

Database documentation: marlin
A metadata database for Ministry of Fisheries research data
K. A. Mackay

NIWA Internal Report No. 90
2000

Contents

| | |
|--|--------------------|
| 1 Database documentation series | 3 |
| 2 Fisheries research metadata | 3 |
| 3 Data structures | 4 |
| 4 Table summaries | 10 |
| 5 marlin tables | 11 |
| 5.1 Table 1: data sets | 11 |
| 5.2 Table 2: defined regions | 14 |
| 5.3 Table 3: keywords | 15 |
| 5.4 Table 4: organisations | 16 |
| 5.5 Table 5: projects | 17 |
| 5.6 Table 6: persons | 18 |
| 5.7 Table 7: positions | 20 |
| 5.8 Table 8: units | 20 |
| 5.9 Table 9: voyages | 21 |
| 5.10 Table 10: platforms | 21 |
| 5.11 Table 11: data set urls | 22 |
| 5.12 Table 12: data set geographic extents | 22 |
| 5.13 Table 13: data set keywords | 23 |
| 5.14 Table 14: data set platforms | 23 |
| 5.15 Table 15: data set species | 23 |
| 6 marlin business rules | 24 |
| 6.1 Introduction to business rules | 24 |
| 6.2 Summary of rules | 25 |
| 7 Acknowledgements | 31 |
| 8 References | 31 |
| Appendix 1 – Reference Code Tables | 32 |

List of Figures

| | |
|--|-------------------|
| Figure 1: Entity Relationship Diagram (ERD) of the marlin database | 6 |
|--|-------------------|

1 Database documentation series

The National Institute of Water and Atmospheric Research (NIWA) currently carries out the role of Data Manager and Custodian for the fisheries research data owned by the Ministry of Fisheries.

The Ministry of Fisheries data set incorporates historic research data, data collected more recently by MAF Fisheries prior to the split in 1995 of Policy to the Ministry of Fisheries and research to NIWA, and currently data collected by NIWA and other agencies for the Ministry of Fisheries.

This document provides an introduction to the Ministry of Fisheries research data metadatabase **marlin**, and is a part of the database documentation series produced by NIWA.

All documents in this series include an introduction to the database design, a description of the main data structures accompanied by an Entity Relationship Diagram (ERD), and a listing of all the main tables. The ERD graphically shows how all the tables fit in together, and their relationships to other databases.

This document is intended as a guide for users and administrators of the **marlin** database.

Access to this database and data is restricted to specific Nominated Personnel as specified in the current Schedule 6 of the Data Management contract between the Ministry of Fisheries and NIWA. Any requests for data should in the first instance be directed to the Ministry of Fisheries.

Fisheries research metadata

2.1 Background

In the past few years, the use of metadata to describe data holdings by research organisations has gradually been accepted as a necessary tool for locating and describing datasets. There is a paucity of appropriate examples of the use of metadata in research organisations, especially relating to marine spatial datasets that NIWA specialises in, forcing NIWA to look elsewhere for relevant metadata standards. Within Australia, initiatives such as the Ocean Rescue 2000 “Blue Pages” theme directory for marine and coastal datasets (AODC, 1996) as well as the Australia New Zealand Land Information Council (ANZLIC)’s developing regional standard for geospatial metadata (ANZLIC, 1996-8) have stimulated organisations such as CSIRO Marine Research to start to collect metadata according to a local standard using metadata elements and terminology compliant with the ANZLIC metadata standard and the “Blue Pages”.

In 1997 CSIRO Marine Research developed an in-house metadatabase termed the Marine Laboratories Information Network or “MarLIN” (Rees & Ryba, 1998). MarLIN was based on the pre-existing “Environmental Data Directory” (EDD or “Green Pages”) software developed by Environment Australia. After a demonstration by CSIRO Marine Research in mid-2000, NIWA

and the Ministry of Fisheries received permission to adopt MarLIN as the basis for their metadatabase for the fisheries research data. MarLIN, in a slightly modified form, is installed on the Ministry of Fisheries research data server “snapper” as a relational database called **marlin** by NIWA, who act as the data custodians.

Nomenclature

- MarLIN refers to the Ministry of Fisheries research data metadatabase at a conceptual level;
- **marlin** is the physical implementation of MarLIN on the “snapper” server.

3 Data structures

3.1 Metadata elements in MarLIN

The “Green Pages” application, used as the basis for CSIRO Marine Research to develop MarLIN, contains metadata fields, which are based on the core metadata elements proposed by ANZLIC, plus certain other fields considered useful to its own organisation (Environment Australia). CSIRO Marine Research reviewed these pre-existing elements and retained many of them in either modified or unmodified form, and then added additional elements for three purposes: first, for compatibility with version 1.0 of the “Blue Pages” data dictionary; second, to hold additional information relevant to CSIRO Marine Research’s activities; and third, to facilitate searching, metadata entry, and database administration. Supporting tables are used whenever it is efficient to do so, for example to store information about contact persons, organisations, vessel (platform) names, and details of research projects and voyages, defined regions and their bounding coordinates, and species codes and names.

One particular area that concerned CSIRO Marine Research when developing MarLIN was the use of subject categories to which datasets could be assigned. This was only partly catered for by the original ANZLIC “Search Words” and Blue Pages “Themes” (which have been retained separately to ensure compliance with these two systems). In essence, the ANZLIC search words were done at a fairly high level (e.g., “Oceanography – Physical”), while the Blue Pages themes were more exact but were not particularly structured in its coverage of marine science topics.

Therefore, MarLIN incorporates subject categories based on those developed by ASFIS (Aquatic Sciences and Fisheries Information System) for their ASFA-1, ASFA-2, ASFA-3, and ASFA-4 bibliographic databases (CSA, 1998). Some minor adjustments have been made to the subject category terminology for the sake of conciseness and/or clarity.

Individual MarLIN subject categories have been assigned to one of eight broad “subject areas” for more rapid searching at a high level. These are presently as follows:

- Aquatic biology, fisheries, pollution, conservation, and biotechnology
- Descriptive and dynamical oceanography
- Aquatic chemistry and geochemistry
- Marine meteorology and climate

- Marine and coastal geology and geophysics
- Aquatic non-living resources
- Underwater acoustics
- Optics and marine technology
- Aquatic law, policy, economics, and sociology

Species-level taxonomic information in MarLIN is handled by incorporating the Ministry of Fishers 3-character species codes into the metadata record. This differs somewhat from the CSIRO Marine Research's implementation, which incorporates CAAB (Codes for Australian Aquatic Biota) numeric species codes.

3.2 Database description

This database contains several tables. The ERD for **marlin** (Figure 1) shows the logical structure of the database and its entities (each entity is implemented as a database *table*) and relationships between these tables. Each table represents an object, event, or concept in the real world that is selected for representation in the database. Each *attribute* of a table is a defining property or quality of the table.

All of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary key¹. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed. Most of the tables in the **marlin** database have some attributes, called foreign keys², which contain standard NIWA fisheries codes, such as species. These attributes provide links to supporting tables within **marlin** and the **rdb** (research dat**a**base) database.

Section 5 shows a listing of all the **marlin** tables as implemented by the Empress DBMS. As can be seen in the listing of the tables, a table's primary key has a unique index on it. Primary keys are generally listed using the format:

Indices: UNIQUE index_name ON (*attribute* [, *attributes*])

where the attribute(s) make up the primary key (the key attributes) and the index name is the primary key name. Note that the typographical convention for the above format is that square brackets [] may contain more than one item or none at all. A unique index prevents records with duplicate key values from being inserted into the table; e.g., a dataset with an existing dataset identifier.

¹ A primary key is an attribute or a combination of attributes that contains an unique value to identify that record.

² A foreign key is any attribute, or a combination of attributes, in a table that is a primary key of another table. Tables are linked together through foreign keys.

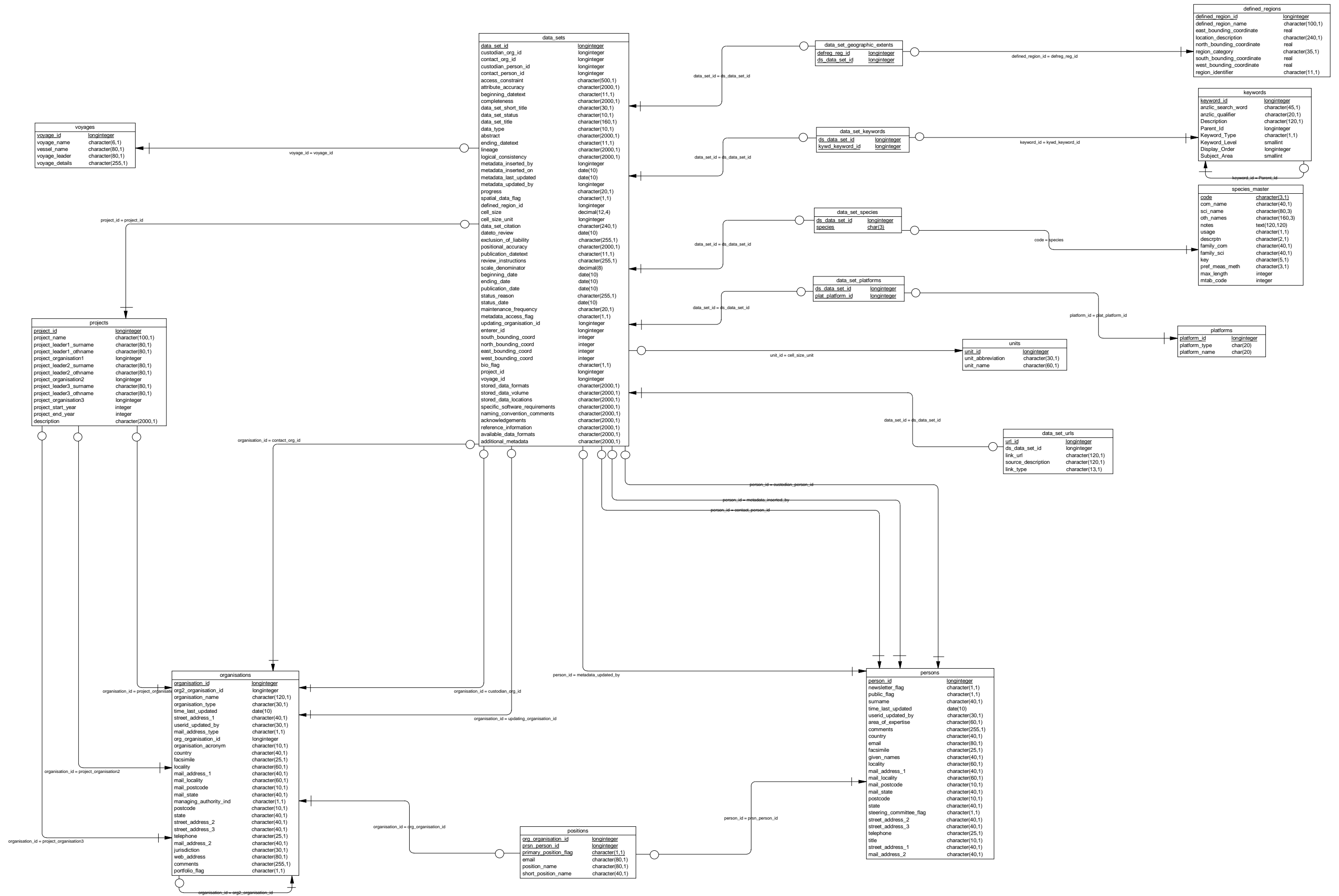


Figure 1: Entity Relationship Diagram (ERD) of the marlin database.

The **marlin** database is implemented as a relational database. That is, each table is a special case of the mathematical construct known as a *relation* and hence elementary relation theory is used to deal with the data within tables and the relationships between them. All relationships in **marlin** are of the type *one-to-many*³. This is shown in the ERD by connecting a single line (indicating ‘many’) from the child table; e.g., *data_sets*, to the parent table; e.g., *projects*, with an arrow-head (indicating ‘one’) pointing to the parent. For example, consider the relationship between the tables’ *projects* (the parent table) and *data_sets* (the child table). Any one project in *projects* can have one or more datasets in *data_sets*, but any one dataset can only be a part of one project. Note that the word ‘many’ applies to the possible number of records another is associated with. For a given instance, there might be zero, one, two, or more associated records, but if it is ever possible to have more than one, we use the word ‘many’ to describe the association.

One-to-many relationships can be either mandatory or optional. The optional relationship, denoted in the ERD by the symbol “O” at one or both ends of the relationship line, means that a record does not have to have any associated records. Conversely, the mandatory relationship denoted in the ERD by a bar symbol across the relationship line, means that a record has to have at least one associated record. For example, if we consider again the one-to-many relationship between the tables *projects* and *data_sets*, which has a mandatory ‘one’ and an optional ‘many’. This means that one project record can have zero or more (many) datasets within it, but one dataset must have one, and only one, associated record in the project table.

These relationships are enforced in the database by the use of referential constraints⁴. Constraints do not allow *orphans* to exist in any table; i.e., where a child record exists without a related parent record. This may happen when:

1. a parent record is deleted;
2. the parent record is altered so the relationship is lost;
3. or a child record is entered without a parent record.

All constraints in **marlin** prevent the latter from occurring. Constraints are shown in the table listings by the following format:

```
Referential:      constraint_name (attribute[, attribute])  INSERT
                                                           DELETE
                   Parent_table (attribute[, attribute])
```

where a constraint can either be INSERT (prevents (iii) and (ii) from occurring) or DELETE (prevents (i) or (ii) from occurring). For example, consider the following constraint found in the table *data_sets*:

```
Referential:      ds_cust_org_fk (organisation_org_id) INSERT organisations
                   (organisation_id)
```

This means that the value of the attribute *organisation_org_id* in the current dataset record must already exist in the parent table *organisations* or the record will be rejected and the following

³ A one-to-many relationship is where one record (the *parent*) in a table relates to one or many records (the *child*) in another table; e.g., one survey in *t_survey* can have many strata in *t_stratum* but any one stratum can only come from one survey.

⁴ Also known as integrity checks.

error message will be displayed:

```
*** User Error: insert constraint 'ds_cust_org_fk' violation
```

All tables in this database are indexed. That is, attributes that are most likely to be used as a searching key have like values linked together to speed up searches. These indices are listed using the following format:

```
Indices:    NORMAL (2, 15) index_name ON (attribute [ , attributes ])
```

Note that indices may be simple, pointing to one attribute, or composite pointing to more than one attribute. The numbers "... (2, 15) ..." in the syntax are Empress DBMS default values relating to the amount of space allocated to index storage.

3.3 Database Design

As reflected by the ERD, the main table of **marlin** is the individual data set record table, *data_sets* (Table 1). Each data set is uniquely identified by an integer, stored as the attribute *data_set_id*. The *data_sets* table is deliberately open in its data types by have large character fields so as to allow more qualitative and descriptive data entry. The data set resolution is recorded either as a scale as stored in the attribute *scale_denominator* (e.g., 1:1,000,000) or in cell size stored in the attribute *cell_size* with the cell size units being stored in the attribute *cell_size_unit*.

For all tables, each record is identified a one (or a combination of) unique integers which make up the tables primary key.

Pre-defined regions are stored in the *defined_regions* table (Table 2). The original specifications for MarLIN, as designed by CSIRO Marine Research, called for generic regions such as oceans and seas to be stored and defined by a minimum-bounding rectangles (MBRs) defined by north, south, east, and west coordinates. This table is also ideal for storing fisheries research areas such as statistical areas, QMAs, and FMAs, however, MBRs are quite unsuitable for these instances of regions. So a future modification would be to link a data set in the *data_sets* table directly to an area polygon in the *map* table of the **rdb** database so that spatial queries can be performed directly.

Keywords and subject categories are hierarchically defined so that each keyword in the *keywords* (Table 3) may be a "parent" or a "child" of other keywords. This hierarchical structure is maintained by have the attribute *parent_id* point to the primary key of another record in the *keywords* table. In the ERD, this is reflected by the keywords entity pointing to itself.

Similarly, organisations are hierarchically defined so that each organisation in the *organisations* (Table 4) may be a "parent" or a "child" of other organisations; e.g., Fisheries with NIWA. The attribute *org_organisation_id* can point to the primary key of another record in the *organisations* table. In the ERD, this is reflected by the keywords entity pointing to itself. General details of the organisation are recorded here, such as mailing address, street address, phone and fax numbers,

and home WWW pages.

Projects result in one or many data sets. Details for individual projects are stored in the *projects* (Table 5) table. It is assumed that projects may be interdisciplinary and involve more than one organisation. Provision has been made to store up to three organisations associated with one project by storing their organisation identifiers in the attributes *project_organisation1*, *project_organisation2*, and *project_organisation3* respectively.

Details for individual people involved with data sets are stored in the table *persons* (Table 6). General details for people are recorded here, such as name, title, mailing address, street address, phone and fax numbers, and e-mail address.

The link between *persons* and *organisations* is through the table positions (Table 7). Conceptually, one person can fill positions within more than one organisations, and an organisation has many people filling position. The table positions records one relationship between a person and an organisation. A Yes/No flag *primary_position_flag* denotes whether-or-not the person/organisation combination constitutes the primary position for that person. The positions e-mail address is recorded here, where it is different to the person personal e-mail address; e.g. dba@frc.niwa.cri.nz

Many data sets may be directly referenced on the WWW. The URL addresses for each data set are stored in the table *data_set_urls* (Table 11).

The tables *data_set_geographic_extents* (Table 12), *data_set_keywords* (Table 13), *data_set_platforms* (Table 14), and *data_set_species* (Table 15) are all classic examples of how to resolve many-to-many relationships in a relational database. In each these cases, one defined region/keyword/platform/species can be represented in many data sets, and one data set can have many defined regions/keywords/platforms/species. To resolve this, these intermediate tables have been created to explicitly store each instance of a data set and defined region/keyword/platform/species relation.

A platform, as stored in the *platforms* table, may need some further defining. In most fisheries research scenarios, a ship is the usual means by which gear is deployed and data collected. However, airplanes may also be used (for aerial sighting of tuna schools, for instance). These are *specializations* of a generic platform entity. The platforms tables records details by storing the platform type (e.g. “Ship” or “Aircraft”) and platform name (e.g., “FV Tangaroa” or “ZK-FDG”).

3.4 Standards for fisheries databases

The **marlin** database was created in mid-2000 as more-or-less a direct copy of the MarLIN system developed by CSIRO Marine Research. In 1993, a set of standards was set in place (Ng 1992) for all fisheries databases. The most significant effect of these standards has been the requirement of adding of the prefix “t_” to the table names and “v_” to view names. However, this raised some potentially serious issues. The **marlin** database is just one part of a metadata system including CGI scripts and HTML pages. Modifying marlin to meet these standards would

therefore have a very significant flow-on effect to all relevant fisheries applications. Therefore, table names in marlin have been intentionally left as written and do not meet fisheries standards.

4 Table summaries

The **marlin** database has 15 tables for containing metadata. The following is a listing and brief outline of the tables contained **marlin**:

1. **data_sets** : contains profile metadata information for individual data sets.
2. **defined_regions** : contains brief descriptions of pre-defined regions (e.g., oceans, statistical areas, QMA, FMA), including the minimum bounding rectangle.
3. **keywords** : contains keywords (words or phrases summarizing aspects of the data set), keyword types, and their subject areas.
4. **organisations** : contains details about the various organisations associated with data sets.
5. **projects** : contains details about the various projects associated with data sets.
6. **persons** : contains details about the various people associated with data sets.
7. **positions** : contains details about positions that people fill within organisations.
8. **units** : Details of units used by data sets
9. **voyages** : contains details of individual voyages resulting in data sets.
10. **platforms** : contains details of the platforms used to collect data for the data sets.
11. **data_set_urls** : contains URL links pertaining to the data set.
12. **data_set_geographic_extents** : records the relationships between data sets and defined regions.
13. **data_set_keywords** : records the relationships between data sets and keywords.
14. **data_set_platforms** : records the relationships between data sets and platforms.
15. **data_set_species** : records the relationships between data sets and species

5 marlin tables

The following are listings of the tables in the **marlin** database, including attribute names, data types (and any range restrictions), and comments.

5.1 Table 1: data_sets

Comment: Profile metadata information for individual data sets.

| Attributes | Data Type | Null? | Comment |
|----------------------|--|-------|---|
| data_set_id | longinteger | No | Unique identifier for metadata record |
| custodian_org_id | longinteger | | Identifier for custodian organisation, refer "organisations" table |
| contact_org_id | longinteger | No | Identifier for contact organisation, refer "organisations" table |
| custodian_person_id | longinteger | | Identifer for custodian person, refer "persons" table |
| contact_person_id | longinteger | No | Identifier for contact person, refer "persons" table |
| access_constraint | char(500,1) | No | Description of any access constraints |
| attribute_accuracy | char(2000,1) | No | Description of attribute accuracy |
| beginning_datetext | char(11,1) | | Dataset beginning date e.g. "May 2000" |
| completeness | char(2000,1) | No | Description of dataset completeness |
| data_set_short_title | char(30,1) | No | Used for quick searching, or to hold pre-existing identifiers such as CD-ROM serial numbers |
| data_set_status | char(10,1) smatch "Active Obsolete Deleted" | No | Data set status flag |
| data_set_title | char(160,1) | No | Data set full title |
| data_type | char(10,1) | No | Data type e.g., Observed, Model, Special, Document, Inventory, Other |
| abstract | char(2000,1) | No | Brief narrative summary |
| ending_datetext | char(11,1) | No | Dataset ending date e.g. "May 2000" |
| lineage | char(2000,1) | No | |
| logical_consistency | char(2000,1) | No | |
| metadata_inserted_by | longinteger | No | Identifier of person who inserted data refer "persons" table |
| metadata_inserted_on | date(4) | No | Date dataset inserted |

| | | | |
|--------------------------|--|----|--|
| metadata_last_updated | date(4) | No | Date dataset last updated |
| metadata_updated_by | longinteger | No | Identifier of person who updated data refer "persons" table |
| progress | char(20,1) smatch "Complete In progress Planned Not Known" | No | Data set progress of flag. |
| spatial_data_flag | char(1,1) smatch "[YN]" | No | Spatial data flag (Y/N) |
| defined_region_id | longinteger | | Identifier for a pre-defined area, refer "defined_regions" table |
| cell_size | decimal(12,4) | | Resolution of data. |
| cell_size_unit | longinteger | | Identifier for data resolution units refer "units" table |
| data_set_citation | char(240,1) | | Data set reference citation |
| dateto_review | date(4) | | Date of next data set review |
| exclusion_of_liability | char(255,1) | | |
| positional_accuracy | char(2000,1) | | Brief comment on the data set positional accuracy (where relevant) |
| publication_datetext | char(11,1) | | Date of publication e.g., "May 1997" |
| review_instructions | char(255,1) | | |
| scale_denominator | decimal(8,0) | | Data set scale (where relevant) |
| beginning_date | date(4) | | Exact dataset beginning date |
| ending_date | date(4) | | Exact dataset ending date |
| publication_date | date(4) | | Date of publication |
| status_reason | char(255,1) | | |
| status_date | date(4) | | |
| maintenance_frequency | char(20,1) smatch "Annually As Required Bi-annually Weekly Not Known Quarterly Not Planned Monthly Irregular Daily Continual" | No | Data set maintenance frequency flag |
| metadata_access_flag | char(1,1) smatch "[YN]" | No | Is the metadata record available to be access yet? (Y/N) |
| updating_organisation_id | longinteger | | Identifier for updating organisation, refer "organisations" table |
| enterer_id | longinteger | No | Identifier for person submitting dataset, refer "persons" table |
| south_bounding_coord | integer | | Minimum latitude of data (-90 to +90) |

| | | |
|------------------------|--------------|---|
| north_bounding_coord | integer | Maximum latitude of data (-90 to +90) |
| east_bounding_coord | integer | Maximum longitude of data (0 to 360) |
| west_bounding_coord | integer | Minimum longitude of data (0 to 360) |
| bio_flag | char(1,1) | Are there any biological parameters for this data set (Y/N) |
| project_id | longinteger | Identifier for project, refer "projects" table |
| voyage_id | longinteger | Identifier for voyage, refer "voyages" table |
| stored_data_formats | char(2000,1) | Description of the format that the data set is stored in. |
| stored_data_volume | char(2000,1) | Description of the volume of the data set |
| stored_data_locations | char(2000,1) | Location(s) of the data set |
| specific_software_req | char(2000,1) | Minimum software requirements needed to access the data set |
| naming_convention_comm | char(2000,1) | Comments on any naming conventions used in the data set |
| acknowledgements | char(2000,1) | List of data set acknowledgements |
| reference_information | char(2000,1) | References cited by the data set |
| available_data_formats | char(2000,1) | Data formats available for export |
| additional_metadata | char(2000,1) | Any additional metadata |

Creator: dba

Referential: ds_unit_fk (cell_size_unit) INSERT units (unit_id)
ds_upd_fk (updating_organisation_id) INSERT organisations (organisation_id)
ds_cus_org_fk (custodian_org_id) INSERT organisations (organisation_id)

Indices: UNIQUE BTREE ds_pk ON (data_set_id)

5.2 Table 2: defined_regions

Comment: Brief descriptions of pre-defined regions (e.g., QMA, FMA), including the minimum bounding rectangle.

| Attributes | Data Type | Null? | Comment |
|---------------------------|--|-------|--|
| defined_region_id | longinteger | No | Unique identifier for the defined region record |
| defined_region_name | char(100,1) | No | Defined region name |
| location_description | char(240,1) | | Brief description of the defined region and its location |
| region_category | char(35,1) smatch "General Regions Ocean and Seas User Defined" | No | Type of region flag |
| north_bounding_coordinate | real | No | Maximum latitude of data (-90 to +90) |
| south_bounding_coordinate | real | No | Minimum latitude of data (-90 to +90) |
| west_bounding_coordinate | real | No | Minimum longitude of data (0 to 360) |
| east_bounding_coordinate | real | No | Maximum longitude of data (0 to 360) |
| region_identififier | char(11,1) | | Short title of the defined region |

Creator: dba

Indices: UNIQUE BTREE defreg_pk ON (defined_region_id)

5.3 Table 3: keywords

Comment: Table of keywords (words or phrases summarizing aspects of the data set), keyword types, and their subject areas.

| Attributes | Data Type | Null? | Comment |
|--------------------|--------------------|-------|---|
| keyword_id | longinteger | No | Unique identifier for each keyword record |
| anzlic_search_word | char(45,1) | | |
| anzlic_qualifier | char(20,1) | | |
| Description | char(120,1) | No | Keyword description |
| Parent_Id | longinteger | No | Identifier to the parent keyword |
| Keyword_Type | char(1,1) | | 1-char code for keyword type. A = ANZLIC search words B = Blue Pages themes E = Equipment H = Habitat P = Blue Pages parameters S = MarLIN subject categories T = taxonomy |
| | smatch "[ABEHPST]" | | |
| Keyword_Level | smallint | | Level within a keyword tree |
| Display_Order | longinteger | | Keyword display order |
| Subject_Area | smallint | | 2-digit code for general subject area 10 = biological 20 = non-biological general 21 = oceanography 22 = chemistry 23 = meteorology 24 = geology and geophysics 25 = non-living resources 26 = technology, acoustics and optics 27 = human related |
| | range 10, 20-27 | | |

Creator: dba
Indices: UNIQUE BTREE kywd_pk ON (keyword_id)

5.4 Table 4: organisations

Comment: Contains details about various organisations associated with data sets.

| Attributes | Data Type | Null? | Comment |
|----------------------|-------------|-------|--|
| organisation_id | longinteger | No | Unique identifier for the organisation |
| organisation_name | char(120,1) | No | Organisation name |
| organisation_type | char(30,1) | No | Organisation type |
| time_last_updated | date(4) | No | Date record last updated |
| userid_updated_by | char(30,1) | No | Person who last update the record |
| org_organisation_id | longinteger | | Identifier for the parent organisation |
| organisation_acronym | char(10,1) | | Organisation acronym (if any) |
| mail_address_1 | char(40,1) | | Mailing address (line 1) |
| mail_address_2 | char(40,1) | | Mailing address (line 2) |
| mail_locality | char(60,1) | | Mailing locality |
| mail_postcode | char(10,1) | | Mailing address postcode |
| mail_state | char(40,1) | | Mailing address state (if relevant) |
| street_address_1 | char(40,1) | No | Street address (line 1) |
| street_address_2 | char(40,1) | | Street address (line 2) |
| street_address_3 | char(40,1) | | Street address (line 3) |
| postcode | char(10,1) | | Postcode (if different from above) |
| state | char(40,1) | | State (if different from above) |
| country | char(40,1) | | Country name |
| telephone | char(25,1) | | Telephone number |
| facsimile | char(25,1) | | Fax number |
| locality | char(60,1) | | Organisation locality |
| jurisdiction | char(30,1) | | |
| web_address | char(80,1) | | Organisation WWW address |
| comments | char(255,1) | | |

Creator: dba

Referential: org_org_fk (org_organisation_id) INSERT organisations (organisation_id)

Indices: UNIQUE BTREE org_pk ON (organisation_id)

5.5 Table 5: projects

Comment: Contains details about the various projects associated with data sets.

| Attributes | Data Type | Null? | Comment |
|-------------------------|--------------|-------|--|
| project_id | longinteger | No | Unique identifier for the project record |
| project_name | char(100,1) | No | Project name |
| project_leader1_surname | char(80,1) | | First project leader's surname |
| project_leader1_othname | char(80,1) | | First project leader's other names |
| project_organisation1 | longinteger | | Identifier for the first project leader's organisation, refer "organisations" table |
| project_leader2_surname | char(80,1) | | Second project leader's surname |
| project_leader2_othname | char(80,1) | | Second project leader's other names |
| project_organisation2 | longinteger | | Identifier for the second project leader's organisation, refer "organisations" table |
| project_leader3_surname | char(80,1) | | Third project leader's surname |
| project_leader3_othname | char(80,1) | | Third project leader's other names |
| project_organisation3 | longinteger | | Identifier for the third project leader's organisation, refer "organisations" table |
| project_start_year | integer | | Project start year (YYYY format) |
| project_end_year | integer | | Project end year (YYYY format) |
| description | char(2000,1) | | |

Creator: dba

Indices: UNIQUE BTREE proj_pk ON (project_id)

5.6 Table 6: persons

Comment: Contains details about the various people associated with data sets.

| Attributes | Data Type | Null? | Comment |
|-------------------------|----------------------------|-------|--|
| person_id | longinteger | No | Unique identifier for each person record |
| surname | char(40,1) | No | Surname |
| given_names | char(40,1) | | Given names |
| title | char(10,1) | | Title e.g. Mr, Mrs, Dr, etc |
| street_address_1 | char(40,1) | | Street address (line 1) |
| street_address_2 | char(40,1) | | Street address (line 2) |
| street_address_3 | char(40,1) | | Street address (line 3) |
| postcode | char(10,1) | | Postcode |
| state | char(40,1) | | State (if relevant) |
| country | char(40,1) | | Country |
| email | char(80,1) | | Email address |
| telephone | char(25,1) | | Telephone number |
| facsimile | char(25,1) | | Fax number |
| locality | char(60,1) | | Locality |
| mail_address_1 | char(40,1) | | Mailing address (line 1) |
| mail_address_2 | char(40,1) | | Mailing address (line 2) |
| mail_locality | char(60,1) | | Mailing address locality |
| mail_postcode | char(10,1) | | Mailing address postcode |
| mail_state | char(40,1) | | Mailing address state (if relevant) |
| area_of_expertise | char(60,1) | | Brief description of the person's area of expertise (if any) |
| steering_committee_flag | char(1,1) smatch "[YN]" | No | Is this person on a steering committee? |
| newsletter_flag | char(1,1) smatch "[YN]" | No | Is this person on the newsletter mailing list? |

| | | | |
|-------------------|---------------|----|--|
| public_flag | char(1,1) | No | Is this person a member of the public? |
| | smatch "[YN]" | | |
| time_last_updated | date(4) | No | Date record last updated |
| userid_updated_by | char(30,1) | No | Person who last updated record |
| comments | char(255,1) | | |

Creator: dba
Indices: UNIQUE BTREE prsn_pk ON (person_id)

5.7 Table 7: positions

Comment: Contains details about positions that people fill within organisations.

| Attributes | Data Type | Null? | Comment |
|-----------------------|----------------------------|-------|--|
| org_organisation_id | longinteger | No | Identifier for the organisation that holds the position, refer "organisations" table |
| prsn_person_id | longinteger | No | Identifier for the person who fills the position, refer "persons" table |
| primary_position_flag | char(1,1) smatch "[YN]" | No | Flag to note whether this is the persons primary position (Y/N) |
| email | char(80,1) | | Position email address |
| position_name | char(80,1) | | Full position name |
| short_position_name | char(40,1) | | Short position name |

Creator: dba

5.8 Table 8: units

Comment: Table containing details of units used by the various data sets.

| Attributes | Data Type | Null? | Comment |
|-------------------|-------------|-------|---------------------------------------|
| unit_id | longinteger | No | Unique identifier for the unit record |
| unit_abbreviation | char(30,1) | | Unit abbreviation e.g., cm, km |
| unit_name | char(60,1) | | Description of units |

Creator: dba

Indices: UNIQUE BTREE unit_pk ON (unit_id)

5.9 Table 9: voyages

Comment: Contains details of individual voyages resulting in data sets.

| Attributes | Data Type | Null? | Comment |
|-----------------|-------------|-------|---|
| voyage_id | longinteger | No | Unique identifier for the voyage record |
| voyage_name | char(7,1) | No | Voyage name or trip code identifier. Can be used to link to other databases such as trawl, plankton, scallop etc. |
| vessel_name | char(80,1) | | Vessel name |
| voyage_leader | char(80,1) | | Voyage leader's name |
| voyage_details | char(255,1) | | Brief description of the voyage |
| Creator: | dba | | |

5.10 Table 10: platforms

Comment: Contains details of the platforms used to collect data for the data sets.

| Attributes | Data Type | Null? | Comment |
|-----------------|---|-------|---|
| platform_id | longinteger | No | Unique identifier for the platform record |
| platform_type | char(20,1) | | Type of platform e.g., SHIP, AIRPLANE |
| platform_name | char(20,1) | | Platform name |
| Creator: | dba | | |
| Indices: | UNIQUE BTREE platform_pk ON (platform_id) | | |

5.11 Table 11: data_set_urls

Comment: Table of URL links pertaining to the data set.

| Attributes | Data Type | Null? | Comment |
|--------------------|---|-------|--|
| url_id | longinteger | No | Unique identifier for the URL record |
| ds_data_set_id | longinteger | No | Identifier for a data set, refer "data_sets" table |
| link_url | char(120,1) | No | URL address |
| source_description | char(120,1) | | Brief description of the URL source |
| link_type | char(13,1) smatch "Graphic Documentation Data" | No | Type of data that the URL links to the data set |

Creator: dba
Referential: dsurl_ds_fk (ds_data_set_id) INSERT data_sets (data_set_id)
Indices: UNIQUE BTREE dsurl_pk ON (url_id)

5.12 Table 12: data_set_geographic_extents

Comment: Table of recording the relationships between data sets and defined regions.

| Attributes | Data Type | Null? | Comment |
|----------------|-------------|-------|---|
| ds_data_set_id | longinteger | No | Identifier for a data set, refer "data_sets" table |
| defreg_reg_id | longinteger | No | Identifier for a define region, refer "defined_regions" table |

Creator: dba
Referential: ds_geo_defreg_fk (defreg_reg_id) INSERT defined_regions (defined_region_id)
ds_geo_ds_fk (ds_data_set_id) INSERT data_sets (data_set_id)
Indices: UNIQUE BTREE ds_geo_pk ON (ds_data_set_id, defreg_reg_id)

5.13 Table 13: data_set_keywords

Comment: Table of recording the relationships between data sets and keywords.

| Attributes | Data Type | Null? | Comment |
|-----------------|-------------|-------|--|
| ds_data_set_id | longinteger | No | Identifier for a data set, refer "data_sets" table |
| kywd_keyword_id | longinteger | No | Identifier for a keyword, refer "keywords" table |

Creator: dba

Referential: dskey_ds_fk (ds_data_set_id) INSERT data_sets (data_set_id)
dskey_kywd_fk (kywd_keyword_id) INSERT keywords (keyword_id)

Indices: UNIQUE BTREE dskey_pk ON (ds_data_set_id, kywd_keyword_id)

5.14 Table 14: data_set_platforms

Comment: Table of recording the relationships between data sets and platforms.

| Attributes | Data Type | Null? | Comment |
|----------------|-------------|-------|--|
| ds_data_set_id | longinteger | No | Identifier for a data set, refer "data_sets" table |
| pl_platform_id | longinteger | No | Identifier for a platform, refer "platforms" table |

Creator: dba

Referential: ds_plat_plat_fk (pl_platform_id) INSERT platforms (pl_platform_id)
ds_plat_ds_fk (ds_data_set_id) INSERT data_sets (data_set_id)

Indices: UNIQUE BTREE ds_plat_pk ON (ds_data_set_id, pl_platform_id)

5.15 Table 15: data_set_species

Comment: Table of recording the relationships between data sets and species codes.

| Attributes | Data Type | Null? | Comment |
|----------------|-------------|-------|---|
| ds_data_set_id | longinteger | No | Identifier for a data set, refer "data_sets" table |
| species | char(3,1) | No | Identifier to a species, refer "curr_spp" table in the rdb database |

Creator: dba

Referential: ds_spp_spp_fk (species) INSERT rdb:curr_spp (code)
ds_spp_ds_fk (ds_data_set_id) INSERT data_sets (data_set_id)

Indices: UNIQUE BTREE ds_spp_pk ON (ds_data_set_id, species)

6 marlin business rules

6.1 Introduction to business rules

The following are a list of business rules pertaining to the **marlin** database. A business rule is a written statement specifying what the information system (i.e., any system that is designed to handle marlin survey data) must do or how it must be structured.

There are three recognized types of business rules:

| | |
|-------------------|---|
| Fact | Certainty or an existence in the information system |
| Formula | Calculation employed in the information system |
| Validation | Constraint on a value in the information system |

Fact rules are shown on the ERD by the cardinality (e.g., one-to-many) of table relationships. Referential constraints, range checks, and algorithms both in the database and during data validation implement the formula and validation type rules.

6.2 Summary of rules

Fisheries research data set (data_sets)

| | |
|------------------------------|--|
| data_set_id | Data set identifier must be unique integer greater than zero. |
| custodian_org_id | Must be a valid organisation identifier as listed in the <i>organisations</i> table. |
| contact_org_id | Must be a valid organisation identifier as listed in the <i>organisations</i> table. |
| custodian_person_id | Must be a valid person identifier as listed in the <i>persons</i> table. |
| contact_person_id | Must be a valid person identifier as listed in the <i>persons</i> table. |
| access_constraint | Must have a value entered. |
| attribute_accuracy | Must have a value entered. |
| completeness | Must have a value entered. |
| data_set_status | Must contain a valid data set status code as listed in Appendix 1. |
| data_set_title | Must have a value entered. |
| data_type | Must have a value entered. |
| abstract | Must have a value entered. |
| ending_datetext | Must have a value entered. |
| lineage | Must have a value entered. |
| logical_consistency | Must have a value entered. |
| metadata_inserted_by | Must be a valid person identifier as listed in the <i>persons</i> table. |
| metadata_inserted_on | Must have a valid date. |
| metadata_last_updated | Must have a valid date. |
| metadata_updated_by | Must be a valid person identifier as listed in the <i>persons</i> table. |
| progress | Must contain a valid progress code as listed in Appendix 1. |
| spatial_data_flag | Spatial data flag must either be equal to “Y” or “N”. |
| defined_region_id | Must be a valid defined region identifier as listed in the <i>defined_regions</i> table. |

| | |
|---------------------------------|---|
| cell_size | Must be a value greater than zero. |
| cell_size_unit | Must by a valid units identifier as list in the <i>units</i> table. |
| scale_denominator | Must be a value greater than zero. |
| beginning_date | Must by a valid date. |
| ending_date | Must by a valid date. |
| | Multiple column check on beginning and end dates: The data set ending date must be after the beginning date. |
| publication_date | Must by a valid date. |
| status_date | Must by a valid date. |
| maintenance_frequency | Must contain a valid maintenance frequency code as listed in Appendix 1. |
| metadata_access_flag | Spatial data flag must either by equal to “Y” or “N”. |
| updating_organisation_id | Must be a valid organisation identifier as listed in the <i>organisations</i> table. |
| enterer_id | Must be a valid person identifier as listed in the <i>persons</i> table. |
| south_bounding_coord | Must be an integer between the values of 90 and –90 inclusive. |
| north_bounding_coord | Must be an integer between the values of 90 and –90 inclusive. |
| | Multiple column check on north and south bounding coordinates: The south-bounding coordinate must be less than the north-bounding coordinate. |
| east_bounding_coord | Must be an integer between the values of 0 and 360 inclusive. |
| west_bounding_coord | Must be an integer between the values of 0 and 360 inclusive. |
| | Multiple column check on east and west bounding coordinates: The west-bounding coordinate must be less than the east-bounding coordinate. |
| project_id | Must be a valid project identifier as listed in the <i>projects</i> table. |
| voyage_id | Must be a valid voyage identifier as listed in the <i>voyages</i> table. |

Pre-defined regions table (defined_regions)

| | |
|-----------------------------|---|
| defined_region_id | Defined region identifier must be unique integer greater than zero. |
| defined_region_name | Must have a value entered. |
| region_category | Must contain a valid region category code as listed in Appendix 1. |
| north_bounding_coord | Must be an integer between the values of 90 and -90 inclusive. |
| south_bounding_coord | Must be an integer between the values of 90 and -90 inclusive. |

Multiple column check on north and south bounding coordinates:
The south-bounding coordinate must be less than the north-bounding coordinate.

| | |
|----------------------------|---|
| east_bounding_coord | Must be an integer between the values of 0 and 360 inclusive. |
| west_bounding_coord | Must be an integer between the values of 0 and 360 inclusive. |

Multiple column check on east and west bounding coordinates:
The west-bounding coordinate must be less than the east-bounding coordinate.

Keywords table (keywords)

| | |
|----------------------|---|
| keyword_id | Keyword identifier must be unique integer greater than zero. |
| Description | Must have a value entered. |
| Parent_id | Must be a valid voyage identifier as listed in the <i>keywords</i> table. |
| Keyword_Type | Must contain a valid keyword type code as listed in Appendix 1. |
| Keyword_Level | Must be a integer greater or equal to zero. |
| Display_Order | Must be a integer greater or equal to zero. |
| Subject_Area | Must contain a valid subject area code as listed in Appendix 1. |

Organisations table (organisations)

| | |
|----------------------------|---|
| organisation_id | Organisation identifier must be unique integer greater than zero. |
| organisation_name | Must have a value entered. |
| organisation_type | Must have a value entered. |
| time_last_updated | Must be a valid date. |
| userid_updated_by | Must be a valid person as listed in the <i>persons</i> table. |
| org_organisation_id | Must be a valid organisation as listed in the <i>organisations</i> table. |
| street_address_1 | Must have a value entered. |

Projects table (projects)

| | |
|------------------------------|---|
| project_id | Project identifier must be unique integer greater than zero. |
| project_name | Must have a value entered. |
| project_organisation1 | Must be a valid organisation as listed in the <i>organisations</i> table. |
| project_organisation2 | Must be a valid organisation as listed in the <i>organisations</i> table. |
| project_organisation3 | Must be a valid organisation as listed in the <i>organisations</i> table. |
| project_start_year | Must be a 4-digit year and fall within the reasonable range of 1900-2000 |
| project_end_year | Must be a 4-digit year and fall within the reasonable range of 1900-2000 |

Multiple column check on project start and end years:

The project end year must be after the project start year..

Unit codes table (units)

| | |
|----------------|---|
| unit_id | Unit identifier must be unique integer greater than zero. |
|----------------|---|

Personnel table (persons)

| | |
|--------------------------------|---|
| person_id | Project identifier must be unique integer greater than zero. |
| surname | Must have a value entered. |
| steering_committee_flag | Steering committee flag must either be equal to “Y” or “N”. |
| newsletter_flag | Newsletter flag must either be equal to “Y” or “N”. |
| public_flag | Public flag must either be equal to “Y” or “N”. |
| time_last_updated | Must be a valid date. |
| userid_updated_by | Must be a valid person as listed in the <i>persons</i> table. |

Personnel positions table (positions)

| | |
|------------------------------|---|
| org_organisation_id | Must be a valid organisation as listed in the <i>organisations</i> table. |
| prsn_person_id | Must be a valid person as listed in the <i>persons</i> table. |
| primary_position_flag | Primary position flag must either be equal to “Y” or “N”. |

Voyage details table (voyages)

| | |
|--------------------|---|
| voyage_id | Voyage identifier must be unique integer greater than zero. |
| voyage_name | Must have a value entered. |

Platform details table (platforms)

| | |
|--------------------|---|
| platform_id | Platform identifier must be unique integer greater than zero. |
|--------------------|---|

URL links table (data_set_urls)

| | |
|-----------------------|---|
| url_id | URL identifier must be unique integer greater than zero. |
| ds_data_set_id | Must be a valid data set as listed in the <i>data_sets</i> table. |
| link_url | Must have a value entered. |
| link_type | Must contain a valid link type code as listed in Appendix 1. |

Data set / Defined regions link table (data_set_geographic_extents)

| | |
|-----------------------|---|
| ds_data_set_id | Must be a valid data set as listed in the <i>data_sets</i> table. |
| defreg_reg_id | Must be a valid defined region as listed in the <i>defined_regions</i> table. |

Data set / Keywords link table (data_set_keywords)

| | |
|------------------------|---|
| ds_data_set_id | Must be a valid data set as listed in the <i>data_sets</i> table. |
| kywd_keyword_id | Must be a valid keyword as listed in the <i>keywords</i> table. |

Data set / Platforms link table (data_set_platforms)

| | |
|-----------------------|---|
| ds_data_set_id | Must be a valid data set as listed in the <i>data_sets</i> table. |
| pl_platform_id | Must be a valid platform as listed in the <i>platforms</i> table. |

Data set / Species link table (data_set_species)

| | |
|-----------------------|---|
| ds_data_set_id | Must be a valid data set as listed in the <i>data_sets</i> table. |
| species | Must be a valid species code as listed in the <i>species_master</i> table in the rdb database. |

Acknowledgements

The author would like to thank Dave Banks for his editorial contribution to this document.

8 References

Australia New Zealand Land Information Council (ANZLIC) (1996-8). Core Metadata Elements for Land and Geographic Directories in Australia and New Zealand. Digital document, available online at: <http://www.anzlic.org.au/download.htm>

Australian Oceanographic Data Centre (AODC) (1996). The marine and coastal data directory of Australia – the Blue Pages, Version 1.0. Online at: <http://www.erin.gov.au/marine/mcdd>

Cambridge Scientific Abstracts (CSA) (c. 1998). Database Classification Codes: ASFA and Oceanic Abstracts. WWW document available online at <http://www.csa.com/helpV3/classcod.html>

Ng, S. 1992: Standards for setting up databases and their applications. *MAF Fisheries Greta Point Internal Report No 180*. 31p.

Rees, A.J.J. and Ryba, M.M. (1998). MarLIN – a metadatabase for research data holding at CSIRO Marine Research. Paper presented at the first Australian Marine and Coastal Data Management Conference, Hobart, November 1998. Available online at: http://www.marine.csiro.au/datacentre/ext_docs/malinpaper.html

Appendix 1 – Reference Code Tables

Data set status code

Active
Obsolete
Deleted

Progress code

Complete
In progress
Planned
Not Known

Maintenance frequency code

Annually
As Required
Bi-annually
Weekly
Not Known
Quarterly
Not Planned
Monthly
Irregular
Daily

Region category code

General
Regions
Ocean and Seas
User Defined

Keyword type code

| | |
|---|---------------------------|
| A | ANZLIC search words |
| B | Blue Pages themes |
| E | Equipment |
| H | Habitat |
| P | Blue Pages parameters |
| S | MarLIN subject categories |
| T | taxonomy |

Subject area code

| | |
|----|----------------------------------|
| 10 | biological |
| 20 | non-biological general |
| 21 | oceanography |
| 22 | chemistry |
| 23 | meteorology |
| 24 | geology and geophysics |
| 25 | non-living resources |
| 26 | technology, acoustics and optics |
| 27 | human related |

URL link type code

Graphic
Documentation
Data