent
	);
	ELSIF UPDATING THEN
	update rmsman.gards_nucl_lines_ided set
	sample_id = :new.sample_id,
	station_id = :new.station_id,
	detector_id = :new.detector_id,
	name = :new.name,
	energy = :new.energy,
	energy_err = :new.energy_err,
	abundance = :new.abundance,
	abundance_err = :new.abundance_err,
	peak = :new.peak,
	activity = :new.activity,
	activ_err = :new.activ_err,
	effic = :new.effic ,
	effic_err = :new.effic_err,
	mda = :new.mda,
	key_flag = :new.key_flag,
	nuclide_id = :new.nuclide_id,
	csc_ratio = :new.csc_ratio,
	csc_ratio_err = :new.csc_ratio_err,
	csc_mod_flag = :new.csc_mod_flag,
	id_percent = :new.id_percent
	where sample_id= :old.sample_id and name = :old.name and energy = :old.energy;
	ELSE
	delete from rmsman.gards_nucl_lines_ided where sample_id = :old.sample_id
	and name = :old.name and energy = :old.energy;
	END IF;
	end;

5.3.82. GARDS_PEAKS_I_U_D

Table	GARDS_PEAKS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_peaks (
	sample_id,
	peak_id,
	centroid,

```
    centroid_err,
    energy, energy_err,
    left_chan,
    width,
     back_count,
    back_uncer,
    fwhm,
    fwhm_err,
    area,
    area_err,
    original_area,
     original_uncer,
    counts_sec,
    counts_sec_err,
    efficiency,
    eff_error,
     back_channel,
    ided,
    fitted,
    multiplet,
    peak_sig,
    lc,
    pss,
    peak_tol,
    detectability,
    bkgnd_area,
    bkgnd_area_err,
    net_area,
    net_area_err
  )
  values (
    :new.sample_id,
    :new.peak_id,
    :new.centroid,
    :new.centroid_err,
    :new.energy,
    :new.energy_err,
    :new.left_chan,
    :new.width,
    :new.back_count,
    :new.back_uncer,
    :new.fwhm,
    :new.fwhm_err,
    :new.area,
    :new.area_err,
    :new.original_area,
    :new.original_uncer,
    :new.counts_sec,
    :new.counts_sec_err,
    :new.efficiency,
    :new.eff_error,
    :new.back_channel,
    :new.ided,
```

```
    :new.fitted,
    :new.multiplet,
    :new.peak_sig,
    :new.lc,
    :new.pss,
    :new.peak_tol,
    :new.detectability,
    :new.bkgnd_area,
    :new.bkgnd_area_err,
    :new.net_area,
    :new.net_area_err
 );
 ELSIF UPDATING THEN
  update rmsman.gards_peaks set
     sample_id = :new.sample_id ,
     peak_id = :new.peak_id,
     centroid = :new.centroid,
     centroid_err = :new.centroid_err,
     energy = :new.energy,
     energy_err = :new.energy_err,
     left_chan = :new.left_chan,
     width = :new.width,
     back_count = :new.back_count,
     back_uncer = :new.back_uncer,
     fwhm = :new.fwhm ,
     fwhm_err = :new.fwhm_err,
     area = :new.area,
     area_err = :new.area_err,
     original_area = :new.original_area,
     original_uncer = :new.original_uncer,
     counts_sec = :new.counts_sec ,
     counts_sec_err = :new.counts_sec_err ,
     efficiency = :new.efficiency,
     eff_error = :new.eff_error ,
     back_channel = :new.back_channel,
     ided = :new.ided,
     fitted = :new.fitted,
     multiplet = :new.multiplet,
     peak_sig = :new.peak_sig,
     lc = :new.lc,
     pss = :new.pss,
     peak_tol = :new.peak_tol,
     detectability = :new.detectability,
     bkgnd_area = :new.bkgnd_area,
     bkgnd_area_err = :new.bkgnd_area_err,
     net_area = :new.net_area,
     net_area_err = :new.net_area_err
  where sample_id= :old.sample_id and peak_id = :old.peak_id;
 ELSE
  delete from rmsman.gards_peaks where sample_id
= :old.sample_id and peak_id = :old.peak_id;
 END IF;
end;
```

5.3.83. GARDS_PEAKS_ORIG_I_U_D

Table	GARDS_PEAKS_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_peaks_orig (
	sample_id,
	peak_id,
	centroid,
	centroid_err,
	energy,
	energy_err,
	left_chan,
	width,
	back_count,
	back_uncer,
	fwhm,
	fwhm_err,
	area,
	area_err,
	original_area,
	original_uncer,
	counts_sec,
	counts_sec_err,
	efficiency,
	eff_error,
	back_channel,
	ided,
	fitted,
	multiplet,
	peak_sig,
	lc,
	pss,
	peak_tol,
	detectability
	)
	values (
	:new.sample_id,
	:new.peak_id,
	:new.centroid,
	:new.centroid_err,
	:new.energy,
	:new.energy_err,
	:new.left_chan,
	:new.width,
	:new.back_count,
	:new.back_uncer,
	:new.fwhm,
	:new.fwhm_err,
	:new.area,

```
    :new.area_err,
     :new.original_area,
    :new.original_uncer,
    :new.counts_sec,
    :new.counts_sec_err,
     :new.efficiency,
    :new.eff_error,
    :new.back_channel,
    :new.ided,
    :new.fitted,
     :new.multiplet,
    :new.peak_sig,
    :new.lc,
    :new.pss,
    :new.peak_tol,
          :new.detectability
  );
 ELSIF UPDATING THEN
  update rmsman.gards_peaks_orig set
     sample_id = :new.sample_id,
     peak_id = :new.peak_id,
     centroid = :new.centroid,
     centroid_err = :new.centroid_err,
     energy = :new.energy,
     energy_err = :new.energy_err ,
     left_chan = :new.left_chan,
     width = :new.width,
     back_count = :new.back_count ,
     back_uncer = :new.back_uncer,
     fwhm = :new.fwhm ,
     fwhm_err = :new.fwhm_err,
     area = :new.area,
     area_err = :new.area_err,
     original_area = :new.original_area,
     original_uncer = :new.original_uncer,
     counts_sec = :new.counts_sec,
     counts_sec_err = :new.counts_sec_err,
     efficiency = :new.efficiency,
     eff_error = :new.eff_error,
     back_channel = :new.back_channel,
     ided = :new.ided,
     fitted = :new.fitted,
     multiplet = :new.multiplet,
     peak_sig = :new.peak_sig,
     lc = :new.lc,
     pss = :new.pss,
     peak_tol = :new.peak_tol,
          detectability = :new.detectability
  where sample_id= :old.sample_id and peak_id = :old.peak_id;
 ELSE
  delete from rmsman.gards_peaks_orig where sample_id
= :old.sample_id and peak_id = :old.peak_id;
 END IF;
```

	end;

5.3.84. GARDS_PRODUCT_I_U_D

Table	GARDS_PRODUCT
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_product
	(sample_id,
	foff,
	dsize,
	dir,
	dfile,
	author,
	revision,
	typeid)
	VALUES (
	:new.sample_id,
	:new.foff,
	:new.dsize,
	:new.dir,
	:new.dfile,
	:new.author,
	:new.revision,
	:new.typeid);
	ELSIF UPDATING THEN
	update rmsman.gards_product set
	sample_id = :new.sample_id,
	foff = :new.foff,
	dsize = :new.dsize,
	dir = :new.dir,
	dfile = :new.dfile,
	author = :new.author,
	revision = :new.revision,
	typeid = :new.typeid
	where sample_id= :old.sample_id and dfile
	= :old.dfile and typeid = :old.typeid;
	ELSE
	delete from rmsman.gards_product where
	sample_id = :old.sample_id and dfile
	= :old.dfile and typeid = :old.typeid;
	END IF;
	end;

5.3.85. GARDS_QCHISTORY_I_U_D

Table	GARDS_QCHISTORY
Type	AFTER EACH ROW

Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_qchistory (
	station_id,
	detector_id,
	sample_id,
	state,
	lddate
	)
	values (
	:new.station_id,
	:new.detector_id,
	:new.sample_id,
	:new.state,
	:new.lddate
	);
	ELSIF UPDATING THEN
	update rmsman.gards_qchistory set
	station_id = :new.station_id,
	detector_id = :new.detector_id,
	sample_id = :new.sample_id ,
	state = :new.state,
	lddate = :new.lddate
	where sample_id= :old.sample_id and state
	= :old.state;
	ELSE
	delete from rmsman.gards_qchistory where sample_id
	= :old.sample_id
	and state = :old.state;
	END IF;
	end;

5.3.86. GARDS_QC_RESULTS_I_U_D

Table	GARDS_QC_RESULTS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_qc_results (
	sample_id,
	test_name,
	flag,
	qc_comment
	)
	values (
	:new.sample_id,
	:new.test_name,
	:new.flag,
	:new.qc_comment

	);
	ELSIF UPDATING THEN
	update rmsman.gards_qc_results set
	sample_id = :new.sample_id,
	test_name = :new.test_name,
	flag = :new.flag,
	qc_comment = :new.qc_comment
	where sample_id = :old.sample_id;
	ELSE
	delete from rmsman.gards_qc_results where sample_id = :old.sample_id;
	END IF;
	end;

5.3.87. GARDS_RESOLUTION_CAL_COV_I_U_D

Table	GARDS_RESOLUTION_CAL_COV
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_resolution_cal_cov (
	sample_id,
	row_index,
	col_index,
	coeff,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.row_index,
	:new.col_index,
	:new.coeff,
	:new.type,
	:new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_resolution_cal_cov set
	sample_id = :new.sample_id,
	row_index = :new.row_index,
	col_index = :new.col_index,
	coeff = :new.coeff,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and NVL(type,'NULL') = NVL(:old.type,'NULL') and row_index= :old.row_index and col_index= :old.col_index;
	ELSE
	delete from rmsman.gards_resolution_cal_cov where sample_id = :old.sample_id and NVL(type,'NULL') = NVL(:old.-

	type,'NULL')
	and row_index= :old.row_index and
	col_index= :old.col_index;
	END IF;
	end;

5.3.88. GARDS_RESOLUTION_CAL_I_U_D

Table	GARDS_RESOLUTION_CAL
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_resolution_cal (
	sample_id,
	coeff1,
	coeff2,
	coeff3,
	coeff4,
	coeff5,
	coeff6,
	coeff7,
	coeff8,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.coeff1,
	:new.coeff2,
	:new.coeff3,
	:new.coeff4,
	:new.coeff5,
	:new.coeff6,
	:new.coeff7,
	:new.coeff8,
	:new.type,
	:new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_resolution_cal set
	sample_id = :new.sample_id,
	coeff1 = :new.coeff1,
	coeff2 = :new.coeff2,
	coeff3 = :new.coeff3,
	coeff4 = :new.coeff4,
	coeff5 = :new.coeff5,
	coeff6 = :new.coeff6,
	coeff7 = :new.coeff7,
	coeff8 = :new.coeff8,
	type = :new.type,

	winner = :new.winner
	where sample_id= :old.sample_id and NVL(type,'NULL') = NVL(:old.type,'NULL');
	ELSE
	delete from rmsman.gards_resolution_cal where sample_id = :old.sample_id and NVL(type,'NULL') = NVL(:old.-type,'NULL');
	END IF;
	end;

5.3.89. GARDS_RESOLUTION_PAIRS_I_U_D

Table	GARDS_RESOLUTION_PAIRS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_resolution_pairs (
	sample_id,
	res_energy,
	resolution,
	res_error,
	type,
	winner
	)
	values (
	:new.sample_id,
	:new.res_energy,
	:new.resolution,
	:new.res_error,
	:new.type,
	:new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_resolution_pairs set
	sample_id = :new.sample_id,
	res_energy = :new.res_energy,
	resolution = :new.resolution,
	res_error = :new.res_error,
	type = :new.type,
	winner = :new.winner
	where sample_id= :old.sample_id and res_energy = :old.res_energy and NVL(type,'NULL') = NVL(:old.type,'NULL');
	ELSE
	delete from rmsman.gards_resolution_pairs where sample_id = :old.sample_id and res_energy = :old.res_en-ergy and NVL(type,'NULL') = NVL(:old.type,'NULL');
	END IF;
	end;

5.3.90. GARDS_RES_CAL_ORIG_I_U_D

Table	GARDS_RESOLUTION_CAL_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_resolution_cal_orig (
	  sample_id,
	  coeff1,
	  coeff2,
	  coeff3,
	  coeff4,
	  coeff5,
	  coeff6,
	  coeff7,
	  coeff8,
	  type,
	  winner
	)
	values (
	  :new.sample_id,
	  :new.coeff1,
	  :new.coeff2,
	  :new.coeff3,
	  :new.coeff4,
	  :new.coeff5,
	  :new.coeff6,
	  :new.coeff7,
	  :new.coeff8,
	  :new.type,
	  :new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_resolution_cal_orig set
	  sample_id = :new.sample_id,
	  coeff1 = :new.coeff1,
	  coeff2 = :new.coeff2,
	  coeff3 = :new.coeff3,
	  coeff4 = :new.coeff4,
	  coeff5 = :new.coeff5,
	  coeff6 = :new.coeff6,
	  coeff7 = :new.coeff7,
	  coeff8 = :new.coeff8,
	  type = :new.type,
	  winner = :new.winner
	where sample_id= :old.sample_id and NVL(type,'NULL') =
	NVL(:old.type,'NULL');
	ELSE
	  delete from rmsman.gards_resolution_cal_orig
	where sample_id = :old.sample_id and NVL(type,'NULL') =
	NVL(:old.type,'NULL');

	END IF; end;

5.3.91. GARDS_RES_PAIRS_ORIG_I_U_D

Table	GARDS_RESOLUTION_PAIRS_ORIG
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_resolution_pairs_orig (
	  sample_id,
	  res_energy,
	  resolution,
	  res_error,
	  type,
	  winner
	)
	values (
	  :new.sample_id,
	  :new.res_energy,
	  :new.resolution,
	  :new.res_error,
	  :new.type,
	  :new.winner
	);
	ELSIF UPDATING THEN
	update rmsman.gards_resolution_pairs_orig set
	  sample_id = :new.sample_id,
	  res_energy = :new.res_energy,
	  resolution = :new.resolution,
	  res_error = :new.res_error,
	  type = :new.type,
	  winner = :new.winner
	where sample_id= :old.sample_id and res_energy = :new.res_energy and NVL(type,'NULL') = NVL(:old.type,'NULL');
	ELSE
	delete from rmsman.gards_resolution_pairs_orig where sample_id = :old.sample_id and res_energy = :new.res_energy and NVL(type,'NULL') = NVL(:old.type,'NULL');
	END IF;
	end;

5.3.92. GARDS_RLR_B_I_U

Table	GARDS_RLR
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE
Trigger body	begin

	:new.moddate := sysdate; end;

5.3.93. GARDS_ROI_CHANNELS_I_U_D

Table	GARDS_ROI_CHANNELS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_roi_channels (
	sample_id, roi, b_chan_start, b_chan_stop, g_chan_start, g_chan_stop
	)
	values (
	:new.sample_id, :new.roi, :new.b_chan_start, :new.b_chan_stop, :new.g_chan_start, :new.g_chan_stop);
	ELSIF UPDATING THEN
	update rmsman.gards_roi_channels set
	sample_id = :new.sample_id,
	roi = :new.sample_id,
	b_chan_start = :new.b_chan_start,
	b_chan_stop = :new.b_chan_stop,
	g_chan_start = :new.g_chan_start,
	g_chan_stop = :new.g_chan_stop
	where sample_id= :old.sample_id and roi = :old.roi;
	ELSE
	delete from rmsman.gards_roi_channels where sample_id = :old.sample_id and roi = :old.roi;
	END IF;
	end;

5.3.94. GARDS_ROI_CONCS_I_U_D

Table	GARDS_ROI_CONCS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_roi_concs (
	sample_id, roi, activity, activ_err, mda, nid_flag, report_mda
	)
	values (
	:new.sample_id, :new.roi, :new.activity, :new.activ_err, :new.mda, :new.nid_flag, :new.report_mda
	);
	ELSIF UPDATING THEN
	update rmsman.gards_roi_concs set

	sample_id = :new.sample_id,
	roi = :new.sample_id,
	activity = :new.activity,
	activ_err = :new.activ_err,
	mda = :new.mda,
	nid_flag = :new.nid_flag,
	report_mda = :new.report_mda
	where sample_id= :old.sample_id and roi
	= :old.roi;
	ELSE
	delete from rmsman.gards_roi_concs where sample_id
	= :old.sample_id and roi = :old.roi;
	END IF;
	end;

5.3.95. GARDS_ROI_COUNTS_I_U_D

Table	GARDS_ROI_COUNTS
Type	AFTER EACH ROW
Event	INSERT OR UPDATING OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_roi_counts (
	sample_id, roi, gross, gross_err, compton,
	compton_err,
	interference, interference_err, memory,
	memory_err, detector_back, detector_back_err, net, net_err, lc
	)
	values (
	:new.sample_id, :new.roi, :new.gross, :new.-
	gross_err, :new.compton,
	:new.compton_err, :new.interference, :new.inter-
	ference_err, :new.memory,
	:new.memory_err, :new.detector_back, :new.de-
	tector_back_err, :new.net, :new.net_err, :new.lc
	);
	ELSIF UPDATING THEN
	update rmsman.gards_roi_counts set
	sample_id = :new.sample_id,
	roi = :new.sample_id,
	gross = :new.gross,
	gross_err = :new.gross_err,
	compton = :new.compton,
	compton_err = :new.compton_err,
	interference = :new.interference,
	interference_err = :new.interference_err,
	memory = :new.memory,
	memory_err = :new.memory_err,
	detector_back = :new.detector_back,
	detector_back_err = :new.detector_back_err,
	net = :new.net,

	net_err = :new.net_err,
	lc = :new.lc
	where sample_id= :old.sample_id and roi
	= :old.roi;
	ELSE
	delete from rmsman.gards_roi_counts where
	sample_id = :old.sample_id and roi = :old.roi;
	END IF;
	end;

5.3.96. GARDS_ROI_LIMITS_I_U_D

Table	GARDS_ROI_LIMITS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_roi_limits (
	sample_id, roi, b_energy_start, b_energy_stop,
	g_energy_start, g_energy_stop
	)
	values (
	:new.sample_id, :new.roi, :new.b_energy_start,
	:new.b_energy_stop, :new.g_energy_start, :new.g_
	energy_stop
	);
	ELSIF UPDATING THEN
	update rmsman.gards_roi_limits set
	sample_id = :new.sample_id,
	roi = :new.sample_id,
	b_energy_start = :new.b_energy_start,
	b_energy_stop = :new.b_energy_stop,
	g_energy_start = :new.g_energy_start,
	g_energy_stop = :new.g_energy_stop
	where sample_id= :old.sample_id and roi
	= :old.roi;
	ELSE
	delete from rmsman.gards_roi_limits where
	sample_id = :old.sample_id and roi = :old.roi;
	END IF;
	end;

5.3.97. GARDS_SAINT_PROC_PAR_I_U_D

Table	GARDS_SAINT_PROCESS_PARAMS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_saint_process_params

	(sample_id, name, value)
	values (:new.sample_id, :new.name, :new.value);
	ELSIF UPDATING THEN
	update rmsman.gards_saint_process_params set
	sample_id = :new.sample_id,
	name = :new.name,
	value = :new.value
	where sample_id= :old.sample_id and name = :old.name;
	ELSE
	delete from rmsman.gards_saint_process_params where
	sample_id = :old.sample_id and name = :old.name;
	END IF;
	end;

5.3.98. GARDS_SAMPLE_CAT_I_U_D

Table	GARDS_AUTO_SAMPLE_CAT
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_sample_cat
	(sample_id,
	name,
	method_id,
	category,
	upper_bound,
	lower_bound,
	central_value,
	delta,
	activity,
	hold)
	values (
	:new.sample_id,
	:new.name,
	:new.method_id,
	:new.category,
	:new.upper_bound,
	:new.lower_bound,
	:new.central_value,
	:new.delta,
	:new.activity,
	:new.hold);
	ELSIF UPDATING THEN
	update rmsman.gards_sample_cat set
	sample_id = :new.sample_id,
	name = :new.name,
	method_id = :new.method_id,
	category = :new.category,
	upper_bound = :new.upper_bound,
	lower_bound = :new.lower_bound,

	central_value = :new.central_value,
	delta = :new.delta,
	activity = :new.activity,
	hold = :new.hold
	where sample_id= :old.sample_id and name = :old.name;
	ELSE
	delete from rmsman.gards_sample_cat where sample_id = :old.sample_id and name = :old.name;
	END IF;
	end;

5.3.99. GARDS_SAMPLE_DATA_I_U_D

Table	GARDS_SAMPLE_DATA
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_sample_data
	
	(site_det_code,sample_id,station_id,detector_id,input_file_name,
	
	sample_type,data_type,geometry,spectral_qualifier,transmit_dtg,
	collect_start,collect_stop,acquisition_start,acquisition_stop,
	acquisition_real_sec,acquisition_live_sec,quantity)
	
	values
	(:new.site_det_code,:new.sample_id,:new.station_id,:new.detector_id,
	:new.input_file_name,:new.sample_type,:new.data_type,:new.geometry,
	:new.spectral_qualifier,:new.transmit_dtg,:new.collect_start,:new.collect_stop,
	:new.acquisition_start,:new.acquisition_stop,:new.acquisition_real_sec,
	:new.acquisition_live_sec,:new.quantity);
	ELSIF UPDATING THEN
	update rmsman.gards_sample_data set
	site_det_code = :new.site_det_code,
	sample_id = :new.sample_id,
	station_id = :new.station_id,
	detector_id = :new.detector_id,
	input_file_name = :new.input_file_name,
	sample_type = :new.sample_type,

	data_type = :new.data_type,
	geometry = :new.geometry,
	spectral_qualifier = :new.spectral_qualifier,
	transmit_dtg = :new.transmit_dtg,
	collect_start = :new.collect_start,
	collect_stop = :new.collect_stop,
	acquisition_start = :new.acquisition_start,
	acquisition_stop = :new.acquisition_stop,
	acquisition_real_sec = :new.acquisition_real_sec,
	acquisition_live_sec = :new.acquisition_live_sec,
	quantity = :new.quantity
	where sample_id= :old.sample_id;
	ELSE
	delete from rmsman.gards_sample_data where sample_id = :old.sample_id;
	END IF;
	end;

5.3.100. GARDS_SAMPLE_FLAGS_I_U_D

Table	GARDS_SAMPLE_FLAGS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_sample_flags (
	sample_id,
	flag_id,
	result,
	value
	)
	values (
	:new.sample_id,
	:new.flag_id,
	:new.result,
	:new.value
	);
	ELSIF UPDATING THEN
	update rmsman.gards_sample_flags set
	sample_id = :new.sample_id,
	flag_id = :new.flag_id,
	result = :new.result,
	value = :new.value
	where sample_id = :old.sample_id and flag_id = :old.flag_id;
	ELSE
	delete from rmsman.gards_sample_flags where sample_id = :old.sample_id and flag_id = :old.flag_id;
	END IF;
	end;

5.3.101. GARDS_SAMPLE_PROC_PARAMS_I_U_D

Table	GARDS_SAMPLE_PROC_PARAMS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	

```
begin
      IF INSERTING THEN
            insert into rmsman.gards_sample_proc_params (
              sample_id, do_back, back_data_type,
back_sample_id, nuclide_lib,
              energy_tol, mda_level, nid_confid, squant_err,
buildtype,
              peak_sense, peak_start, peak_end,
fwhm_mult_width,
              left_fwhm_lim, right_fwhm_lim, back_chan,
back_type,
              fit_singlets, crit_level, fix_fwhm, area_reject,
mdc_width,
              lc_abscissa, do_pd_calc, do_csc)
            values (
              :new.sample_id, :new.do_back, :new.back_data_typ
e, :new.back_sample_id, :new.nuclide_lib,
              :new.energy_tol, :new.mda_level, :new.nid_con-
fid, :new.squant_err, :new.buildtype,
              :new.peak_sense, :new.peak_start, :new.peak_end,
:new.fwhm_mult_width,
              :new.left_fwhm_lim, :new.right_fwhm_lim, :new.-
back_chan, :new.back_type,

        :new.fit_singlets, :new.crit_level, :new.fix_fwhm, :ne
w.area_reject, :new.mdc_width,
              :new.lc_abscissa, :new.do_pd_calc, :new.do_csc);
      ELSIF UPDATING THEN
            update rmsman.gards_sample_proc_params set
              sample_id       = :new.sample_id,
              do_back         = :new.do_back,
              back_data_type  = :new.back_data_type,
              back_sample_id  = :new.back_sample_id,
              nuclide_lib     = :new.nuclide_lib,
              energy_tol      = :new.energy_tol,
              mda_level       = :new.mda_level,
              nid_confid      = :new.nid_confid,
              squant_err      = :new.squant_err,
              buildtype       = :new.buildtype,
              peak_sense      = :new.peak_sense,
              peak_start      = :new.peak_start,
              peak_end        = :new.peak_end,
              fwhm_mult_width = :new.fwhm_mult_width,
              left_fwhm_lim   = :new.left_fwhm_lim,
              right_fwhm_lim  = :new.right_fwhm_lim,
              back_chan       = :new.back_chan,
              back_type       = :new.back_type,
```

	fit_singlets = :new.fit_singlets,
	crit_level = :new.crit_level,
	fix_fwhm = :new.fix_fwhm,
	area_reject = :new.area_reject,
	mdc_width = :new.mdc_width,
	lc_abscissa = :new.lc_abscissa,
	do_pd_calc = :new.do_pd_calc,
	do_csc = :new.do_csc
	where sample_id= :old.sample_id;
	ELSE
	delete from rmsman.gards_sample_proc_params where sample_id = :old.sample_id;
	END IF;
	end;

5.3.102. GARDS_SAMPLE_RATIOS_I_U_D

Table	GARDS_SAMPLE_RATIOS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_sample_ratios (
	sample_id, ratio_id, upper_roi_number, lower_roi_number, count_ratio, count_ratio_err
	)
	values (
	:new.sample_id, :new.ratio_id, :new.upper_roi_number, :new.lower_roi_number,
	:new.count_ratio, :new.count_ratio_err
	);
	ELSIF UPDATING THEN
	update rmsman.gards_sample_ratios set
	sample_id = :new.sample_id,
	ratio_id = :new.ratio_id,
	upper_roi_number = :new.upper_roi_number,
	lower_roi_number = :new.lower_roi_number,
	count_ratio = :new.count_ratio,
	count_ratio_err = :new.count_ratio_err
	where sample_id= :old.sample_id and ratio_id = :old.ratio_id;
	ELSE
	delete from rmsman.gards_sample_ratios where sample_id = :old.sample_id and ratio_id = :old.ratio_id;
	END IF;
	end;

5.3.103. GARDS_SAMPLE_STATUS_I_U_D

Table	GARDS_SAMPLE_STATUS

Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_sample_status (
	sample_id,entry_date,cnf_begin_date,
	
	cnf_end_date,review_date,review_time,analyst,status,category,auto_category
	)
	values (
	:new.sample_id, :new.entry_date, :new.cnf_begin_date,
	:new.cnf_end_date,:new.review_date,:new.review_time, :new.analyst, :new.status,
	:new.category,:new.auto_category
	);
	ELSIF UPDATING THEN
	update rmsman.gards_sample_status set
	sample_id = :new.sample_id,
	entry_date = :new.entry_date,
	cnf_begin_date = :new.cnf_begin_date,
	cnf_end_date = :new.cnf_end_date,
	review_date = :new.review_date,
	review_time = :new.review_time,
	analyst = :new.analyst,
	status = :new.status,
	category = :new.category,
	auto_category = :new.auto_category
	where sample_id= :old.sample_id;
	ELSE
	delete from rmsman.gards_sample_status where sample_id = :old.sample_id;
	END IF;
	end;

5.3.104. GARDS_SAMP_UPD_PRMS_I_U_D

Table	GARDS_SAMPLE_UPDATE_PARAMS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_sample_update_params (
	sample_id, mrp_used, mrp_sample_id, gainshift, zeroshift, area_lim,
	use_weight, use_mult, f_linear, bootstrap, min_lookup, rer_intercept,
	rer_slope, ecr_slope, do_reru
	)

```
            values (
              :new.sample_id, :new.mrp_used, :new.mrp_sample_i
d, :new.gainshift, :new.zeroshift,
              :new.area_lim, :new.use_weight, :new.use_mult, :
new.f_linear, :new.bootstrap,

        :new.min_lookup, :new.rer_intercept, :new.rer_slope, :
new.ecr_slope, :new.do_reru
            );
      ELSIF UPDATING THEN
            update rmsman.gards_sample_update_params set
              sample_id     = :new.sample_id,
              mrp_used      = :new.mrp_used,
              mrp_sample_id = :new.mrp_sample_id,
              gainshift     = :new.gainshift,
              zeroshift     = :new.zeroshift,
              area_lim      = :new.area_lim,
              use_weight    = :new.use_weight,
              use_mult      = :new.use_mult,
              f_linear      = :new.f_linear,
              bootstrap     = :new.bootstrap,
              min_lookup    = :new.min_lookup,
              rer_intercept = :new.rer_intercept,
              rer_slope     = :new.rer_slope,
              ecr_slope     = :new.ecr_slope,
              do_reru       = :new.do_reru
            where sample_id= :old.sample_id;
      ELSE
            delete from rmsman.gards_sample_update_params
where sample_id = :old.sample_id;
      END IF;
end;
```

5.3.105. GARDS_SMPLE_XE_P_PRMS_I_U_D

Table	GARDS_SAMPLE_XE_PROC_PARAMS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE
Trigger body	

```
begin
      IF INSERTING THEN
            insert into rmsman.gards_sample_xe_proc_params (
              sample_id, roi, lc_abscissa,beta_ecr_order,
gamma_ecr_order, compton, det_back_used, gas_back_used
            )
            values (
              :new.sample_id, :new.roi, :new.lc_abscissa,:new.
beta_ecr_order, :new.gamma_ecr_order, :new.compton,
              :new.det_back_used, :new.gas_back_used
            );
      ELSIF UPDATING THEN
            update rmsman.gards_sample_xe_proc_params set
```

	sample_id = :new.sample_id,
	roi = :new.sample_id,
	lc_abscissa = :new.lc_abscissa,
	beta_ecr_order = :new.beta_ecr_order,
	gamma_ecr_order = :new.gamma_ecr_order,
	compton = :new.compton,
	det_back_used = :new.det_back_used,
	gas_back_used = :new.gas_back_used
	where sample_id= :old.sample_id and roi
	= :old.roi;
	ELSE
	delete from rmsman.gards_sample_xe_proc_params
	where sample_id = :old.sample_id and roi = :old.roi;
	END IF;
	end;

5.3.106. GARDS_SOH_EVALUATION_I_U_D

Table	GARDS_SOH_EVALUATION
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_soh_evaluation
	
	(sample_id,srid,soh_status,soh_flag,author,user_comment, ld-
	date)
	values
	(:new.sample_id,:new.srid,:new.soh_status,:new.soh_flag,
	:new.author,:new.user_comment,:new.lddate);
	ELSIF UPDATING THEN
	update rmsman.gards_soh_evaluation set
	sample_id = :new.sample_id,
	srid = :new.srid,
	soh_status = :new.soh_status,
	soh_flag = :new.soh_flag,
	author= :new.author,
	user_comment = :new.user_comment,
	lddate = :new.lddate
	where sample_id= :old.sample_id;
	ELSE
	delete from rmsman.gards_soh_evaluation where
	sample_id = :old.sample_id;
	END IF;
	end;

5.3.107. GARDS_TOTAL_EFFIC_I_U_D

Table	GARDS_TOTAL_EFFIC
Type	AFTER EACH ROW

Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_total_effic (
	sample_id, effic_energy, efficiency , effic_error
	)
	values (
	:new.sample_id, :new.effic_energy, :new.efficiency , :new.effic_error
	);
	ELSIF UPDATING THEN
	update rmsman.gards_total_effic set
	sample_id = :new.sample_id ,
	effic_energy = :new.effic_energy,
	efficiency = :new.efficiency,
	effic_error = :new.effic_error
	where sample_id= :old.sample_id and effic_energy = :old.effic_energy;
	ELSE
	delete from rmsman.gards_total_effic where sample_id = :old.sample_id and effic_energy = :old.effic_energy;
	END IF;
	
	end;

5.3.108. GARDS_USER_COMMENTS_I_U_D

Table	GARDS_USER_COMMENTS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_user_comments (
	comment_id,
	comment_text,
	sample_id
	) values (
	:new.comment_id,
	:new.comment_text,
	:new.sample_id
	);
	ELSIF UPDATING THEN
	update rmsman.gards_user_comments set
	comment_id = :new.comment_id,
	comment_text = :new.comment_text,
	sample_id = :new.sample_id
	;
	ELSE

	delete from rmsman.gards_user_comments where comment_id = :old.comment_id;
	END IF;
	
	end;

5.3.109. GARDS_USER_DELETE

Table	GARDS_USERS
Type	AFTER EACH ROW
Event	DELETE

Trigger body	begin
	
	update gards_station_assignments set user_id = null where user_id = :old.user_id;
	
	end;

5.3.110. GARDS_XE_RESULTS_I_U_D

Table	GARDS_XE_RESULTS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_xe_results
	(sample_id,
	method_id,
	nuclide_id,
	conc,
	conc_err,
	mdc,
	mdi,
	activity_acq_start,
	activity_err_acq_start,
	mda_acq_start,
	nid_flag,
	lc,
	ld,
	sample_act,
	cov_xe_131m,
	cov_xe_133m,
	cov_xe_133,
	cov_xe_135,
	cov_radon)
	VALUES (
	:new.sample_id,
	:new.method_id,
	:new.nuclide_id,
	:new.conc,

```
                        :new.conc_err,
                        :new.mdc,
                        :new.mdi,
                        :new.activity_acq_start,
                        :new.activity_err_acq_start,
                        :new.mda_acq_start,
                        :new.nid_flag,
                        :new.lc,
                        :new.ld,
                        :new.sample_act,
                        :new.cov_xe_131m,
                        :new.cov_xe_133m,
                        :new.cov_xe_133,
                        :new.cov_xe_135,
                        :new.cov_radon);
                ELSIF UPDATING THEN
                        update rmsman.gards_xe_results set
                        sample_id = :new.sample_id,
                        method_id = :new.method_id,
                        nuclide_id = :new.nuclide_id,
                        conc = :new.conc,
                        conc_err = :new.conc_err,
                        mdc = :new.mdc,
                        mdi = :new.mdi,
                        activity_acq_start
= :new.activity_acq_start,
                        activity_err_acq_start = :new.activ-
ity_err_acq_start,
                        mda_acq_start = :new.mda_acq_start,
                        nid_flag = :new.nid_flag,
                        lc = :new.lc,
                        ld = :new.ld,
                        sample_act = :new.sample_act,
                        cov_xe_131m = :new.cov_xe_131m,
                        cov_xe_133m = :new.cov_xe_133m,
                        cov_xe_133 = :new.cov_xe_133,
                        cov_xe_135 = :new.cov_xe_135,
                        cov_radon = :new.cov_radon
                     where sample_id= :old.sample_id;
                ELSE
                        delete from rmsman.gards_xe_results
where
                                sample_id = :old.sample_id;
                END IF;
        end;
```

5.3.111. GARDS_XE_SAMPLE_CATEGORY_I_U

Table	GARDS_XE_SAMPLE_CATEGORY
Type	AFTER EACH ROW
Event	INSERT OR UPDATE

Trigger body	BEGIN
	  IF INSERTING THEN
	  INSERT INTO rmsman.GARDS_xe_sample_category
	  (
	sample_id ,
	prev_valid_samples ,
	abn_xe131m ,
	abn_xe133m ,
	abn_xe133 ,
	abn_xe135 ,
	median_xe131m ,
	median_xe133m ,
	median_xe133 ,
	median_xe135 ,
	bayes_133m_131m ,
	bayes_133m_133 ,
	bayes_135_133 ,
	conc_133m_131m,
	conc_133m_133,
	conc_135_133,
	conc_133_131m_mdc,
	cat_xe131m ,
	cat_xe133m ,
	cat_xe133 ,
	cat_xe135 ,
	flag_133m_131m ,
	flag_133m_133 ,
	flag_135_133 ,
	cat_sample ,
	flag_atm ,
	flag_shortterm,
	moddate
	  )
	  VALUES
	  (
	:new.sample_id ,
	:new.prev_valid_samples ,
	:new.abn_xe131m ,
	:new.abn_xe133m ,
	:new.abn_xe133 ,
	:new.abn_xe135 ,
	:new.median_xe131m ,
	:new.median_xe133m ,
	:new.median_xe133 ,
	:new.median_xe135 ,
	:new.bayes_133m_131m ,
	:new.bayes_133m_133 ,
	:new.bayes_135_133 ,
	:new.conc_133m_131m,
	:new.conc_133m_133,
	:new.conc_135_133,
	:new.conc_133_131m_mdc,
	:new.cat_xe131m ,

	:new.cat_xe133m ,
	:new.cat_xe133 ,
	:new.cat_xe135 ,
	:new.flag_133m_131m ,
	:new.flag_133m_133 ,
	:new.flag_135_133 ,
	:new.cat_sample ,
	:new.flag_atm ,
	:new.flag_shortterm,
	:new.moddate);
	ELSIF UPDATING THEN
	UPDATE rmsman.GARDS_xe_sample_category
	SET
	prev_valid_samples = :new.prev_valid_samples ,
	abn_xe131m = :new.abn_xe131m ,
	abn_xe133m = :new.abn_xe133m ,
	abn_xe133 = :new.abn_xe133 ,
	abn_xe135 = :new.abn_xe135 ,
	median_xe131m = :new.median_xe131m ,
	median_xe133m = :new.median_xe133m ,
	median_xe133 = :new.median_xe133 ,
	median_xe135 = :new.median_xe135 ,
	bayes_133m_131m = :new.bayes_133m_131m ,
	bayes_133m_133 = :new.bayes_133m_133 ,
	bayes_135_133 = :new.bayes_135_133 ,
	conc_133m_131m = :new.conc_133m_131m,
	conc_133m_133 = :new.conc_133m_133,
	conc_135_133 = :new.conc_135_133,
	conc_133_131m_mdc = :new.conc_133_131m_mdc,
	cat_xe131m = :new.cat_xe131m ,
	cat_xe133m = :new.cat_xe133m ,
	cat_xe133 = :new.cat_xe133 ,
	cat_xe135 = :new.cat_xe135 ,
	flag_133m_131m = :new.flag_133m_131m ,
	flag_133m_133 = :new.flag_133m_133 ,
	flag_135_133 = :new.flag_135_133 ,
	cat_sample = :new.cat_sample ,
	flag_atm = :new.flag_atm ,
	moddate = :new.moddate
	WHERE sample_id= :old.sample_id and flag_shortterm=:old.-
	flag_shortterm;
	END IF;
	END;

5.3.112. GARDS_XE_UNCORR_RES_A_I_U

Table	GARDS_XE_UNCORRECTED_RESULTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE
Trigger body	begin :new.moddate := sysdate;

	end;

5.3.113. GARDS_XE_UNCORR_RES_I_U_D

Table	GARDS_XE_UNCORRECTED_RESULTS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE OR DELETE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsman.gards_xe_uncorrected_results (
	sample_id,
	method_id,
	nuclide_id,
	conc,
	conc_error,
	mdc,
	mdi,
	lc
	)
	values (
	:new.sample_id,
	:new.method_id,
	:new.nuclide_id,
	:new.conc,
	:new.conc_error,
	:new.mdc,
	:new.mdi,
	:new.lc
	);
	ELSIF UPDATING THEN
	update rmsman.gards_xe_uncorrected_results set
	sample_id = :new.sample_id,
	method_id = :new.method_id,
	nuclide_id = :new.nuclide_id,
	conc = :new.conc,
	conc_error = :new.conc_error,
	mdc = :new.mdc,
	mdi = :new.mdi,
	lc = :new.lc
	where sample_id= :old.sample_id and method_id
	= :old.method_id and nuclide_id = :old.nuclide_id;
	ELSE
	delete from rmsman.gards_xe_uncorrected_results
	where sample_id = :old.sample_id and method_id
	= :old.method_id and nuclide_id = :old.nuclide_id;
	END IF;
	end;

5.3.114. HISTOGRAM_B_I_U

Table	GARDS_HISTOGRAM

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.115. INTERVAL_B_I_U

Table	GARDS_INTERVAL
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.116. INTERVAL_S_I

Table	GARDS_INTERVAL
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`declare`
	`new_id number;`
	`begin`
	`select intvl_seq.nextval into new_id from dual;`
	`:new.intvlid := new_id;`
	`end;`

5.3.117. IRF_B_I_U

Table	GARDS_IRF
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.118. LAB_CAT_DESC_B_I_U

Table	GARDS_LAB_CATEGORY_DESCRIPTION
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.119. MET_DATA_B_I_U

Table	GARDS_MET_DATA
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.120. MET_DATA_S_I

Table	GARDS_MET_DATA
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`declare`
	`new_id number;`
	`begin`
	`select met_seq.nextval into new_id from dual;`
	`:new.met_id := new_id;`
	`end;`

5.3.121. MET_HEADER_B_I_U

Table	GARDS_MET_HEADER
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.122. MET_HEADER_S_I

Table	GARDS_MET_HEADER
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`declare`
	`new_id number;`
	`begin`
	`select met_header_seq.nextval into new_id from dual;`
	`:new.met_group_id := new_id;`
	`end;`

5.3.123. NOTIFY_B_I_U

Table	GARDS_NOTIFY

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.124. NUCL2QUANTIFY_B_I_U

Table	GARDS_NUCL2QUANTIFY
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.125. NUCL_IDED_B_I_U

Table	GARDS_NUCL_IDED
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.126. NUCL_IDED_ORIG_B_I_U

Table	GARDS_NUCL_IDED_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.127. NUCL_LIB_B_I_U

Table	GARDS_NUCL_LIB
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.128. NUCL_LINES_IDED_B_I_U

Table	GARDS_NUCL_LINES_IDED

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.129. NUCL_LINES_IDED_ORIG_B_I_U

Table	GARDS_NUCL_LINES_IDED_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.130. NUCL_LINES_LIB_B_I_U

Table	GARDS_NUCL_LINES_LIB
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.131. PEAKS_B_I_U

Table	GARDS_PEAKS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.132. PEAKS_ORIG_B_I_U

Table	GARDS_PEAKS_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.133. PERMISSIONS_B_I_U

Table	GARDS_PERMISSIONS

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.134. POC_B_I_U

Table	GARDS_POC
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.135. PROCESSING_ERRORS_B_I_U

Table	GARDS_PROCESSING_ERRORS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.136. PROCESS_PARAMS_B_I_U

Table	GARDS_SAINT_PROCESS_PARAMS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.137. PROC_PARAMS_TEMPLATE_B_I_U

Table	GARDS_PROC_PARAMS_TEMPLATE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.138. PRODUCT_B_I_U

Table	GARDS_PRODUCT

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.139. QAT_CONFIG_B_I_U

Table	GARDS_QAT_CONFIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.140. QAT_QUERY_FILTER_B_I_U

Table	GARDS_QAT_QUERY_FILTER
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.141. QCHISTORY_B_I_U

Table	GARDS_QCHISTORY
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.142. QCPARAMS_B_I_U

Table	GARDS_QCPARAMS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.143. QCTARGETS_B_I_U

Table	GARDS_QCTARGETS

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.144. QC_RESULTS_B_I_U

Table	GARDS_QC_RESULTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.145. QUERY_RESULTS_B_I_U

Table	GARDS_QUERY_RESULTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.146. REFLINE_MASTER_B_I_U

Table	GARDS_REFLINE_MASTER
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.147. RELEVANT_NUCLIDES_B_I_U

Table	GARDS_RELEVANT_NUCLIDES
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.148. RESOLUTION_CAL_B_I_U

Table	GARDS_RESOLUTION_CAL

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.149. RESOLUTION_CAL_COV_B_I_U

Table	GARDS_RESOLUTION_CAL_COV
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.150. RESOLUTION_CAL_ORIG_B_I_U

Table	GARDS_RESOLUTION_CAL_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.151. RESOLUTION_PAIRS_B_I_U

Table	GARDS_RESOLUTION_PAIRS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.152. RESOLUTION_PAIRS_ORIG_B_I_U

Table	GARDS_RESOLUTION_PAIRS_ORIG
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.153. RESOLUTION_PAIRS_ORIG_B_I_U

Table	GARDS_RESOLUTION_PAIRS_ORIG

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.154. RLR_CONCLUSIONS_B_I_U

Table	GARDS_RLR_CONCLUSIONS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.155. RLR_OBJECTIVE_B_I_U

Table	GARDS_RLR_OBJECTIVE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.156. RLR_RATIOS_B_I_U

Table	GARDS_RLR_RATIOS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.157. RLR_RESULTS_B_I_U

Table	GARDS_RLR_RESULTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.158. RLR_SSREB_B_I_U

Table	GARDS_RLR_SSREB

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.159. RN_INTERVAL_S_I

Table	GARDS_RN_INTERVAL
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`declare`
	`new_id number;`
	`begin`
	`select rn_intvl_seq.nextval into new_id from dual;`
	`:new.intvlid := new_id;`
	`end;`

5.3.160. ROI_CHANNELS_B_I_U

Table	GARDS_ROI_CHANNELS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.161. ROI_CONCS_B_I_U

Table	GARDS_ROI_CONCS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.162. ROI_COUNTS_B_I_U

Table	GARDS_ROI_COUNTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.163. ROI_LIB_B_I_U

Table	GARDS_ROI_LIB
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.164. ROI_LIMITS_B_I_U

Table	GARDS_ROI_LIMITS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.165. ROLES_B_I_U

Table	GARDS_ROLES
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.166. ROLES_PERMISSIONS_B_I_U

Table	GARDS_ROLES_PERMISSIONS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.167. SAMPLE_AUX_B_I_U

Table	GARDS_SAMPLE_AUX
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.168. SAMPLE_CAT_B_I_U

Table	GARDS_SAMPLE_CAT
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.169. SAMPLE_CERT_B_I_U

Table	GARDS_SAMPLE_CERT
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.170. SAMPLE_CERT_LINES_B_I_U

Table	GARDS_SAMPLE_CERT_LINES
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.171. SAMPLE_DATA_B_I_U

Table	GARDS_SAMPLE_DATA
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.172. SAMPLE_DATA_INSERT

Table	GARDS_SAMPLE_DATA
Type	AFTER EACH ROW
Event	INSERT

Trigger body	`begin`
	
	`  if newstadet(:new.station_id, :new.detector_id) then`
	`  insert into gards_stadet (`

	station_id,
	detector_id
	)
	values (
	:new.station_id,
	:new.detector_id
	);
	end if;
	end;

5.3.173. SAMPLE_DESCRIPTION_B_I_U

Table	GARDS_SAMPLE_DESCRIPTION
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.174. SAMPLE_FLAGS_B_I_U

Table	GARDS_SAMPLE_FLAGS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.175. SAMPLE_PROC_PARAMS_B_I_U

Table	GARDS_SAMPLE_PROC_PARAMS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.176. SAMPLE_RATIOS_B_I_U

Table	GARDS_SAMPLE_RATIOS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.177. SAMPLE_STATUS_B_I_U

Table	GARDS_SAMPLE_STATUS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.178. SAMPLE_STATUS_UPDATE

Table	GARDS_SAMPLE_STATUS
Type	AFTER EACH ROW
Event	UPDATE

Trigger body	`begin`
	
	`  IF :new.category >= 4 AND :old.category IS NULL AND :new.status = 'R' THEN`
	`  insert into gards_fpe (`
	`  fpid,`
	`  revid,`
	`  dtg,`
	`  sample_id`
	`  )`
	`  values (`
	`  gards_fpid_seq.nextval,`
	`  1,`
	`  sysdate,`
	`  :new.sample_id`
	`  );`
	`  END IF;`
	`end;`

5.3.179. SAMPLE_UPDATE_PARAMS_B_I_U

Table	GARDS_SAMPLE_UPDATE_PARAMS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.180. SAMPLE_XE_PROC_PARAMS_B_I_U

Table	GARDS_SAMPLE_XE_PROC_PARAMS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.181. SOH_CHAR_DATA_B_I_U

Table	GARDS_SOH_CHAR_DATA
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.182. SOH_CHAR_DATA_S_I

Table	GARDS_SOH_CHAR_DATA
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`declare`
	`  new_id number;`
	`begin`
	`  select soh_char_seq.nextval into new_id from dual;`
	`  :new.soh_char_id := new_id;`
	`end;`

5.3.183. SOH_CODE_B_I_U

Table	GARDS_SOH_CODE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.184. SOH_EVALUATION_B_I

Table	GARDS_SOH_EVALUATION
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`begin`
	`  :new.lddate := sysdate;`
	`end;`

5.3.185. SOH_EVALUATION_B_I_U

Table	GARDS_SOH_EVALUATION

Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.186. SOH_HEADER_B_I_U

Table	GARDS_SOH_HEADER
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.187. SOH_NUM_DATA_B_I_U

Table	GARDS_SOH_NUM_DATA
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.188. SOH_NUM_DATA_S_I

Table	GARDS_SOH_NUM_DATA
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	`declare`
	`new_id number;`
	`begin`
	`select soh_num_seq.nextval into new_id from dual;`
	`:new.soh_num_id := new_id;`
	`end;`

5.3.189. SOH_SENSOR_DATA_B_I_U

Table	GARDS_SOH_SENSOR_DATA
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.190. SOH_SENSOR_DATA_S_I

Table	GARDS_SOH_SENSOR_DATA
Type	BEFORE EACH ROW
Event	INSERT

Trigger body	declare
	new_id number;
	begin
	select soh_sensor_seq.nextval into new_id from dual;
	:new.soh_sensor_id := new_id;
	end;

5.3.191. SPECTRUM_B_I_U

Table	GARDS_SPECTRUM
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.192. STADET_B_I_U

Table	GARDS_STADET
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.193. STATIONS_B_I_U

Table	GARDS_STATIONS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.194. STATIONS_INSERT

Table	GARDS_STATIONS
Type	AFTER EACH ROW
Event	INSERT

Trigger body	begin
	insert into gards_station_assignments (
	station_id,
	user_id
	)
	values (
	:new.station_id,
	NULL
	);
	end;

5.3.195. STATIONS_SCHEDULE_B_I_U

Table	GARDS_STATIONS_SCHEDULE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.196. STATION_ASSIGNMENTS_B_I_U

Table	GARDS_STATION_ASSIGNMENTS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.197. STATUS_HISTORY_B_I_U

Table	GARDS_STATUS_HISTORY
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.198. TOTAL_EFFIC_B_I_U

Table	GARDS_TOTAL_EFFIC
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	:new.moddate := sysdate;
	end;

5.3.199. TRENDVUE_B_I_U

Table	GARDS_TRENDVUE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  :new.moddate := sysdate;`
	`end;`

5.3.200. UPDATE_NUCL_ORIG_I_U_D

Table	GARDS_NUCL_IDED
Type	AFTER EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`  IF INSERTING THEN`
	`  insert into rmsauto.gards_nucl_ided_orig (`
	`  sample_id,`
	`  station_id,`
	`  detector_id,`
	`  nuclide_id,`
	`  name,`
	`  type,`
	`  halflife,`
	`  ave_activ,`
	`  ave_activ_err,`
	`  activ_key,`
	`  activ_key_err,`
	`  mda,`
	`  mda_err,`
	`  ave_activ_acq_start,`
	`  ave_activ_err_acq_start,`
	`  activ_key_acq_start,`
	`  activ_key_err_acq_start,`
	`  mda_acq_start,`
	`  mda_err_acq_start,`
	`  nid_flag,`
	`  activ_decay,`
	`  activ_decay_err,`
	`  comp_confid,`
	`  report_mda,`
	`  pd_mod_flag,`
	`  csc_ratio,`
	`  csc_ratio_err,`
	`  csc_mod_flag`
	`  )`
	`  values (`
	`  :new.sample_id,`
	`  :new.station_id,`
	`  :new.detector_id,`

```
                    :new.nuclide_id,
                    :new.name, :new.type,
                    :new.halflife ,
                    :new.ave_activ,
                    :new.ave_activ_err,
                    :new.activ_key,
                    :new.activ_key_err,
                    :new.mda,
                    :new.mda_err,
                    :new.ave_activ_acq_start,
                    :new.ave_activ_err_acq_start,
                    :new.activ_key_acq_start,
                    :new.activ_key_err_acq_start,
                    :new.mda_acq_start,
                    :new.mda_err_acq_start,
                    :new.nid_flag,
                    :new.activ_decay,
                    :new.activ_decay_err,
                    :new.comp_confid,
                    :new.report_mda,
                    :new.pd_mod_flag,
                    :new.csc_ratio,
                    :new.csc_ratio_err,
                    :new.csc_mod_flag
                  );
        ELSIF UPDATING THEN
                  update rmsauto.gards_nucl_ided_orig set
                    sample_id = :new.sample_id,
                    station_id = :new.station_id,
                    detector_id = :new.detector_id,
                    nuclide_id = :new.nuclide_id,
                    name = :new.name,
                    type = :new.type,
                    halflife = :new.halflife ,
                    ave_activ = :new.ave_activ,
                    ave_activ_err = :new.ave_activ_err ,
                    activ_key = :new.activ_key,
                    activ_key_err = :new.activ_key_err,
                    mda = :new.mda,
                    mda_err = :new.mda_err,
                    ave_activ_acq_start
= :new.ave_activ_acq_start,
                    ave_activ_err_acq_start
= :new.ave_activ_err_acq_start ,
                    activ_key_acq_start
= :new.activ_key_acq_start,
                    activ_key_err_acq_start
= :new.activ_key_err_acq_start,
                    mda_acq_start = :new.mda_acq_start,
                    mda_err_acq_start = :new.mda_err_acq_start,
                    nid_flag = :new.nid_flag,
                    activ_decay = :new.activ_decay,
                    activ_decay_err = :new.activ_decay_err,
```

	comp_confid = :new.comp_confid,
	report_mda = :new.report_mda,
	pd_mod_flag = :new.pd_mod_flag,
	csc_ratio = :new.csc_ratio,
	csc_ratio_err = :new.csc_ratio_err,
	csc_mod_flag = :new.csc_mod_flag
	where sample_id= :old.sample_id and name
	= :old.name;
	ELSE
	delete from rmsauto.gards_nucl_ided_orig where
	sample_id = :old.sample_id and name = :old.name;
	END IF;
	end;

5.3.201. UPDATE_NUCL_ORIG_LINES_I_U_D

Table	GARDS_NUCL_LINES_IDED
Type	AFTER EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsauto.gards_nucl_lines_ided_orig (
	sample_id,
	station_id,
	detector_id,
	name,
	energy,
	energy_err,
	abundance,
	abundance_err,
	peak,
	activity,
	activ_err,
	effic,
	effic_err,
	mda,
	key_flag,
	nuclide_id,
	csc_ratio,
	csc_ratio_err,
	csc_mod_flag,
	id_percent
	)
	values (
	:new.sample_id,
	:new.station_id,
	:new.detector_id,
	:new.name,
	:new.energy,
	:new.energy_err,
	:new.abundance,

```
    :new.abundance_err,
    :new.peak,
    :new.activity,
    :new.activ_err,
    :new.effic,
    :new.effic_err,
    :new.mda,
    :new.key_flag,
    :new.nuclide_id,
                  :new.csc_ratio,
                  :new.csc_ratio_err,
                  :new.csc_mod_flag,
                  :new.id_percent
  );
 ELSIF UPDATING THEN
  update rmsauto.gards_nucl_lines_ided_orig set
     sample_id = :new.sample_id,
     station_id = :new.station_id,
     detector_id = :new.detector_id,
     name = :new.name,
     energy = :new.energy,
     energy_err = :new.energy_err,
     abundance = :new.abundance,
     abundance_err = :new.abundance_err,
     peak = :new.peak,
     activity = :new.activity,
     activ_err = :new.activ_err,
     effic = :new.effic ,
     effic_err = :new.effic_err,
     mda = :new.mda,
     key_flag = :new.key_flag,
     nuclide_id = :new.nuclide_id,
                  csc_ratio = :new.csc_ratio,
                  csc_ratio_err = :new.csc_ratio_err,
                  csc_mod_flag = :new.csc_mod_flag,
                  id_percent = :new.id_percent
  where sample_id= :old.sample_id and name = :old.name and en-
ergy = :old.energy;
 ELSE
  delete from rmsauto.gards_nucl_lines_ided_orig where
sample_id = :old.sample_id
   and name = :old.name and energy = :old.energy;
 END IF;
end;
```

5.3.202. UPDATE_PARAMS_TEMPLATE_B_I_U

Table	GARDS_UPDATE_PARAMS_TEMPLATE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE
Trigger body	`begin`

	:new.moddate := sysdate; end;

5.3.203. UPDATE_PEAKS_ORIG_I_U_D

Table	GARDS_PEAKS
Type	AFTER EACH ROW
Event	INSERT OR UPDATE

Trigger body	begin
	IF INSERTING THEN
	insert into rmsauto.gards_peaks_orig (
	sample_id,
	peak_id,
	centroid,
	centroid_err,
	energy, energy_err,
	left_chan,
	width,
	back_count,
	back_uncer,
	fwhm,
	fwhm_err,
	area,
	area_err,
	original_area,
	original_uncer,
	counts_sec,
	counts_sec_err,
	efficiency,
	eff_error,
	back_channel,
	ided,
	fitted,
	multiplet,
	peak_sig,
	lc,
	pss,
	peak_tol,
	detectability
	)
	values (
	:new.sample_id,
	:new.peak_id,
	:new.centroid,
	:new.centroid_err,
	:new.energy,
	:new.energy_err,
	:new.left_chan,
	:new.width,
	:new.back_count,
	:new.back_uncer,

```
   :new.fwhm,
   :new.fwhm_err,
   :new.area,
   :new.area_err,
   :new.original_area,
   :new.original_uncer,
   :new.counts_sec,
   :new.counts_sec_err,
   :new.efficiency,
   :new.eff_error,
   :new.back_channel,
   :new.ided,
   :new.fitted,
   :new.multiplet,
   :new.peak_sig,
   :new.lc,
   :new.pss,
   :new.peak_tol,
   :new.detectability );
ELSIF UPDATING THEN
 update rmsauto.gards_peaks_orig set
    sample_id = :new.sample_id ,
    peak_id = :new.peak_id,
    centroid = :new.centroid,
    centroid_err = :new.centroid_err,
    energy = :new.energy,
    energy_err = :new.energy_err,
    left_chan = :new.left_chan,
    width = :new.width,
    back_count = :new.back_count,
    back_uncer = :new.back_uncer,
    fwhm = :new.fwhm ,
    fwhm_err = :new.fwhm_err,
    area = :new.area,
    area_err = :new.area_err,
    original_area = :new.original_area,
    original_uncer = :new.original_uncer,
    counts_sec = :new.counts_sec ,
    counts_sec_err = :new.counts_sec_err ,
    efficiency = :new.efficiency,
    eff_error = :new.eff_error ,
    back_channel = :new.back_channel,
    ided = :new.ided,
    fitted = :new.fitted,
    multiplet = :new.multiplet,
    peak_sig = :new.peak_sig,
    lc = :new.lc,
    pss = :new.pss,
    peak_tol = :new.peak_tol,
    detectability = :new.detectability
 where sample_id= :old.sample_id and peak_id = :old.peak_id;
ELSE
 delete from rmsauto.gards_peaks_orig where sample_id
```

	`= :old.sample_id and peak_id`
	`= :old.peak_id;`
	`END IF;`
	`end;`

5.3.204. UPDATE_REFLINES_B_I_U

Table	GARDS_UPDATE_REFLINES
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.205. USERENV_B_I_U

Table	GARDS_USERENV
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.206. USERS_B_I_U

Table	GARDS_USERS
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.207. USERS_ROLES_B_I_U

Table	GARDS_USERS_ROLES
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.208. USER_COMMENTS_B_I_U

Table	GARDS_USER_COMMENTS
Type	BEFORE EACH ROW

Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.209. XE_NUCL_LIB_B_I_U

Table	GARDS_XE_NUCL_LIB
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.210. XE_NUCL_LINES_LIB_B_I_U

Table	GARDS_XE_NUCL_LINES_LIB
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.211. XE_PROC_PARAMS_TEMPLATE_B_I_U

Table	GARDS_XE_PROC_PARAMS_TEMPLATE
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.212. XE_REFLINE_MASTER_B_I_U

Table	GARDS_XE_REFLINE_MASTER
Type	BEFORE EACH ROW
Event	INSERT OR UPDATE

Trigger body	`begin`
	`:new.moddate := sysdate;`
	`end;`

5.3.213. XE_RESULTS_B_I_U

Table	GARDS_XE_RESULTS
Type	BEFORE EACH ROW

Event	INSERT OR UPDATE

Trigger body	`begin`
	`    :new.moddate := sysdate;`
	`end;`

5.4. Radionuclide Index Descriptions

Table name	Index name	Unique	Index columns
GARDS_ALERTS	ALERTS_UQ	Yes	station_id, dtg, alert_type, alert_id
	GA_DTG_NDX	No	dtg
	PK_ALERTS	Yes	alert_id
GARDS_AUTO_SAMPLE_CAT	PK_AUTOSAMPCAT	Yes	sample_id, name
	SAMP_CAT_NDX	No	sample_id
GARDS_AUX_LIB	PK_GARDS_AUX_LIB	Yes	name
GARDS_AUX_LINES_LIB	PK_AUX_LINS_LIB	Yes	name, energy, abundance_act
GARDS_BASELINE	PK_GARDS_BASELINE	Yes	detector_id, sample_type, data_type, baseline_type, index_no
GARDS_BG_CONFIG_PARAMS	PK_GARDS_BG_CONFIG_PARAMS	Yes	detector_id
GARDS_BG_DETECTOR_STD_SPECTRA	PK_GARDS_BG_DETECTOR_STD_SPEC	Yes	detector_id, sample_id
GARDS_BG_EFFICIENCY_PAIRS	BG_EFFICIENCY_PAIRS_NDX	No	sample_id
	PK_GARDS_BG_EFFICIENCY_PAIRS	Yes	sample_id, roi
GARDS_BG_ENERGY_CAL	PK_BG_EN_CAL	Yes	sample_id
GARDS_BG_ENERGY_CAL_DIRECTIVE	PK_BG_EN_CAL_DIRECTIVE	Yes	detector_id, ondate, cal_mode, sample_id
GARDS_BG_ENERGY_CAL_ORIG	PK_BG_EN_CAL_ORIG	Yes	sample_id
GARDS_BG_ISOTOPE_CONCS	BG_ISOTOPE_CONCS_NDX	No	sample_id
	PK_GARDS_BG_ISOTOPE_CONCS	Yes	sample_id, nuclide_id, method_id
GARDS_BG_PROC_PARAMS	BG_PROC_PARAMS_NDX	Yes	sample_id
GARDS_BG_PROC_PARAMS_ROI	BG_PROC_PARAMS_ROI_NDX	No	sample_id
	PK_GARDS_BG_PROC_PARAMS_ROI	Yes	sample_id, roi
GARDS_BG_QC_RESULT	BG_QC_RESULT_NDX	Yes	sample_id
GARDS_BG_ROI_CONCS	BG_ROI_CONCS_NDX	No	sample_id
	PK_GARDS_BG_ROI_CONCS	Yes	sample_id, roi, method_id

Table name	Index name	Unique	Index columns
GARDS_BG_ROI_COUNTS	BG_ROI_COUNTS_NDX	No	sample_id
	PK_GARDS_BG_ROI_COUNTS	Yes	sample_id, roi, method_id
GARDS_BG_STD_SPECTRA	BG_STD_SPECTRA_NDX	Yes	sample_id
GARDS_BG_STD_SPECTRA_RESULT	BG_STD_SPEC_RES_NDX	No	sample_id
	PK_GARDS_BG_STD_SPECTRA_RESULT	Yes	sample_id, std_spectra_id
GARDS_B_ENERGY_PAIRS	B_ENERGY_PAIRS_NDX	No	sample_id
	PK_GARDS_B_ENERGY_PAIRS	Yes	sample_id, cal_energy, type
GARDS_B_ENERGY_PAIRS_ORIG	B_ENERGY_PAIRS_ORIG_NDX	No	sample_id
	PK_GARDS_B_ENERGY_PAIRS_ORIG	Yes	sample_id, cal_energy, type
GARDS_B_RESOLUTION_CAL	PK_GARDS_B_RESOLUTION_CAL	Yes	sample_id, type
GARDS_B_RESOLUTION_PAIRS	B_RESOLUTION_PAIRS_NDX	No	sample_id
	PK_GARDS_B_RESOLUTION_PAIRS	Yes	sample_id, res_energy, type
GARDS_B_RESOLUTION_PAIRS_ORIG	B_RESOLUTION_PAIRS_ORIG_NDX	No	sample_id
	PK_GARDS_B_RESOLUTION_PAIRS_O	Yes	sample_id, res_energy, type
GARDS_CAT_CRITERIA_TESTS	PK_CAT_CRIT_TEST_ID	Yes	test_id
	UQ_CAT_CRIT_TEST_CODE	Yes	test_code
GARDS_CAT_TEMPLATE	CAT_TEMPLATE_DET_NDX	No	detector_id
	CAT_TEMPLATE_STA_NDX	No	station_id
	PK_GARDS_CAT_TEMPLATE	Yes	station_id, detector_id, name, begin_date
GARDS_CODES	PK_GARDS_CODES	Yes	type, code
GARDS_COMMENTS	COMMENTS_NDX	No	sample_id, analyst
	PK_COMMENTS	Yes	comment_id
GARDS_COMMENTS_DEFS	PK_COMMENTS_DEF	Yes	comment_type
GARDS_CSC_MODCOEFF_LIB	PK_CSC_MODCOEFF	Yes	detector_id, name, energy
GARDS_DATA_LOG	DATA_LOG_MOD_NDX	No	moddate
	DATA_LOG_NDX	No	station_id, data_type, status
	DATA_LOG_TYPE_DTG_NDX	No	data_type, dtg

Table name	Index name	Unique	Index columns
	GDL_MSG_ID_NDX	No	msg_id
	GDL_RMS_ID_NDX	No	rms_id
	PK_GARDS_DATA_LOG	Yes	dlid
GARDS_DETECTORS	PK_DETECTORS	Yes	detector_id
	UQ_DETECTORS	Yes	detector_code
GARDS_DIST_SAMPLE_QUEUE	PK_DIST_SAMPLE_QUEUE	Yes	sample_id
GARDS_EFFICIENCY_CAL	EFFICIENCY_CAL_NDX	Yes	sample_id
GARDS_EFFICIENCY_ERROR_CAL	EFFICIENCY_CAL_ERR_NDX	No	sample_id
	PK_GARDS_EFFICIENCY_ERROR_CAL	Yes	sample_id, type
GARDS_EFFICIENCY_IDC_PAIRS	EFFICIENCY_IDC_PAIRS_NDX	No	detector_id, station_id, begin_date
	PK_GARDS_EFF_IDC_PAIRS	Yes	detector_id, station_id, begin_date, effic_energy
GARDS_EFFICIENCY_PAIRS	EFFICIENCY_PAIRS_EFF_NDX	No	efficiency
	EFFICIENCY_PAIRS_EN_NDX	No	effic_energy
	EFFICIENCY_PAIRS_NDX	No	sample_id
	PK_GARDS_EFFICIENCY_PAIRS	Yes	sample_id, effic_energy
GARDS_EFFICIENCY_VGSL_PAIRS	EFFICIENCY_VGSL_PAIR_NDX	No	detector_id
	PK_GARDS_EFF_VGSL_PAIRS	Yes	detector_id, begin_date, end_date, effic_energy
GARDS_ENERGY_CAL	ENERGY_CAL_NDX	No	sample_id
	PK_GARDS_ENERGY_CAL	Yes	sample_id, type
GARDS_ENERGY_CAL_COV	ENERGY_CAL_COV_NDX	No	sample_id
	PK_GARDS_ENERGY_CAL_COV	Yes	sample_id, type, row_index, col_index
GARDS_ENERGY_CAL_ORIG	ENERGY_CAL_ORIG_NDX	Yes	sample_id
	PK_GARDS_ENERGY_CAL_ORIG	Yes	sample_id, type
GARDS_ENERGY_PAIRS	ENERGY_PAIRS_NDX	No	sample_id
	ENERGY_PAIRS_TYPE_NDX	No	sample_id, type
	PK_GARDS_ENERGY_PAIRS	Yes	sample_id, cal_energy, type
GARDS_ENERGY_PAIRS_	ENERGY_PAIRS_ORIG_NDX	No	sample_id

Table name	Index name	Unique	Index columns
ORIG	PK_GARDS_ENERGY_PAIRS_ORIG	Yes	sample_id, cal_energy, type
GARDS_ENVIRONMENT	ENVIRONMENT_NDX	No	sample_id
GARDS_FLAGS	SYS_C0028368	Yes	flag_id
GARDS_FPE	FPE_NDX	No	sample_id
	PK_GARDS_FPE	Yes	fpid
GARDS_HISTOGRAM	HISTOGRAM_NDX	Yes	sample_id
GARDS_INTERVAL	GINT_LDDATEX	No	lddate
	GINT_MODDATEX	No	moddate
	PK_GARDS_INTERVAL	Yes	intvlid
GARDS_IRF	IRF_NDX	No	detector_id
GARDS_LAB_CAT-EGORY_DESCRIPTION	PK_LAB_CATEGORY_DE-SCRIPTION	Yes	lab_sample_category
GARDS_MDAS2REPORT	PK_GARDS_MDAS2REPORT	Yes	name, sample_type
GARDS_MET_DATA	MET_DATA_NDX	No	station_id, start_time, end_time
	PK_GARDS_MET_DATA	Yes	met_id
GARDS_MET_HEADER	GMH_END_TIME_NDX	No	end_time
	GMH_START_TIME_NDX	No	start_time
	PK_GARDS_MET_HEADER	Yes	met_group_id
	UQ_MET_HEADER	Yes	station_id, start_time, end_time
GARDS_NIC	PK_GARDS_NIC	Yes	sample_id, name, sample_status
GARDS_NOTIFY	NOTIFY_NDX	No	poc_id
	PK_GARDS_NOTIFY	Yes	event, email_addr, dtg_begin
GARDS_NUC-L2QUANTIFY	PK_GARDS_NUC-L2QUANTIFY	Yes	name, dtg_begin
GARDS_NUCL_IDED	NUCL_IDED_NDX	No	sample_id, nuclide_id
	PK_NUCL_IDED	Yes	sample_id, name
GARDS_NUCL_IDED_ORIG	NUCL_IDED_ORIG_NDX	No	sample_id, nuclide_id
	PK_NUCL_IDED_ORIG	Yes	sample_id, name
GARDS_NUCL_LIB	PK_GARDS_NUCL_LIB	Yes	name
GARDS_NUCL_LIB_ARCHIVE	PK_ARCH_NUCL_LIB	Yes	name, begin_date
GARDS_NUCL_LINES_IDED	NUCL_LINES_IDED_NDX	No	sample_id, peak, name
	PK_GARDS_NUCL_LINES_IDED	No	sample_id, sample_id, name, name, nuclide_id, nuclide_id,

Table name	Index name	Unique	Index columns
			energy, energy, peak, peak, peak, peak
GARDS_NUCL_LINES_IDED_ORIG	NUCL_LINES_IDED_ORIG_NDX	No	sample_id, peak, name
	NUCL_LINES_IDED_ORIG_NDX2	No	sample_id, nuclide_id, peak
	PK_GARDS_NUCL_LINES_IDED_ORIG	Yes	sample_id, name, energy, peak
GARDS_NUCL_LINES_LIB	PK_GARDS_NUCL_LINES_LIB	Yes	name, energy
GARDS_NUCL_LINES_LIB_ARCHIVE	PK_ARCH_NUCL_LINES_LIB	Yes	name, energy, begin_date
GARDS_PEAKS	PEAKS_NDX	No	sample_id, sample_id, peak_id, peak_id, energy, energy, ided, ided
	PK_GARDS_PEAKS	Yes	sample_id, peak_id
GARDS_PEAKS_ORIG	PEAKS_ORIG_NDX	No	sample_id, peak_id, energy, ided
	PK_GARDS_PEAKS_ORIG	Yes	sample_id, peak_id
GARDS_PERMISSIONS	PK_PERMISSIONS	Yes	permission_id
GARDS_POC	PK_POC	Yes	pocid
GARDS_PROC_PARAMS_TEMPLATE	PK_GARDS_PROC_PARAMS_TEMPLATE	Yes	station_id, detector_id, data_type, spectral_qualifier
	PROC_PAR_TPL_DET_NDX	No	detector_id
	PROC_PAR_TPL_STA_NDX	No	station_id
GARDS_PRODUCT	GARDS_PRODUCT_PK	Yes	sample_id, typeid, revision
	PRODUCT_NDX	No	sample_id
GARDS_QCHISTORY	PK_GARDS_QCHISTORY	Yes	sample_id, state
	QCHISTORY_NDX	No	sample_id
GARDS_QCPARAMS	PK_GARDS_QCPARAMS	No	detector_id
GARDS_QCTARGETS	PK_GARDS_QCTARGETS	Yes	name, detector_id, station_id, energy
	QCTARGETS_DET_NDX	No	detector_id
	QCTARGETS_STA_NDX	No	station_id
GARDS_QC_RESULTS	GARDS_QC_RESULTS_PK	Yes	sample_id, test_name
	PK_GARDS_QC_RESULTS	Yes	sample_id, test_name
	QC_RESULTS_NDX	No	sample_id
	QC_RESULTS_SID_NDX	No	sample_id

Table name	Index name	Unique	Index columns
GARDS_RECEIPT_LOG	PK_GARDS_RECEIPT_LOG	Yes	pocid, subid, prodid
GARDS_REFLINE_MAS-TER	PK_GARDS_REFLINE_MAS-TER	Yes	refpeak_energy, data_type, spectral_qualifier
GARDS_RELEVANT_NUCLIDES	PK_GARDS_RELEVANT_NUCL	Yes	name, sample_type
GARDS_RESOLUTION_CAL	PK_GARDS_RESOLUTION_CAL	Yes	sample_id, type
	RESOLUTION_CAL_NDX	No	sample_id
GARDS_RESOLUTION_CAL_COV	PK_GARDS_RESOLUTION_CAL_COV	Yes	sample_id, type, row_index, col_index
	RESOLUTION_CAL_COV_SID_NDX	No	sample_id
GARDS_RESOLUTION_CAL_ORIG	PK_GARDS_RESOLUTION_CAL_ORIG	Yes	sample_id, type
	RESOLUTION_CAL_ORIG_NDX	Yes	sample_id
GARDS_RESOLUTION_PAIRS	PK_GARDS_RESOLUTION_PAIRS	Yes	sample_id, res_energy, type
	RESOLUTION_PAIRS_NDX	No	sample_id
	RESOLUTION_PAIRS_TYPE_NDX	No	sample_id, type
GARDS_RESOLUTION_PAIRS_ORIG	PK_GARDS_RESOLUTION_PAIRS_ORIG	Yes	sample_id, res_energy, type
	RESOLUTION_PAIRS_ORIG_NDX	No	sample_id
GARDS_RLR	RLRID_PK	Yes	rlr_id
GARDS_RLR_CONCLU-SIONS	PK_RLR_CONCLUSIONS	Yes	rlr_id
GARDS_RLR_OBJECTIVE	PK_RLR_OBJECTIVE	Yes	rlr_id
GARDS_RLR_RATIOS	PK_RLR_RATIOS	Yes	rlr_id, nuclide1, nuc-lide2, activity_ratio
	RLR_RATIOS_NDX	Yes	rlr_id
GARDS_RLR_RESULTS	PK_RLR_RESULTS	Yes	rlr_id, nuclide_name, activity
	RLR_RESULTS_NDX	No	rlr_id
GARDS_RLR_SSREB	PK_RLR_SSREB	Yes	rlr_id, sample_id
	RLR_SSREB_NDX	Yes	rlr_id
	RLR_SSREB_SID_NDX	No	sample_id
GARDS_RN_INTERVAL	GARDS_RN_INTERVAL_CNE_NDX	No	class, name, end_time

Table name	Index name	Unique	Index columns
	PK_RN_INTERVAL	Yes	intvlid
	RN_INTERVAL_MOD_NDX	No	moddate
	RN_INTERVAL_REF_NDX	No	refid, refidtype, state
GARDS_ROI_CHANNELS	PK_GARDS_ROI_CHANNELS	Yes	sample_id, roi
	ROI_CHANNELS_NDX	No	sample_id
GARDS_ROI_CONCS	PK_GARDS_ROI_CONCS	Yes	sample_id, roi
	ROI_CONCS_NDX	No	sample_id
GARDS_ROI_COUNTS	PK_GARDS_ROI_COUNTS	Yes	sample_id, roi
	ROI_COUNTS_NDX	No	sample_id
GARDS_ROI_LIB	PK_GARDS_ROI_LIB	Yes	roi
GARDS_ROI_LIMITS	PK_GARDS_ROI_LIMITS	Yes	sample_id, roi
	ROI_LIMITS_NDX	No	sample_id
GARDS_ROLES	PK_ROLES	Yes	role_id
GARDS_ROLES_PERMISSIONS	PK_GARDS_ROLES_PERM	Yes	role_id, permission_id
	ROLES_PERMISSIONS_NDX	No	role_id
	ROLES_PERM_NDX	No	permission_id
GARDS_SAINT_DEFAULT_PARAMS	PK_GARDS_SAINT_DEFAULT_PARAMS	Yes	name
GARDS_SAINT_PROCESS_PARAMS	PK_GARDS_SAINT_PROCESS_PARAMS	Yes	sample_id, name
	SAINT_PROC_PARAMS_NDX	No	sample_id
GARDS_SAMPLE_AUX	PK_SAMPLE_AUX	Yes	sample_id
	SAMPLE_AUX_MES_NDX	No	measurement_id, sample_id
	SAMPLE_AUX_MSG_ID_NDX	No	msg_id
	SAMPLE_AUX_REFID_NDX	No	sample_ref_id, sample_id
GARDS_SAMPLE_CAT	PK_GARDS_SAMPLE_CAT	Yes	sample_id, name, method_id
	SAMP_CAT_NDX	No	sample_id
GARDS_SAMPLE_CERT	SAMPLE_CERT_NDX	Yes	sample_id
GARDS_SAMPLE_CERT_LINES	PK_GARDS_SAMPLE_CERT_LINES	Yes	sample_id, nucl_name, energy, b_energy
	SAMPLE_CERT_LINES_NDX	No	sample_id
GARDS_SAMPLE_DATA	GARDS_SAMPLE_DATA_NDX	No	station_id, detector_id, spectral_qualifier, data_type, acquisition_start
	GSD_ACQUISITION_START_NDX	No	acquisition_start

Table name	Index name	Unique	Index columns
	GSD_ACQUISITION_STOP_NDX	No	acquisition_stop
	GSD_COLLECT_START_NDX	No	collect_start
	GSD_COLLECT_STOP_NDX	No	collect_stop
	GSD_DSX	No	data_type, spectral_qualifier
	NUQ2_SAMPLE_DATA	No	data_type, detector_id, spectral_qualifier, sample_type, acquisition_start, sample_id
	PK_SAMPLE_DATA	Yes	sample_id
	SAMPLE_DATA_NDX	No	station_id, detector_id, collect_stop, data_type, sample_type, spectral_qualifier
	UQ2_SAMPLE_DATA	Yes	data_type, detector_id, spectral_qualifier, sample_type, acquisition_start, sample_id
	UQ_SAMPLE_DATA	No	station_id, detector_id, spectral_qualifier, collect_stop, acquisition_stop, transmit_dtg
GARDS_SAMPLE_DESCRIPTION	PK_SAMPLE_DESCRIPTION	Yes	sample_id
GARDS_SAMPLE_FLAGS	PK_GARDS_SAMPLE_FLAGS	Yes	sample_id, flag_id
	SAMPLE_FLAGS_NDX	No	sample_id
GARDS_SAMPLE_PROC_PARAMS	PK_GARDS_SAMPLE_PROC_PARAMS	Yes	sample_id
	PK_GARDS_SAMPLE_PROC_PARS	Yes	sample_id
GARDS_SAMPLE_RATIOS	PK_GARDS_SAMPLE_RATIOS	Yes	sample_id, ratio_id
	SAMPLE_RATIOS_NDX	No	sample_id
GARDS_SAMPLE_STATUS	PK_SAMPLE_STATUS	Yes	sample_id
	SAMPLE_STATUS_NDX	Yes	status, sample_id
GARDS_SAMPLE_TESTS	PK_SAMPLE_TESTS	Yes	test_id
GARDS_SAMPLE_UPDATE_PARAMS	PK_GARDS_SAMPLE_UPD_PAR	No	sample_id
	PK_SAMPLE_UPDATE_PARAMS	Yes	sample_id
GARDS_SAMPLE_XE_PROC_PARAMS	PK_GARDS_SAMPLE_XE_PROC_PARAMS	Yes	sample_id, roi

Table name	Index name	Unique	Index columns
	SAMPLE_XE_PROC_PARAMS_SID_NDX	No	sample_id
GARDS_SOH_CHAR_DATA	FK_SOH_ID_CHAR_DATA	No	soh_id
	PK_CHAR_DATA	Yes	soh_char_id
	SOH_CHAR_DATA_NDX1	No	dtg_end
	SOH_CHAR_VALUE_NDX	Yes	station_id, param_code, value, dtg_begin, dtg_end
	UQ_SOH_CHAR_DATA	Yes	station_id, param_code, dtg_begin, dtg_end
GARDS_SOH_CODE	PK_SOH_CODE	Yes	param
GARDS_SOH_EVALUATION	PK_GSE	Yes	sample_id
	SYS_C002723457	Yes	sample_id
GARDS_SOH_HEADER	GSH_DTG_BEGIN_NDX	No	dtg_begin
	GSH_DTG_END_NDX	No	dtg_end
	PK_SOH_HEADER	Yes	soh_id
	UQ_SOH_HEADER	Yes	station_id, dtg_begin, dtg_end
GARDS_SOH_NUM_DATA	FK_SOH_ID_NUM_DATA	No	soh_id
	PK_NUM_DATA	Yes	soh_num_id
	SOH_NUM_DTG_BEGIN_NDX	No	dtg_begin
	SOH_NUM_DTG_END_NDX	No	dtg_end
	SOH_NUM_STAPAR_NDX	No	station_id, detector_id, param_code
	UQ_SOH_NUM_DATA	Yes	station_id, param_code, dtg_begin, dtg_end
GARDS_SOH_SENSOR_DATA	FK_SOH_ID_SENSOR_DATA	No	soh_id
	GSSD_DTG_ENDX	No	dtg_end
	PK_SENSOR_DATA	Yes	soh_sensor_id
	UQ_SENSOR_DATA	Yes	station_id, sensor_type, sensor_name, dtg_begin, dtg_end
GARDS_SPECTRUM	PK_GARDS_SPECTRUM	Yes	sample_id, sample_type
	SPECTRUM_NDX	No	sample_id
GARDS_STADET	GARDS_STADET_DID_NDX	No	detector_id
	PK_STADET	Yes	station_id, detector_id
GARDS_STATIONS	PK_STATIONS	Yes	station_id

Table name	Index name	Unique	Index columns
	UQ_STATIONS	Yes	station_code
GARDS_STATIONS_SCHEDULE	PK_GARDS_STATIONS_SCHEDULE	Yes	station_id, code, begin_date
	STATIONS_SCHEDULE_NDX	No	station_id
GARDS_STATION_ASSIGNMENTS	PK_GARDS_STATION_ASS	No	station_id
	STATION_ASSIGN_UID_NDX	No	user_id
GARDS_STATUS_HISTORY	STATUS_HISTORY_NDX	No	sample_id
GARDS_TOTAL_EFFIC	PK_GARDS_TOTAL_EFFIC	Yes	sample_id, effic_energy
	TOTAL_EFFIC_NDX	No	sample_id
GARDS_UPDATE_PARAMS_TEMPLATE	PK_GARDS_UPDATE_PAR_TEMPL	Yes	detector_id
GARDS_UPDATE_REFLINES	PK_GARDS_UPDATE_REFLINES	Yes	station_id, detector_id, refpeak_energy, data_type, spectral_qualifier
	UPDATE_REFLINES_DET_NDX	No	detector_id
	UPDATE_REFLINES_NDX	No	station_id
GARDS_USERENV	PK_GARDS_USERENV	Yes	name
GARDS_USERS	PK_USERS	Yes	user_id
GARDS_USERS_ROLES	PK_GARDS_USERS_ROLES	Yes	user_id, role_id
	USERS_ROLES_ROLE_NDX	No	role_id
	USERS_ROLES_UID_NDX	No	user_id
GARDS_USER_COMMENTS	PK_GARDS_USER_COMMENTS	Yes	comment_id
	USER_COMMENTS_SID_NDX	No	sample_id
GARDS_XE_CATEGORY_PARAMS	GARDS_XE_CAT_PAR_PK	Yes	start_date
GARDS_XE_NUCL_LIB	PK_GARDS_XE_NUCL_LIB	Yes	name
GARDS_XE_NUCL_LINES_LIB	PK_GARDS_XE_NUCL_LINES_LIB	Yes	name, energy
GARDS_XE_PROC_PARAMS_TEMPLATE	PK_GARDS_XE_PROC_PARAMS_TEMP	Yes	detector_id, roi
	XE_PROC_PARAMS_TEMPL_NDX	No	detector_id
GARDS_XE_REFLINE_MASTER	PK_GARDS_XE_REFLINE_MASTER	Yes	refpeak_energy, data_type, spectral_qualifier
GARDS_XE_RESULTS	PK_GARDS_XE_RESULTS	Yes	sample_id, method_id,

Table name	Index name	Unique	Index columns
			nuclide_id
	XE_RESULTS_NDX	No	sample_id
GARDS_XE_SAMPLE_CATEGORY	PK_SAMPLE_ID	Yes	sample_id, flag_shortterm
GARDS_XE_UNCORRECTED_RESULTS	PK_XE_UNCORRECTED_RESULTS	Yes	sample_id, method_id, nuclide_id
RMS_MSGDISC	PK_RMS_MSGDISC	Yes	msgid
	RMS_MSGDISC_LDDATE_NDX	No	lddate
	RMS_MSGDISC_MOD_NDX	No	moddate
	RMS_MSGDISC_NDX	Yes	status, msgid
RN_LOGS	PK_RN_LOGS	Yes	comment_id
RN_LOGS_TXT	PK_RN_LOGS_TXT	Yes	comment_id

5.5. Radionuclide Synonym Descriptions

Schema	Synonym	Table	Table Owner
RMSAUTO	FILEPRODUCT	FILEPRODUCT	IDCX
	FPDESCRIPTION	FPDESCRIPTION	IDCX
	GARDS_AUX_LIB	GARDS_AUX_LIB	RMSMAN
	GARDS_AUX_LINES_LIB	GARDS_AUX_LINES_LIB	RMSMAN
	GARDS_BG_CONFIG_PARAMS	GARDS_BG_CONFIG_PARAMS	RMSMAN
	GARDS_BG_ENERGY_CAL_DIRECTIVE	GARDS_BG_ENERGY_CAL_DIRECTIVE	RMSMAN
	GARDS_CAT_CRITERIA_TESTS	GARDS_CAT_CRITERIA_TESTS	RMSMAN
	GARDS_CAT_TEMPLATE	GARDS_CAT_TEMPLATE	RMSMAN
	GARDS_CODES	GARDS_CODES	RMSMAN
	GARDS_COMMENTS_DEFS	GARDS_COMMENTS_DEFS	RMSMAN
	GARDS_CSC_MODCOEFF_LIB	GARDS_CSC_MODCOEFF_LIB	RMSMAN
	GARDS_DBROLE_OWNER	GARDS_DBROLE_OWNER	RMSMAN
	GARDS_DETECTORS	GARDS_DETECTORS	RMSMAN
	GARDS_DIST_SAMPLE_QUEUE	GARDS_DIST_SAMPLE_QUEUE	RMSMAN
	GARDS_EFFICIENCY_VGSL_PAIRS	GARDS_EFFICIENCY_VGSL_PAIRS	RMSMAN
	GARDS_IRF	GARDS_IRF	RMSMAN
	GARDS_LAB_CATEGORY_DESCRIPTION	GARDS_LAB_CATEGORY_DESCRIPTION	RMSMAN
	GARDS_MDAS2REPORT	GARDS_MDAS2REPORT	RMSMAN
	GARDS_NUCL2QUANTIFY	GARDS_NUCL2QUANTIFY	RMSMAN
	GARDS_NUCL_LIB	GARDS_NUCL_LIB	RMSMAN
	GARDS_NUCL_LINES_LIB	GARDS_NUCL_LINES_LIB	RMSMAN
	GARDS_PERMISSIONS	GARDS_PERMISSIONS	RMSMAN
	GARDS_POC	GARDS_POC	RMSMAN
	GARDS_PROCESSING_ERRORS	GARDS_PROCESSING_ERRORS	RMSMAN
	GARDS_PROC_PARAMS_TEMPLATE	GARDS_PROC_PARAMS_TEMPLATE	RMSMAN
	GARDS_QAT_CONFIG	GARDS_QAT_CONFIG	RMSMAN
	GARDS_QAT_NOTIFY	GARDS_QAT_NOTIFY	RMSMAN
	GARDS_QAT_QUERY_FILTER	GARDS_QAT_QUERY_FILTER	RMSMAN

Schema	Synonym	Table	Table Owner
	GARDS_QCPARAMS	GARDS_QCPARAMS	RMSMAN
	GARDS_QCTARGETS	GARDS_QCTARGETS	RMSMAN
	GARDS_READ_SAMPLE_CAT	GARDS_SAMPLE_CAT	RMSMAN
	GARDS_READ_SAMPLE_STATUS	GARDS_SAMPLE_STATUS	RMSMAN
	GARDS_REFLINE_MASTER	GARDS_REFLINE_MASTER	RMSMAN
	GARDS_RELEVANT_NUCLIDES	GARDS_RELEVANT_NUCLIDES	RMSMAN
	GARDS_ROI_LIB	GARDS_ROI_LIB	RMSMAN
	GARDS_ROLES	GARDS_ROLES	RMSMAN
	GARDS_ROLES_PERMISSIONS	GARDS_ROLES_PERMISSIONS	RMSMAN
	GARDS_SAMPLE_CAT	GARDS_AUTO_SAMPLE_CAT	RMSAUTO
	GARDS_SAMPLE_TESTS	GARDS_SAMPLE_TESTS	RMSMAN
	GARDS_SOH_CODE	GARDS_SOH_CODE	RMSMAN
	GARDS_STADET	GARDS_STADET	RMSMAN
	GARDS_STATIONS	GARDS_STATIONS	RMSMAN
	GARDS_STATIONS_SCHEDULE	GARDS_STATIONS_SCHEDULE	RMSMAN
	GARDS_STATION_ASSIGNMENTS	GARDS_STATION_ASSIGNMENTS	RMSMAN
	GARDS_STATUS_HISTORY	GARDS_STATUS_HISTORY	RMSMAN
	GARDS_UPDATE_PARAMS_TEMPLATE	GARDS_UPDATE_PARAMS_TEMPLATE	RMSMAN
	GARDS_UPDATE_REFLINES	GARDS_UPDATE_REFLINES	RMSMAN
	GARDS_USERENV	GARDS_USERENV	RMSMAN
	GARDS_USERS	GARDS_USERS	RMSMAN
	GARDS_USERS_ROLES	GARDS_USERS_ROLES	RMSMAN
	GARDS_WRITE_SAMPLE_CAT	GARDS_AUTO_SAMPLE_CAT	RMSAUTO
	GARDS_XE_CATEGORY_PARAMS	GARDS_XE_CATEGORY_PARAMS	RMSMAN
	GARDS_XE_NUCL_LIB	GARDS_XE_NUCL_LIB	RMSMAN
	GARDS_XE_NUCL_LINES_LIB	GARDS_XE_NUCL_LINES_LIB	RMSMAN
	GARDS_XE_PROC_PARAMS_TEMPLATE	GARDS_XE_PROC_PARAMS_TEMPLATE	RMSMAN
	GARDS_XE_REFLINE_MASTER	GARDS_XE_REFLINE_MASTER	RMSMAN

Schema	Synonym	Table	Table Owner
	GBEPO	GARDS_B_ENERGY_PAIRS_ORIG	RMSAUTO
	GBGEP	GARDS_BG_EFFICIENCY_PAIRS	RMSAUTO
	GBRP	GARDS_B_RESOLUTION_PAIRS	RMSAUTO
	GBRPO	GARDS_B_RESOLUTION_PAIRS_ORIG	RMSAUTO
	GEPO	GARDS_ENERGY_PAIRS_ORIG	RMSAUTO
	GNLIO	GARDS_NUCL_LINES_IDED_ORIG	RMSAUTO
	GRCO	GARDS_RESOLUTION_CAL_ORIG	RMSAUTO
	GRPO	GARDS_RESOLUTION_PAIRS_ORIG	RMSAUTO
	GSCL	GARDS_SAMPLE_CERT_LINES	RMSAUTO
	GSD	GARDS_SAMPLE_DATA	RMSAUTO
	GSDESC	GARDS_SAMPLE_DESCRIP-TION	RMSAUTO
	GSPP	GARDS_SAMPLE_PROC_PARAMS	RMSAUTO
	GSS	GARDS_SAMPLE_STATUS	RMSAUTO
	GSUP	GARDS_SAMPLE_UPDATE_PARAMS	RMSAUTO
	GSXPP	GARDS_SAMPLE_XE_PROC_PARAMS	RMSAUTO
	INTERVAL	GARDS_INTERVAL	RMSAUTO
	LASTID	LASTID	IDCX
	MSGDISC	MSGDISC	IDCX
	PIPELINE_CONTROL	PIPELINE_CONTROL	RMSMAN
RMSMAN	FILEPRODUCT	FILEPRODUCT	IDCX
	FPDESCRIPTION	FPDESCRIPTION	IDCX
	GARDS_ALERTS	GARDS_ALERTS	RMSAUTO
	GARDS_BG_DETECTOR_STD_SPECTRA	GARDS_BG_DETECTOR_STD_SPECTRA	RMSAUTO
	GARDS_BG_STD_SPECTRA	GARDS_BG_STD_SPECTRA	RMSAUTO
	GARDS_DATA_LOG	GARDS_DATA_LOG	RMSAUTO
	GARDS_ENVIRONMENT	GARDS_ENVIRONMENT	RMSAUTO
	GARDS_HISTOGRAM	GARDS_HISTOGRAM	RMSAUTO

Schema	Synonym	Table	Table Owner
	GARDS_INTERVAL	GARDS_INTERVAL	RMSAUTO
	GARDS_MET_DATA	GARDS_MET_DATA	RMSAUTO
	GARDS_MET_HEADER	GARDS_MET_HEADER	RMSAUTO
	GARDS_READ_SAMPLE_CAT	GARDS_SAMPLE_CAT	RMSMAN
	GARDS_READ_SAMPLE_STATUS	GARDS_SAMPLE_STATUS	RMSMAN
	GARDS_RECEIPT_LOG	GARDS_RECEIPT_LOG	RMSAUTO
	GARDS_RLR	GARDS_RLR	RMSAUTO
	GARDS_RLR_CONCLUSIONS	GARDS_RLR_CONCLUSIONS	RMSAUTO
	GARDS_RLR_OBJECTIVE	GARDS_RLR_OBJECTIVE	RMSAUTO
	GARDS_RLR_RATIOS	GARDS_RLR_RATIOS	RMSAUTO
	GARDS_RLR_RESULTS	GARDS_RLR_RESULTS	RMSAUTO
	GARDS_RLR_SSREB	GARDS_RLR_SSREB	RMSAUTO
	GARDS_RN_INTERVAL	GARDS_RN_INTERVAL	RMSAUTO
	GARDS_SAMPLE_AUX	GARDS_SAMPLE_AUX	RMSAUTO
	GARDS_SAMPLE_CERT	GARDS_SAMPLE_CERT	RMSAUTO
	GARDS_SAMPLE_CERT_LINES	GARDS_SAMPLE_CERT_LINES	RMSAUTO
	GARDS_SAMPLE_DESCRIPTION	GARDS_SAMPLE_DESCRIPTION	RMSAUTO
	GARDS_SOH_CHAR_DATA	GARDS_SOH_CHAR_DATA	RMSAUTO
	GARDS_SOH_HEADER	GARDS_SOH_HEADER	RMSAUTO
	GARDS_SOH_NUM_DATA	GARDS_SOH_NUM_DATA	RMSAUTO
	GARDS_SOH_SENSOR_DATA	GARDS_SOH_SENSOR_DATA	RMSAUTO
	GARDS_SPECTRUM	GARDS_SPECTRUM	RMSAUTO
	GARDS_WRITE_SAMPLE_CAT	GARDS_SAMPLE_CAT	RMSMAN
	GBEPO	GARDS_B_ENERGY_PAIRS_ORIG	RMSMAN
	GBGEP	GARDS_BG_EFFICIENCY_PAIRS	RMSMAN
	GBRP	GARDS_B_RESOLUTION_PAIRS	RMSMAN
	GBRPO	GARDS_B_RESOLUTION_PAIRS_ORIG	RMSMAN
	GCCT	GARDS_CAT_CRITERIA_TESTS	RMSMAN
	GNLIO	GARDS_NUCL_LINES_IDED	RMSMAN

Schema	Synonym	Table	Table Owner
		_ORIG	
	GPPT	GARDS_PROC_PARAMS_TEMPLATE	RMSMAN
	GRCO	GARDS_RESOLUTION_CAL_ORIG	RMSMAN
	GRPO	GARDS_RESOLUTION_PAIRS_ORIG	RMSMAN
	GSD	GARDS_SAMPLE_DATA	RMSMAN
	GSPP	GARDS_SAMPLE_PROC_PARAMS	RMSMAN
	GSS	GARDS_SAMPLE_STATUS	RMSMAN
	GSTA	GARDS_STATION_ASSIGN-MENTS	RMSMAN
	GSUP	GARDS_SAMPLE_UPDATE_PARAMS	RMSMAN
	GSXPP	GARDS_SAMPLE_XE_PROC_PARAMS	RMSMAN
	GUPT	GARDS_UPDATE_PARAMS_TEMPLATE	RMSMAN
	GXPPT	GARDS_XE_PROC_PARAMS_TEMPLATE	RMSMAN
	LASTID	LASTID	IDCX
	MSGDISC	MSGDISC	IDCX

5.6. Radionuclide Foreign Key Descriptions

Table name	Constraint name	Constrained column	Foreign table
GARDS_AUTO_SAMPLE_CAT	FK_SID_AUTOSAMPCAT	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_CONFIG_PARAMS	FK_BG_CONFIG_PARAMS	detector_id	GARDS_DETECTORS
GARDS_BG_EFFICIENCY_PAIRS	FK_BG_EFFICIENCY_PAIRS	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_ENERGY_CAL	FK_BG_EN_CAL_SID	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_ENERGY_CAL_DIRECTIVE	FK_BG_EN_CAL_DIR_DETID	detector_id	GARDS_DETECTORS
GARDS_BG_ENERGY_CAL_ORIG	FK_BG_EN_CAL_ORIG_SID	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_ISOTOPE_CONCS	FK_BG_ISOTOPE_CONCS	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_PROC_PARAMS	FK_BG_PROC_PARAMS	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_PROC_PARAMS_ROI	FK_BG_PROC_PARAMS_ROI	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_QC_RESULT	FK_BG_QC_RESULT	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_ROI_CONCS	FK_BG_ROI_CONCS	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_ROI_COUNTS	FK_BG_ROI_COUNTS	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_STD_SPECTRA	FK_BG_STD_SPECTRA	sample_id	GARDS_SAMPLE_DATA
GARDS_BG_STD_SPECTRA_RESULT	FK_BG_STD_SPECTRA_RESULT	sample_id	GARDS_SAMPLE_DATA
GARDS_B_ENERGY_PAIRS	FK_B_ENERGY_PAIRS	sample_id	GARDS_SAMPLE_DATA
GARDS_B_ENERGY_PAIRS_ORIG	FK_B_ENERGY_PAIRS_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_B_RESOLUTION_CAL	FK_GARDS_B_RESOLUTION_CAL	sample_id	GARDS_SAMPLE_DATA
GARDS_B_RESOLUTION_PAIRS	FK_B_RESOLUTION_PAIRS	sample_id	GARDS_SAMPLE_DATA
GARDS_B_RESOLUTION_PAIRS_ORIG	FK_B_RESOLUTION_PAIRS_ORIG	sample_id	GARDS_SAMPLE_DATA

Table name	Constraint name	Con-strained column	Foreign table
GARDS_CAT_TEM-PLATE	FK_DETECTOR	detector_id	GARDS_DETECTORS
GARDS_CAT_TEM-PLATE	FK_STATION	station_id	GARDS_STATIONS
GARDS_COMMENTS	FK_COMMENTS	sample_id	GARDS_SAMPLE_DATA
GARDS_CSC_MOD-COEFF_LIB	FK_CSC_MODCFF_LIB	detector_id	GARDS_DETECTORS
GARDS_DIST_SAMPLE_QUEUE	FK_DIST_SAMPLE_QUEUE	sample_id	GARDS_SAMPLE_DATA
GARDS_EFFI-CIENCY_CAL	FK_EFFICIENCY_CAL	sample_id	GARDS_SAMPLE_DATA
GARDS_EFFI-CIENCY_ERROR_CAL	FK_EFFICIENCY_ERROR_CAL	sample_id	GARDS_SAMPLE_DATA
GARDS_EFFI-CIENCY_PAIRS	FK_EFFICIENCY_PAIRS	sample_id	GARDS_SAMPLE_DATA
GARDS_EFFI-CIENCY_VGSL_PAIRS	FK_EFFICIENCY_VGSL_PAIRS	detector_id	GARDS_DETECTORS
GARDS_ENERGY_CAL	FK_ENERGY_CAL	sample_id	GARDS_SAMPLE_DATA
GARDS_ENERGY_CAL_COV	FK_ENERGY_CAL_COV	sample_id	GARDS_SAMPLE_DATA
GARDS_ENERGY_CAL_ORIG	FK_ENERGY_CAL_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_ENERGY_PAIRS	FK_ENERGY_PAIRS	sample_id	GARDS_SAMPLE_DATA
GARDS_ENERGY_PAIRS_ORIG	FK_ENERGY_PAIRS_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_ENVIRON-MENT	FK_ENVIRONMENT	sample_id	GARDS_SAMPLE_DATA
GARDS_FPE	FK_FPE	sample_id	GARDS_SAMPLE_DATA
GARDS_HISTO-GRAM	FK_HISTOGRAM	sample_id	GARDS_SAMPLE_DATA
GARDS_IRF	FK_IRF	detector_id	GARDS_DETECTORS
GARDS_MET_DATA	FK_MET_DATA	station_id	GARDS_STATIONS
GARDS_MET_DATA	FK_MET_GROUP_ID_MET_HEADER	met_group_id	GARDS_MET_HEADER
GARDS_MET_HEADER	FK_MET_HEADER	station_id	GARDS_STATIONS

Table name	Constraint name	Con-strained column	Foreign table
GARDS_NOTIFY	FK_NOTIFY_POC	poc_id	GARDS_POC
GARDS_NUCL_IDED	FK_NUCL_IDED	sample_id	GARDS_SAMPLE_DATA
GARDS_NUCL_IDED_ORIG	FK_NUCL_IDED_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_NUCL_IDED_ORIG	FK_SID_NUCLIDEID	name	GARDS_NUCL_IDED
GARDS_NUCL_IDED_ORIG	FK_SID_NUCLIDEID	sample_id	GARDS_NUCL_IDED
GARDS_NUCL_LINES_IDED	FK_NUCL_LINES_IDED	sample_id	GARDS_SAMPLE_DATA
GARDS_NUCL_LINES_IDED_ORIG	FK_NUCL_LINES_IDED_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_NUCL_LINES_IDED_ORIG	FK_SID_ET_AL	nuclide_id	GARDS_NUCL_LINES_IDED
GARDS_NUCL_LINES_IDED_ORIG	FK_SID_ET_AL	peak	GARDS_NUCL_LINES_IDED
GARDS_NUCL_LINES_IDED_ORIG	FK_SID_ET_AL	sample_id	GARDS_NUCL_LINES_IDED
GARDS_PEAKS	FK_PEAKS	sample_id	GARDS_SAMPLE_DATA
GARDS_PEAKS_ORIG	FK_PEAKS_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_PEAKS_ORIG	FK_SID_PEAKID	peak_id	GARDS_PEAKS
GARDS_PEAKS_ORIG	FK_SID_PEAKID	sample_id	GARDS_PEAKS
GARDS_PROC_PARAMS_TEMPLATE	FK_PPTEMPL_DET	detector_id	GARDS_DETECTORS
GARDS_PROC_PARAMS_TEMPLATE	FK_PPTEMPL_STA	station_id	GARDS_STATIONS
GARDS_PRODUCT	FK_PRODUCT	sample_id	GARDS_SAMPLE_DATA
GARDS_QCHISTORY	FK_QCHISTORY	sample_id	GARDS_SAMPLE_DATA
GARDS_QCPARAMS	FK_QCPARAMS_DET	detector_id	GARDS_DETECTORS
GARDS_QCTARGETS	FK_QCTARGS_DET	detector_id	GARDS_DETECTORS
GARDS_QCTARGETS	FK_QCTARGS_STA	station_id	GARDS_STATIONS
GARDS_QC_RESULTS	FK_QC_RESULTS	sample_id	GARDS_SAMPLE_DATA
GARDS_RESOLU-	FK_RESOLUTION_CAL	sample_id	GARDS_SAMPLE_DA

Table name	Constraint name	Constrained column	Foreign table
TION_CAL			TA
GARDS_RESOLUTION_CAL_COV	FK_RESOLUTION_CAL_COV	sample_id	GARDS_SAMPLE_DATA
GARDS_RESOLUTION_CAL_ORIG	FK_RESOLUTION_CAL_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_RESOLUTION_PAIRS	FK_RESOLUTION_PAIRS	sample_id	GARDS_SAMPLE_DATA
GARDS_RESOLUTION_PAIRS_ORIG	FK_RESOLUTION_PAIRS_ORIG	sample_id	GARDS_SAMPLE_DATA
GARDS_RLR_CONCLUSIONS	SYS_C0028565	rlr_id	GARDS_RLR
GARDS_RLR_OBJECTIVE	SYS_C0028566	rlr_id	GARDS_RLR
GARDS_RLR_RATIOS	SYS_C0028567	rlr_id	GARDS_RLR
GARDS_RLR_RESULTS	SYS_C0028568	rlr_id	GARDS_RLR
GARDS_RLR_SSREB	SYS_C0028569	rlr_id	GARDS_RLR
GARDS_RLR_SSREB	SYS_C0028570	sample_id	GARDS_SAMPLE_DATA
GARDS_ROI_CHANNELS	FK_ROI_CHANNELS	sample_id	GARDS_SAMPLE_DATA
GARDS_ROI_CONCS	FK_ROI_CONCS	sample_id	GARDS_SAMPLE_DATA
GARDS_ROI_COUNTS	FK_ROI_COUNTS	sample_id	GARDS_SAMPLE_DATA
GARDS_ROI_LIMITS	FK_ROI_LIMITS	sample_id	GARDS_SAMPLE_DATA
GARDS_ROLES_PERMISSIONS	SYS_C0028509	permission_id	GARDS_PERMISSIONS
GARDS_ROLES_PERMISSIONS	SYS_C0028510	role_id	GARDS_ROLES
GARDS_SAINT_PROCESS_PARAMS	FK_SAINT_PROC_PARAMS	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_AUX	FK_SAMPLE_AUX	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_CAT	FK_SAMP_ID	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_CERT	FK_SAMPLE_CERT	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_CE	FK_SAMPLE_CERT_LINES	sample_id	GARDS_SAMPLE_DA

Table name	Constraint name	Con-strained column	Foreign table
RT_LINES			TA
GARDS_SAMPLE_DESCRIPTION	FK_SAMPLE_DESCRIPTION	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_FLAGS	FK_SAMPLE_FLAGS	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_PROC_PARAMS	FK_SAMPLE_PROC_PARAMS	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_RATIOS	FK_SAMPLE_RATIOS	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_STATUS	FK_SAMPLE_STATUS	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_UPDATE_PARAMS	FK_SAMPLE_UPDATE_PARAMS	sample_id	GARDS_SAMPLE_DATA
GARDS_SAMPLE_XE_PROC_PARAMS	FK_SAMPLE_XE_PROC_PARAMS	sample_id	GARDS_SAMPLE_DATA
GARDS_SOH_CHAR_DATA	FK_SOH_ID_CHAR_DATA	soh_id	GARDS_SOH_HEADER
GARDS_SOH_EVAL-UATION	FK_BGE_SID	sample_id	GARDS_SAMPLE_DATA
GARDS_SOH_EVAL-UATION	FK_GSE_SID	sample_id	GARDS_SAMPLE_DATA
GARDS_SOH_HEADER	FK_SOH_HEADER	station_id	GARDS_STATIONS
GARDS_SOH_NUM_DATA	FK_SOH_ID_NUM_DATA	soh_id	GARDS_SOH_HEADER
GARDS_SOH_SENSOR_DATA	FK_SOH_ID_SENSOR_DATA	soh_id	GARDS_SOH_HEADER
GARDS_SPECTRUM	FK_SPECTRUM	sample_id	GARDS_SAMPLE_DATA
GARDS_STADET	FK_STADET_DET	detector_id	GARDS_DETECTORS
GARDS_STADET	FK_STADET_STA	station_id	GARDS_STATIONS
GARDS_STATIONS_SCHEDULE	FK_STA_SCHEDULE_STAID	station_id	GARDS_STATIONS
GARDS_STATION_ASSIGNMENTS	FK_STA_ASSIGN_STAID	station_id	GARDS_STATIONS
GARDS_STATION_ASSIGNMENTS	FK_STA_ASSIGN_USRID	user_id	GARDS_USERS
GARDS_STATUS_HISTORY	FK_STATUS_HISTORY	sample_id	GARDS_SAMPLE_DATA
GARDS_TOTAL_EFFIC	FK_TOTAL_EFFIC	sample_id	GARDS_SAMPLE_DATA

Table name	Constraint name	Constrained column	Foreign table
GARDS_UPDATE_PARAMS_TEMPLATE	FK_UPD_PARAMS_TMPL	detector_id	GARDS_DETECTORS
GARDS_UPDATE_REFLINES	FK_UPD_REFLINES_DET	detector_id	GARDS_DETECTORS
GARDS_UPDATE_REFLINES	FK_UPD_REFLINES_STA	station_id	GARDS_STATIONS
GARDS_USERS_ROLES	FK_ROLE_ID	role_id	GARDS_ROLES
GARDS_USERS_ROLES	FK_USER_ID	user_id	GARDS_USERS
GARDS_USER_COMMENTS	FK_USER_COMMENTS	comment_id	GARDS_COMMENTS
GARDS_USER_COMMENTS	FK_USER_COMMENTS_SID	sample_id	GARDS_SAMPLE_DATA
GARDS_XE_PROC_PARAMS_TEMPLATE	XE_PPT_DET	detector_id	GARDS_DETECTORS
GARDS_XE_RESULTS	FK_XE_RESULTS	sample_id	GARDS_SAMPLE_DATA
GARDS_XE_SAMPLE_CATEGORY	FK_XESAMPLECAT	flag_shortterm	GARDS_XE_SAMPLE_CATEGORY
GARDS_XE_SAMPLE_CATEGORY	FK_XESAMPLECAT	sample_id	GARDS_XE_SAMPLE_CATEGORY
GARDS_XE_UNCORRECTED_RESULTS	FK_XE_UNCORRECTED_RESULTS	sample_id	GARDS_SAMPLE_DATA

6. FUNCTIONS, PROCEDURES AND PACKAGES

This section describes PL/SQL functions, procedures and packages that are defined in or used by objects in the S/H/I and radionuclide parts of the IDC database.

6.1. Functions

No functions found in the database.

6.2. Procedures

No procedures found in the database.

6.3. Packages

No packages found in the database.

REFERENCES

Anderson, J., Farrell, W., Garcia, K., Given, J., and Swanger, H. (1990), *CSS Version 3 Database: Schema Reference Manual*, Science Applications International Corporation.

Carter, J., and Bowman, J. R. (1997), *IDC Database Schema*, CMR-97/28.

Flinn, E. A., Engdhal, E. R., and Hill, A. R. (1974), "Seismic and Geographical Regionalization," *Bulletin of the Seismological Society of America*, Volume 64, No. 3, Part ii, pp. 771-992.

Gane, C., and Sarson, T. (1979), *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

IDC-5.1.1 (1998), *Database Schema (Part 1 and Part 2)*, Science Applications International Corporation, Pacific-Sierra Research, Inc. SAIC-98/3009, PSR-98/TN1127.

IDC-5.1.1.Rev1 (1999), *Database Schema (Parts 1, 2, and 3)*, Revision 1, Science Applications International Corporation, Pacific-Sierra Research Corporation, SAIC-99/3009, PSR-99/TN1142.

IDC-5.1.1.Rev5 (Oct. 2002), *Database Schema, (Part 1, Part 2, and Part 3),* Revision 5, Science Applications International Corporation, Veridian Systems, TN2924.

IDC-5.1.2 (1999), *Database Tutorial*, Science Applications International Corporation, Pacific-Sierra Research Corporation, SAIC-99/3022, PSR-99/TN1145.

IDC-5.1.3.Rev.0.1 (2001), *Configuration of PIDC Databases*, Rev.0.1, Science Applications International Corporation, Veridian Pacific- Sierra Research, SAIC-01/3022, PSR-99/TN1114.

IDC-7.1.1 (2001), *Detection and Feature Extraction (DFX)–Scheme Files*, Science Applications International Corporation, SAIC-01/3000.

IDC-7.1.3 (2001), *Surface Wave Identification and Measurement*, Science Applications International Corporation, SAIC-01/3008.

IDC-7.1.10.Rev.3 (2002), *Radionuclide Software Design*, Revision 3, Veridian Systems, TN2927.

IDC-7.4.2 (1998), *Message Subsystem*, Science Applications International Corporation, SAIC-98/3003.

IDC-7.4.3 (2001), *Retrieve Subsystem*, Science Applications International Corporation, SAIC-01/3024.

IDC-7.4.4 (1998), *Subscription Subsystem*, Science Applications International Corporation, SAIC-98/3001.

IDC-ENG-SPC-100.Rev.1 (2002), *Formats and Protocols for Continuous Data CD-1.0*.

Provisional Technical Secretariat of the CTBTO Preparatory Commission.

IDC-ENG-SPC-101.Rev.3 (2002), *Formats and Protocols for Continuous Data CD-1.1*. Provisional Technical Secretariat of the CTBTO Preparatory Commission.

IDC-ENG-SPC-401.Rev.0 (2016), *IDC DACS Subsystem Software Design*. Provisional Technical Secretariat of the CTBTO Preparatory Commission.

IDC-OPS-MAN-001.Rev1 (2002), *IDC Processing of Seismic, Hydroacoustic, and Infrasonic Data*. Provisional Technical Secretariat of the CTBTO Preparatory Commission.

IDC-OPS-MAN-050.Rev3 (2002), *IDC Processing of Radionuclide Data*. Provisional Technical Secretariat of the CTBTO Preparatory Commission.

IDC-OPS-MAN-103 (2001), *Archiving Subsystem*, Science Applications International Corporation, SAIC-01/3013.

IDC-OPS-MAN-201 (2011), *Station Processing (StaPro)*, Science Applications International Corporation, SAIC-00/3013.

IDC-OPS-MAN-202 (2001), *Global Association (GA) Subsystem*, Science Applications International Corporation, SAIC-01/3009.

IDC-OPS-MAN-203 (2001), *Event Location Software*, Science Applications International Corporation, SAIC-01/3010.

IDC-OPS-MAN-204 (2011), *Event Magnitude Software*, Science Applications International Corporation, SAIC-01/3011.

IDC-OPS-MAN-206 (2000), *WaveExpert*, Science Applications International Corporation, SAIC- 00/3029.

Swanger, H., Given, J., and Anderson, J. (1991), *IMS Extensions to the Center Version 3 Database*, Science Applications International Corporation, SAIC-91/1138.

Swanger, H., Anderson, J., Sereno, T., Given, J., and Williams, D. (1993), *Extensions to the Center Version 3 Database*, Revision 1, Science Applications International Corporation, SAIC-93/1123.

GLOSSARY

2-D	Two-dimensional.
3-C	Three-components.
abundance	Fraction of a decay event that results in the radiation(s) or interest (for example, a gamma line at a specific energy or a beta-gamma coincidence pair). Intensity is sometimes used to mean abundance.
activation products	Nuclides produced from the absorption of a neutron by a nucleus.
activity	Decay rate of a radionuclide; usually expressed in Becquerels (disintegrations per second), Bq.
amplitude	Zero-to-peak height of a waveform in nanometers.
array	Collection of sensors distributed over a finite area (usually in a cross or concentric pattern) and referred to as a single station.
arrival	Signal that has been associated to an event. First, the Global Association (*GA*) software associates the signal to an event. Later during interactive processing, many arrivals are confirmed and improved by visual inspection
ASCII	American Standard Code for Information Interchange. Standard, unformatted 256-character set of letters and numbers.
attribute	(1) A database column. (2) Characteristic of an item; specifically, a quantitative measure of an S/H/I arrival such as azimuth, slowness, period, and amplitude.
authentication signature	Series of bytes that are unique to a set of data and that are used to verify the authenticity of the data.
background	Contribution to a spectrum from naturally occurring radionuclides as well as interactions between radiation and materials in the vicinity of the detector.
baseline	Contribution to a spectrum from the partial energy deposition of a gamma-ray in a detector.
beta-gamma coincidence event	Nuclear decay that produces both a gamma ray and a beta particle within a very short time scale. May also refer to other photon-electron coincidence events such as an X-ray with a conversion electron.
beta particle	Electron that is produced from a nuclear decay. May also refer to other electron radiations, for example, a conversion electron.
blank subtraction	Process of removing counts from a sample spectrum or a region of interest within a sample spectrum originating from the air filter.
BLANKPHD	Blank Pulse Height Data; ASCII data message containing the pulse height data of an unexposed air filter.
CALIBPHD	Calibration Pulse Height Data; ASCII data message containing the pulse height data of a certified standard source.

calibration coefficients	Numbers that define the energy, resolution, and efficiency equations.
CD	Continuous Delivery.
centroid	Energy (in keV) or channel number at the centre of a fitted peak.
channel	(1 – RN) Energy window (in keV) representing a differential increment of pulse height. (2 – SHI) Component of motion or distinct stream of data.
CLF	Concise List of Frames.
coherent	Quality of having a fixed phase relationship; as signals from a wavefront detected on numerous seismic or infrasonic array station elements.
concentration	Activity per unit volume of air.
counts	(1 – RN) Number of pulses observed within a spectrum channel. (2 – SHI) Units of digital waveform data.
critical level/limit	Minimum net counts that must be contained in an ROI for nuclide identification (Lc).
CSCI	Computer Software Configuration Item.
CSS	Center for Seismic Studies
CTBT	Comprehensive Nuclear Test-Ban Treaty (the Treaty).
DACS	Distributed Application Control System.
DB	Database.
dB	Decibel.
defining	Arrival attribute, such as arrival time, azimuth, or slowness, which is used in calculating the event's location or magnitude.
defining phase	Associated phase for which features are used in the estimation of the location and origin time of an S/H/I event.
DETBKPHD	Detector Background Pulse Height Data; ASCII data message containing the pulse height data from a background count.
detection limit	The smallest amount of activity that can be reliably detected and quantified in a spectrum. This quantity is used to determine the MDC.
ECR	Energy versus Channel Regression; an equation providing the initial detector-specific relationship between channel number and energy. The equation contains calibration coefficients and is estimated from a transmitted calibration dataset.
EER	Efficiency versus Energy Regression; an equation providing the detector-specific relationship between efficiency and energy.
energy	Usually refers to the measured kinetic energy of radiation quanta deposited in a detector. The unit most appropriate for such measure-

	ments is keV.
epoch(al) time	Number of seconds after January 1, 1970 00:00:00.0. Conventionally used to 1-second resolution, but extended in the context of the IDC Database Schema to a resolution of one millisecond.
ERD	Entity-Relationship Diagram.
event	(1 – RN) Occurrence that displays characteristics indicative of a possible nuclear test. (2 – SHI) Unique source of seismic, hydroacoustic, or infrasonic wave energy that is limited in both time and space.
FFT	Fast Fourier Transform.
fission (P)	Particulates created in a fission event.
fission (G)	Gases created in a fission event.
f-k	Frequency versus wavenumber (k) analysis that maps phase power from an array as a function of azimuth and slowness.
FPID	Fission Product Identification.
FTP	File Transfer Protocol; protocol for transferring files between computers.
FULL SPHD	Full Sample Pulse Height Data; ASCII data message containing the pulse height data of a sample acquired for a complete collection interval.
FWHM	Full Width at Half-Maximum; metric of detector resolution and equivalent to the width of a photopeak (in keV) taken at the peak height equal to half the maximum peak counts.
GA	Global Association application. *GA* associates S/H/I phases to events.
gamma	Gamma-ray.
gamma ray	Photon that is produced from a nuclear transition; may also imply other photon radiations, for example, an X-ray.
GARDS	Global Atmospheric Radionuclide Detection System.
GASBKPHD	Gas Background Pulse Height Data. Data type sent by noble gas monitoring systems that observe a memory effect during sample acquisition due to atoms from the previous sample adsorbed onto the walls of the gas cell.
GMT	Greenwich Mean Time.
HASE	Hydroacoustic Azimuth and Slowness Estimator.
hydroacoustic	Pertaining to sound in the ocean.
IDC	International Data Centre.
IIR	Infinite Impulse Response (filters also referred to as recursive filters).
IMS	International Monitoring System.

infrasonic	Pertaining to low-frequency (sub- audible) sound in the atmosphere.
Internet	World-wide network of computers linked by means of the IP protocol.
IP	Internet Protocol.
kB	kilobyte. 1,024 bytes.
keV	kilo-electron Volts; a metric of kinetic energy.
keyline	Photon with the highest detection probability.
Lc	Critical level.
local	(1) (distance) Source to seismometer separations of a few degrees or less. (2) (event) Recorded at distances where the first P and S waves from shallow events have travelled along direct paths within the crust.
m	(1) Meter(s). (2) Megabyte(s); 1,024 kilobytes. (3) Month(s). (4) Minute(s).
mb	Magnitude of a seismic body wave.
mbar	Millibar.
mbmle	Magnitude of an event based on maximum likelihood estimation using seismic body waves.
mBq	MilliBecquerel.
MDA	Minimum Detectable Activity.
MDC	Minimum Detectable Concentration.
ML	Magnitude based on waves measured near the source.
mm	Millimetre.
Ms	Magnitude of seismic surface waves.
Msmle	Magnitude of an event based on maximum likelihood estimation using surface waves.
multiplet	Spectral region of interest comprised of more than one photopeak.
NA	Not Applicable.
NDC	National Data Centre.
NID	Nuclide Identification.
nm	Nanometre.
non-defining	Arrival attribute, such as arrival time, azimuth, or slowness, which is associated, but not used in calculating the event's location or magnitude.
non-defining phase	Associated phase for which features are not used in the estimating the location and origin time of an S/H/I event.
NSE	Noise Spectrum Equalization.

nuclide	One of many combinations of nucleons that may comprise an atomic nucleus. Because all nuclides of interest with respect to CTBT compliance verification are radioactive, this term is often used to refer specifically to radionuclide.
Oracle	Vendor of IDC database management system.
origin	Hypothesized time and location of a seismic, hydroacoustic, or infrasonic event. Any event may have many origins. Characteristics such as magnitudes and error estimates may be associated with an origin.
parameter (par) file	ASCII file containing values for parameters of a program. Par files are used to replace command line arguments. The files are formatted as a list of [token = value] strings.
peak	Statistically significant increase in counts above a spectrum baseline at an energy associated with a gamma line of a particular radionuclide or other phenomenon.
PHD	Pulse Height Data; a format for spectral data messages. Possible PHD data message types include BLANKPHD, CALIBPHD, DETBKPHD, GASBKPHD, QCPHD, and SAMPLEPHD.
PIDC	Refers to the former Prototype International Data Centre, Arlington, Virginia
polarization	Form of three-component analysis used to derive azimuth and slowness information from non-array stations.
PREL SPHD	Preliminary Sample Pulse Height Data; ASCII data message containing the pulse height data of a sample acquired for less than a complete collection interval.
primary seismic	IMS seismic station(s) or data that is (are) part of the detection network.
PS	Peak Search.
QC	Quality Control.
QCPHD	Quality Control Pulse Height Data; ASCII data message containing the pulse height data of a certified source.
quefrency	Time-delay axis with units of seconds for a cepstrum.
RN	Radionuclide.
radioactivity	See “activity”.
radionuclide	Nuclide that has an unstable nucleus, that is, a radioactive nuclide.
REB	Reviewed Event Bulletin; the bulletin formed of all S/H/I events that have passed analyst inspection and quality assurance review.
region of interest	Region of a radionuclide spectrum or histogram that corresponds to a particular radionuclide.
regional	(1) (distance) Source to seismometer separations between a few degrees and 20 degrees. (2) (event) Recorded at distances where the first P and S waves from shallow events have travelled along paths

	through the uppermost mantle.
RER	Resolution (versus) Energy Regression; an equation providing the initial detector- specific relationship between resolution and energy. This equation contains calibration coefficients and is interpolated from a transmitted calibration spectrum.
residual	Difference in time, azimuth, or slowness between a calculated attribute and its corresponding theoretical value.
RLR	Radionuclide Lab Report; report containing sample analysis results from a certified radionuclide laboratory.
RMS	Radionuclide Monitoring System; the part of the IMS that monitors the atmosphere for radionuclide.
RNPS	Radionuclide Network Product Summary; daily report containing a summary of the Radionuclide Network for a three-day period, including the data received, their products, and any relevant nuclides.
ROI	Region of interest.
RRR	Reviewed Radionuclide Report. Electronic file containing the final results of the interactive review of the automated radionuclide processing. It contains sections on sample information, measurement categorization, measured radionuclide quantities, MDCs, radionuclide identification, analyst editing, processing parameters, data quality flags, event screening flags, calibration equations, and field of regard.
sample	Any physical entity counted on a detector.
SAMPLEPHD	Sample Pulse Height Data; ASCII data message containing pulse height data acquired by counting a gas or particulate sample with a detector system.
S/H/I	Seismic, hydroacoustic, and infrasonic.
SID	Sample ID; unique alphanumeric string assigned to a sample during the automated processing for identification and accounting purposes.
singlet	Spectrum photopeak consisting of counts from one mono-energetic gamma- ray; photopeak containing counts from multiple photons, but fit as if it is comprised of counts from only one because contributions from the individual radiations cannot be separated, as in a multiplet.
SLSD	Standard List of Signal Detections.
SOH	State of Health; indicator of a system's operability.
spectrum	(1 – RN) Plot of the differential number of pulses (in counts) per differential pulse height (in channels or keV). (2 – SHI) Plot of the energy contained in waveforms as a function of frequency.
SPHD	Sample Pulse Height Data; ASCII data message type containing the pulse height data of a sample, as well as other information. The two

	types of SPHDs are full and preliminary.
SQL	Structured Query Language; a language for manipulating data in a relational database.
SSREB	Standard Screened Radionuclide Event Bulletin; bulletin generated by the IDC when fission or activation products are detected at a radionuclide station above normal limits. A SSREB contains information on the possible event, source location, fission products, activation products detected, any isotopic ratios calculated, and any certified laboratory results. New event information can be added to the SSREB as it arrives, therefore, multiple revisions of an SSREB are possible.
STA/LTA	Short-term average/long-term average ratio.
Sum/Natural	Artificial entries in the nuclide library that enable the automated processing to identify commonly observed sum peaks in a spectrum.
TCP/IP	Transmission Control Protocol/Internet Protocol.
teleseismic	(1) (distance) Source to seismometer separations of 20 degrees or more. (2) (event) Recorded at distances where the first P and S waves from shallow events have travelled paths through the mantle/core.
time series	Time ordered sequence of data samples. Typically a waveform or derived from waveforms, such as a beam.
Treaty	Comprehensive Nuclear-Test-Ban Treaty (CTBT).
Type I Error	Spectral region of interest falsely identified as a peak by the automated processing.
Type II Error	Peak undetected by the automated processing.