

Database documentation for the Ministry for Primary Industries

market sampling database:

market

D. O. Fisher & K. A. Mackay

NIWA Fisheries Data Management
Database documentation series

Revised 25 Sep 2020

Contents

1	Introduction to the Database Document series.....	6
2	Market Sampling Programme	6
2.1	Data sources.....	6
2.2	Data validation.....	7
3	Data Structures	8
3.1	Table relationships.....	8
3.2	Database design	13
3.2.1	Paua and eel data in the <i>t_landing</i> table	15
3.2.2	Otoliths.....	15
4	Table Summaries.....	15
5	market Tables	16
5.1	Table 1: <i>t_landing</i>	16
5.2	Table 2: <i>t_catch</i>	18
5.3	Table 3: <i>t_stratum</i>	19
5.4	Table 4: <i>t_cluster</i>	21
5.5	Table 5: <i>t_length</i>	22
5.5.1	View: <i>v_eel_lf</i>	24
5.6	Table 6: <i>t_fish_bio</i>	25
6	market business rules	27
6.1	Introduction to business rules	27
6.2	Summary of rules	28
	Appendix 1 – Data Integrity	37
	Appendix 2 – Reference code tables.....	39
	Appendix 3 – Data Forms.....	43

List of Figures

Figure 1: Entity Relationship Diagram (ERD) for the market database.	9
Figure 2: ERD showing the relationships between the market tables and tables in the rdb database.....	12

Revision History

Version	Change	Date	Responsible
1.0	First release as MAF Fisheries Greta Point Internal Report No. 198	1992	Kevin Mackay
1.1	NIWA Internal Report No. 22. Dropped area from t_catch. Dropped area from primary key of t_stratum, etc.	1998	David Fisher with Kevin Mackay
1.2	Added Business Rules. Added cluster_no to t_fish_bio. Changed t_landing.landing_weight from longinteger to decimal(7,1).	Aug 2000	David Fisher
1.3	Appendix 2 expanded to include reasonable weight ranges	21 Mar 2001	David Fisher
1.4	Added proj_code attribute to table t_landing	27 Aug 2001	David Fisher
1.5	Added Revision History, changed stratum_code from 3 to 4 char	May 2005 Updated ERDs Aug05	David Fisher
1.6	Added measure_meth2 and lgth2 attributes to t_fish_bio.	Feb 2008	Christopher Dick
1.7	Increased length of comments char fields: t_landing - vessel1_name & port from 20 and shed from 25. t_cluster and t_fish_bio comments fields increased from char(60,1).	July 2008	Christopher Dick
1.8	Updated forms in Appendix 3.	October 2009	David Fisher
1.9	Added rdb reference code tables to Appendix 2.	November 2013	David Fisher
2.1	Postgres version.	April 2015	D Fisher & F Wei
2.1.1	Updated ERD, Figure 1 re species Corrected column order in ERD for t_cluster	Sep 2015 Sep 2016	D Fisher D Fisher
2.1.2	Dropped Not Null on t_fish_bio measure_meth column.	Sep 2020	D Fisher

1 Introduction to the Database Document 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 for Primary Industries (MPI) formerly the Ministry of Fisheries.

This MPI data set, incorporates historic research data, data collected by MAF Fisheries prior to the split in 1995 of Policy to the Ministry of Fisheries and research to NIWA, and data collected by NIWA and other agencies for the Ministry of Fisheries and subsequently for MPI.

This document is a brief introduction to the market and shed sampling database **market**, and is part of the database documentation series produced by NIWA. It supersedes the previous documentation by Fisher & Mackay (1998)¹ on this database.

Up to September 1992 this database was known as the **csdb** database. To avoid confusion with the catch sampling data, all market and shed sampling data was moved into the **market** database.

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 link together and their relationship with other databases.

This document is intended as a guide for users and administrators of the **market** database. This database has been implemented as a schema within the Postgres database called **fish**.

2 Market Sampling Programme

2.1 Data sources

The **market** database is designed for market sampling data as collected by the MAF Fisheries Stock Monitoring Programme (SMP) and various regional market sampling programmes. Latterly, market samples are taken by NIWA staff collecting research data in all regions. The measuring usually takes place at a fish processing facility or shed, and is representative of that portion of the catch that is marketed or sold. A shed or market sample involves measuring a sample (or subset of the landed catch) from a landing of fish at, or subsequent to the point of landing. Primarily this database contains data for snapper, hoki, paua, and red cod, but also includes data on orange roughy, trevally, kahawai, freshwater eels, gemfish, bluenose, scallops, alfonsino, ruby fish, jack mackerels, black cardinal fish, English mackerel, grey mullet, red gurnard, kingfish, and tarakihi.

The basic unit for a market sample is a fishing trip terminating with a landing of fish. Any one landing can contain several species available to be sampled; e.g., JMD, JMM. Each species is treated as a different sample. It is also important to note that species from the same landing, but caught from a different area, are treated as different samples.

¹ FISHER, D.O. and MACKAY, K.A., 1998: Database documentation: market. *NIWA Internal Report No. 22*. 28p.

Each landing is given a landing number, starting with a four-digit year; e.g., 2000, followed by a 4 digit sequential landing number would represent a landing from 2000. Up to and including 1999 only 2 digits were used for the year. Some samples prior to 1991 have a 3 digit sequential number following the year. Landing number formats change with different species or sampling programmes.

Each sample can be divided into different strata, and each stratum can be further subdivided into clusters. A stratum may be a truckload of boxes from the boat, a fish slurry, or pre-graded fish. For some species, notably snapper sampled by NIWA Auckland, fish have already been divided into strata when landed. A sub-sample is then generated by randomly selecting individual clusters (where a cluster may be a box, bin, or a sack of fish) from the different strata. Individual fish are measured from these clusters.

Usually, the landing is unstratified. In this case the whole landing is treated as one stratum. Therefore, a sample for a species is generated by randomly selecting clusters from the whole landing. Individual fish are measured from these clusters.

A further requirement for some market sampling programmes is to collect biological data from each sample. A number of fish from each sample are selected at random for further biological measurements, including gonad staging, removal of a subset of otoliths, and stomach content analysis. Historically, this has been a requirement for the red cod programme, and more recently biological data for freshwater eels, both the short- and long-finned species, comprising length, sex, gonad stage, and weight were collected for the entire sample.

2.2 Data validation

While the **market** database enforces data validation and integrity rules with the use of referential constraints and range checks, data go through rigorous data validation and error checking process before being entered.

These process include instructions for data recording, simple data validation using Perl, and C programming language scripts, followed by inserting data into a loading database. See Appendix 1 for a more detailed description of the processes involved.

3 Data Structures

3.1 Table relationships

This database contains several tables. The ERD for **market** (Figure 1) shows the logical structure² of the database and its entities (each entity is implemented as a database *table*) and relationships between these tables and tables in other databases. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed. The ERD's in this document show attributes within the tables with generic data-types.

Each table represents an object, event, or concept in the real world that is selected to be represented 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 key³.

Note that Figure 1 shows the main tables only. All the tables in the **market** database have attributes, called foreign keys⁴, which contain standard NIWA/MPI fisheries codes, such as *species* and *fishing_method*. These foreign keys not only define the relationships between the tables in **market** but also provide links to the **rdb** (research database) database, which contains the definitive list of these standard codes. A more detailed ERD for these tables follows (Figure 2) showing the relationships between **market** and **rdb**.

All tables within external databases, such as those in **rdb** and **age**, are shown in the ERDs as being enclosed in dashed-line boxes.

The **market** database is implemented as a relational database; i.e., 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. There are three types of relationships possible between tables, but only one exists in **market**: one-to-many⁵. These relationships can be seen in ERDs by connecting a single line (indicating 'many') from the child table; e.g., *t_catch*, to the parent table; e.g., *t_landing*, with an arrowhead (indicating 'one') pointing to the parent.

Every relationship has a mandatory or optional aspect to it. If a relationship is mandatory, then it has to occur at least once, while an optional relationship might not occur at all. For example, in Figure 1, consider that relationship between the table *t_stratum* and its child table *t_cluster*. The symbol 'o' by the child *t_cluster* means that a cluster record can have zero or many cluster records (eg fish boxes or bins sampled), while the bar by the parent *t_stratum* means that for every cluster record there must be a matching stratum record.

² Also known as a database *schema*.

³ A primary key is an attribute or a combination of attributes that contains an unique value to identify that record.

⁴ 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.

⁵ 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 landing in *t_landing* can have many catches in *t_catch* but one catch can only come from one landing.

Physical Data Model
 Model: market
 Package:
 Diagram: PHYSICALDIAGRAM_1
 Author: sma Date : 24/09/2015
 Version : 1.8

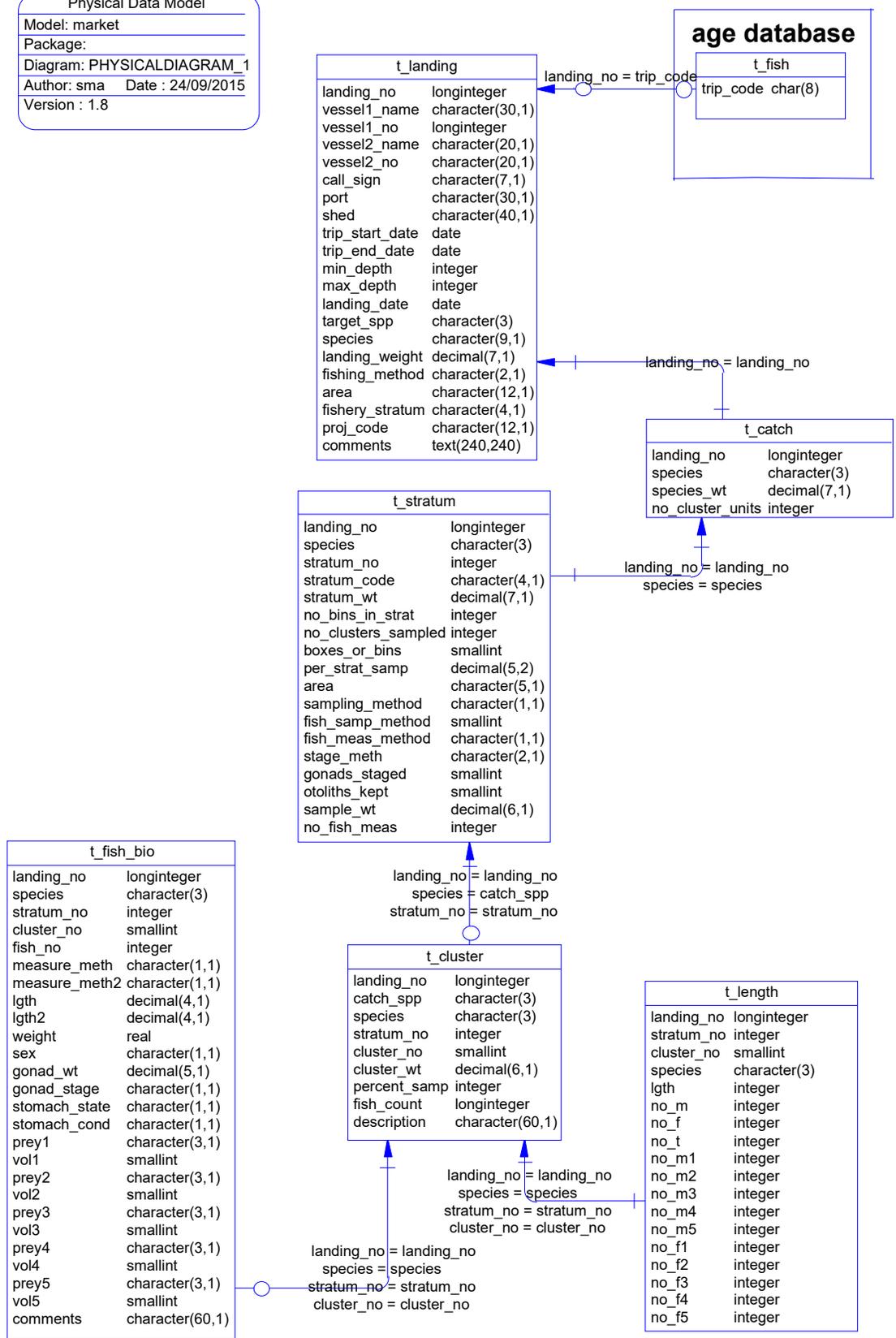


Figure 1: Entity Relationship Diagram (ERD) for the market database.

These links are enforced by referential constraints⁶. 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: a parent record is deleted; the parent record is altered so the relationship is lost; or a child record is entered without a parent record

Constraints are shown in the table listings by the following format:

```
Foreign-key constraints:  
  "foreign key name" FOREIGN KEY (attribute[,attribute]) REFERENCES  
    parent table (attribute[, attribute])
```

Note that the typographical convention for the above format is that square brackets [] may contain more than one item or none at all. Items stacked between vertical lines | | are options of which one must be chosen.

For example, consider the following constraint found in the table *t_landing*:

```
Foreign-key constraints:  
  "fk_t_landing_species_master" FOREIGN KEY (target_spp) REFERENCES  
    rdb.species_master(code)
```

This means that the value of the attribute *target_spp* in the current record must already exist in the parent table *species_master* of the **rdb** schema or the record will be rejected and the following message will be displayed:

```
ERROR: insert or update on table "t_landing" violates foreign key constraint  
"fk_t_landing_species_master"
```

For tables residing in different schemas, the parent table name will be prefixed by the name of the database.

Section 5 lists all the **market** tables as implemented by the Postgres 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 following format:

```
Indices: index_name PRIMARY KEY, btree (attribute [, attributes ])
```

where attribute(s) make up the primary key and the index name is the primary key name. These prevent records with duplicate keys from being inserted into the tables; e.g., a record with an existing landing number.

The database listing (Tables 1-6) show that the tables also have indices on some attributes. 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: index_name btree (attribute)
```

⁶ Also known as integrity checks.

Note that indices may be simple, pointing to one attribute or composite pointing to more than one attribute.

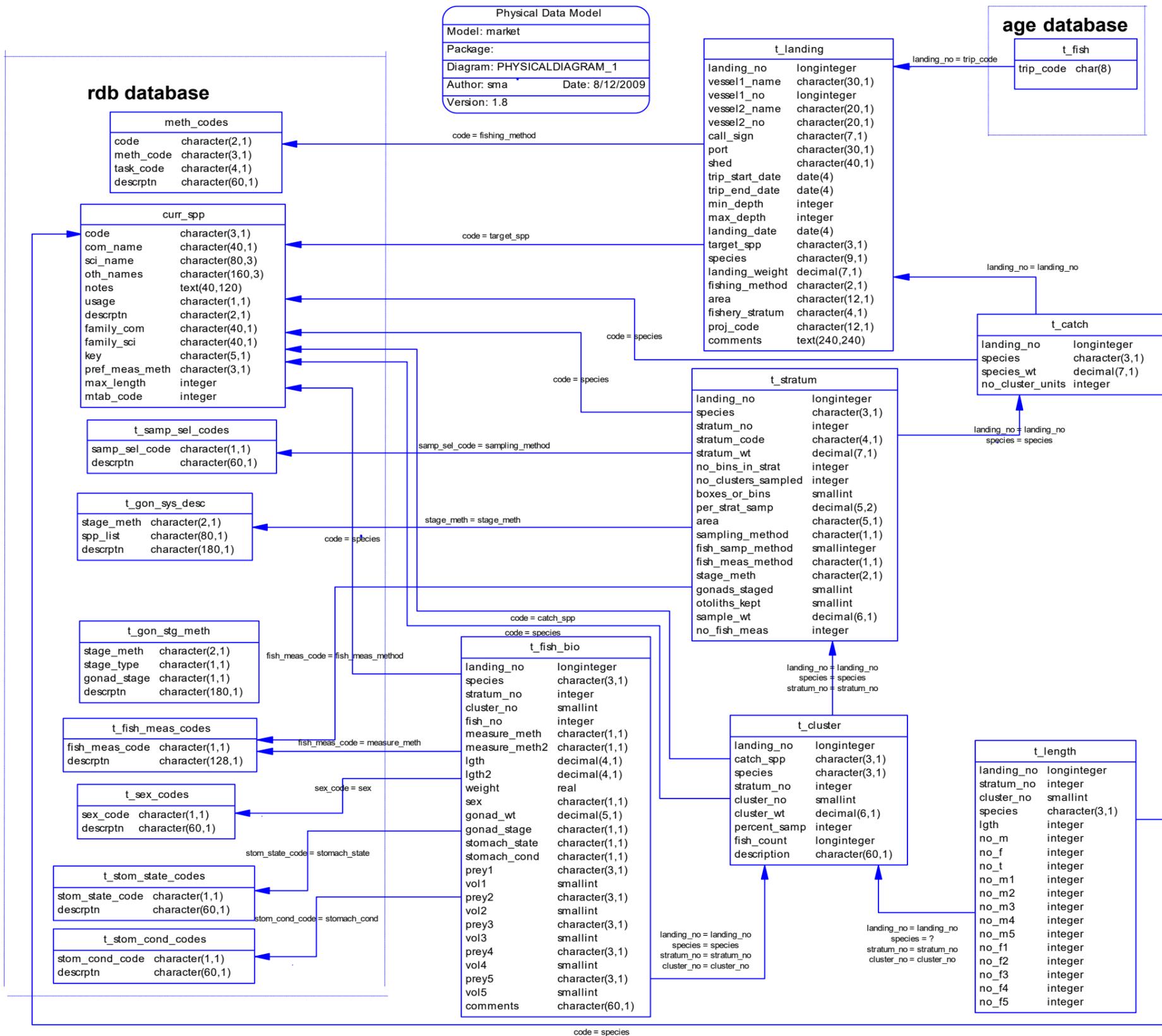


Figure 2: ERD showing the relationships between the **market** tables and tables in the **rdb** database.

3.2 Database design

The top-level table in **market** is *t_landing* (Table 1). This holds information for each landing made by a vessel(s) or fisher(s). Each record is identified by a landing number, represented by the attribute *landing_no*, which is also its primary key. Only details for landed species that were sampled are recorded in this table.

The primary key *landing_no* also provides a one-to-many link to the **age** database. Through this key, both databases can be joined together to extract age, length, and catch data for any one landing.

To cope with a landing of fish comprising of multiple species from multiple areas the attributes *species* and *area* differ slightly from usage in other tables within market, or indeed other databases. Normally their attributes are 3 and 4 character fields respectively. This table, however, stores them as 36 and 12 character fields respectively. The table has been structured this way to allow at the most twelve combinations of species and three combinations of area for any one landing. For example, consider an individual landing with the following species/area combinations:

Species	Area
ORH	2A
ORH	2B
EPT	2A

This would be stored in the *t_landing* table as:

ATTRIBUTE:	<i>species</i>	<i>area</i>
VALUE:	ORHORHEPT	2A 2B 2A

Any future instances of more combinations of species and/or area for any one landing will have to be handled by simply increasing the size of the attributes by multiples of 3 or 4 respectively.

Note that the *t_landing* attribute, *fishing_meth* uses the NIWA/MPI fisheries standard research codes, and therefore *t_landing* has a referential constraint to the *meth_codes* table in **rdb** (see Figure 2). A landing record can only be inserted into this table if and only if the value in *fishing_meth* exists in the **rdb** table *meth_codes*.

The design of the sampling programme for some fisheries or species involves stratifying the year's landings from the fishery. This may involve stratification; e.g., by fishing method, season or vessel size. This is recorded in the attribute *fishery_stratum*. For example, the sampling of the species bluenose is stratified by 3 vessel types: large catch trawlers, other trawlers, and all other fishing methods, as well as by season.

A typical market sample involves collecting a representative, roughly random sample of fish lengths, which is held in the *t_length* and or *t_fish_bio* tables. Some sampling however involves collecting a random sample for age only, typically 30 to 50 fish at present. These samples may only be recorded in the *t_landing*, *t_catch*, and *t_stratum* tables of the **market** database, with the data on individual fish held in the *t_fish* table in the **age** database.

For each landing, the total landed weight and total number of cluster units for each species from which a sample was derived are stored in the table *t_catch* (Table 2) with a primary key of *landing_no* and *species*.

Some species have a stratified sampling strategy within a catch. In such cases, the sample is subdivided into several strata before the fish are measured. Each stratum from the catch becomes a record in the table *t_stratum* (Table 3). However, for most species, the catch is one stratum, so the *t_stratum* records are identical to the *t_catch* records. In either case, this adds the attribute *stratum_no* to the catch primary key to provide the new unique key for this table. Three other attributes in this table, *sampling_method*, *fish_meas_meth*, and *stage_meth*, are for codes that are documented in **rdb** and therefore each have referential constraints to the **rdb** database.

Within each stratum, the catch is subdivided into subsamples or clusters. From these clusters, individual fish are selected for measurement. A cluster unit is usually a box, sack, or bin of fish. Each cluster sampled is recorded in the table *t_cluster* (Table 4). The attribute *cluster_no* is added to the primary key of *t_stratum* to become the primary key for this table.

Samples of fish are usually taken for length measurements from at least one cluster, but often from many. These measurements are summed by centimetre length class and sex/gonad stage to become records in the table *t_length* (Table 5). The exception are shellfish, which are summed by millimetre length class. For a length class the number of males, females, and total fish are stored. There is no attribute in *t_length* for numbers of unsexed fish. This value can be calculated by:

$$\text{number unsexed} = no_t - (no_m + no_f)$$

Currently where recorded in *t_length*, all gonads are staged to a 5-point scale. These gonad stages are hard coded into the table as attributes, so the numbers of stage 3 females are stored in the attribute *no_f3*. However, the exact definition of what is a stage 3 female is dynamic, and different species, and sometimes different surveys of the same species, have their own unique gonad staging methodology. This methodology is denoted by a code recorded in the *t_stratum* attribute *stage_meth* and relates to a full description as recorded in the *t_gon_sys_desc* and *t_gon_stg_meth* tables in the **rdb** database (Figure 2).

For some samples, a number of fish are selected for more detailed biological examination. These biological records are held in the table *t_fish_bio* (Table 6). The primary key *landing_no*, *species*, *stratum_no*, *cluster_no*, and *fish_no* uniquely identify records. The attribute *fish_no* is an integer that identifies each fish within the catch. Most measurements for fish in the *t_fish_bio* table are recorded in cm rounded down. Generally the fish are selected from throughout the whole stratum.

All measured specimens for both species of freshwater eels (short- and long-finned) are weighed and data held in *t_fish_bio*.

These eel data are summed by centimetre length class and loaded into *t_length* as standard length frequency data. Unlike all other species, eel data includes the value 'immature' as a valid sex code. This is stored in the *t_length* table in the *no_ml* attribute. To clarify this situation, a view *v_eel_lf* (section 5.5.1) is provided, with more suitable attribute names, as the preferred means for users to access these data. This view filters the records of *t_length* for these particular species only.

3.2.1 Paua and eel data in the *t_landing* table

Paua and eels may produce confusion for users in *t_landing*, as the fisher's name and Fisher Identification Number (FIN) are recorded in the attributes *vessel_name* and *vessel_no* respectively. This is because fishing for these species may not involve a vessel, and **market** does not have attributes specifically for the fisher's name and FIN, which are the equivalent identifiers to vessel name and vessel number.

3.2.2 Otoliths

The *market* database does not hold data on the number of otoliths or other specimens that have been collected for aging purposes. The *age* database should be consulted for these data.

4 Table Summaries

The **market** database has six tables containing market sampling data and one view showing species specific length data.

The following is a listing and brief outline of the tables contained in **market**:

1. **t_landing** : contains relevant information for a landing from a fishing trip from which fish were sampled.
2. **t_catch** : contains the catch weights of species sampled from the landing.
3. **t_stratum** : contains details of a stratum within a catch. Weight, size, fish quality, or statistical area may stratify a catch.
4. **t_cluster** : contains details of clusters (bins or sacks of fish) sampled.
5. **t_length** : holds length frequency and gonad staging information.
 - (a) **v_eel_lf** : a view of length frequency data specifically for freshwater eels
6. **t_fish_bio** : contains biological data for fish selected from the sample.

5 market Tables

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

5.1 Table 1: t_landing

Comment: Table describing the header information for each landing from which a sample was taken.

Column	Type	Null?	Description
landing_no	integer	No	Landing number, composed of a 4 digit year followed by a number for each sample, unique per year. Prior to 2000 year was identified by 2 digits.
vessel1_name	character varying(30)	No	Vessel name - Name of the vessel that caught the sampled fish or the name of the fisher for paua or eels.
vessel1_no	integer		Vessel number - MAF licence number or the Fisher Identification Number (FIN) of the fisher.
vessel2_name	character varying(20)		Name of the pair vessel (if any used).
vessel2_no	character varying(20)		Pair vessel number - MAF licence number.
call_sign	character varying(7)		Radio call sign of the first vessel.
port	character varying(30)	No	Port where the fish was landed.
shed	character varying(40)	No	Fish shed or processor where the fish was measured and usually where the fish was landed.
trip_start_date	date		Start date of the fishing trip.
trip_end_date	date		Finish date of the fishing trip.
min_depth	integer		Minimum depth (metres) fished for the trip.
max_depth	integer		Maximum depth (metres) fished for the trip.
landing_date	date	No	Date the fish were landed.

target_spp	character(3)		3 character species code of the target species, refer rdb.curr_spp.
species	character varying(36)	No	List of 3 character species codes, refer rdb.curr_spp. Up to 12 different species can be recorded in this field.
landing_weight	numeric(7,1)		Total landed weight of all species for this landing in kg.
fishing_method	character varying(2)	No	2 digit fishing method code, refer rdb.meth_codes.
area	character varying(12)		List of 4 character area codes. Codes depend on the species sampled. Up to 3 different areas can be recorded in this field.
fishery_stratum	character varying(4)		Code for stratification of the landings from the fishery, e.g. by method, season or vessel size.
proj_code	character varying(12)		Project code for the data collection. This field is not populated for data received prior to 1 Sep 2001.
comments	text		

Indexes:

```
"pk_t_landing" PRIMARY KEY, btree (landing_no)
"nx_t_landing_area" btree (area)
"nx_t_landing_species" btree (species)
```

Foreign-key constraints:

```
"fk_t_landing_meth_codes" FOREIGN KEY (fishing_method)
REFERENCES rdb.meth_codes(code)
"fk_t_landing_species_master" FOREIGN KEY (target_spp)
REFERENCES rdb.species_master(code)
```

5.2 Table 2: t_catch

Comment: The landed weights and other details for each species sampled per landing.

Column	Type	Null?	Description
landing_no	integer	No	Landing number, composed of a 4 digit year followed by a number for each sample, unique per year. Prior to 2000 year was identified by 2 digits.
species	character(3)	No	3 character species code, refer rdb.curr_spp.
species_wt	numeric(7,1)		Weight landed (kg) of this species for this landing.
no_cluster_units	integer		Number of clusters (e.g. boxes or bins) landed of this species.

Indexes:

"pk_t_catch" PRIMARY KEY, btree (landing_no, species)

Foreign-key constraints:

"fk_t_catch_curr_spp" FOREIGN KEY (species)
REFERENCES rdb.curr_spp(code)

"fk_t_catch_t_landing" FOREIGN KEY (landing_no)
REFERENCES market.t_landing(landing_no)

5.3 Table 3: t_stratum

Comment: Details of a stratum from the catch; i.e. a sample taken from a catch. Usually only one stratum from the catch, but with some species there may be many.

Column	Type	Null?	Description
landing_no	integer	No	Landing number, composed of a 4 digit year followed by a number for each sample, unique per year. Prior to 2000 year was identified by 2 digits.
species	character(3)	No	3 character species code, refer rdb.curr_spp.
stratum_no	integer	No	Identifying number for each stratum in the catch.
stratum_code	character varying(4)		Code describing the basis of the stratum.
stratum_wt	numeric(7,1)		Species weight (kg) landed in this stratum.
no_bins_in_strat	integer		Number of clusters (e.g. boxes or bins) of this species landed in this stratum.
no_clusters_sampled	integer		Number of clusters (e.g. boxes or bins) which fish were sampled from.
boxes_or_bins	smallint		1 digit code for the type of container from which fish were stored : boxes = 1, bins = 2, other = 3. From 2000 trays=4, sacks=6.
per_strat_samp	numeric(5,2)		Percentage of the stratum that was sampled.
area	character varying(5)		Up to 5 character code for area, refer rdb.area_codes.
sampling_method	character varying(1)		1 digit code for the method of selecting a sample, refer rdb.t_samp_sel_codes.
fish_samp_method	smallint		1 digit code for the method of selecting fish from a cluster for measurement: 1 = whole box, 2 = other.
fish_meas_method	character varying(1)		1 character code for the fish measurement method in t_length, refer rdb.t_fish_meas_codes or

Appendix 2 of the database documentation.

stage_meth	character varying(2)		2 character code for gonad staging method used, refer rdb.t_gon_sys_desc.
gonads_staged	smallint	No	Are the gonads staged ? : 1 = Yes, 2 = No, 3 = Males, 4 = Females, 5 = both.
otoliths_kept	smallint		Material collected for ageing analysis ? : 1 = Yes, 2 = No.
sample_wt	numeric(6,1)		Weight (kg) of fish sampled from this stratum.
no_fish_meas	integer		Total number of fish sampled from this stratum.

Indexes:

"pk_t_stratum" PRIMARY KEY, btree (landing_no, species, stratum_no)

Check constraints:

"t_stratum_boxes_or_bins_check" CHECK (boxes_or_bins >= 1 AND boxes_or_bins <= 7)
"t_stratum_fish_samp_method_check" CHECK (fish_samp_method >= 1 AND fish_samp_method <= 2)
"t_stratum_gonads_staged_check" CHECK (gonads_staged >= 1 AND gonads_staged <= 5)
"t_stratum_otoliths_kept_check" CHECK (otoliths_kept >= 1 AND otoliths_kept <= 2)

Foreign-key constraints:

"fk_t_stratum_area_codes" FOREIGN KEY (area)
REFERENCES rdb.area_codes(code)
"fk_t_stratum_species_master" FOREIGN KEY (species)
REFERENCES rdb.species_master(code)
"fk_t_stratum_t_catch" FOREIGN KEY (landing_no, species)
REFERENCES market.t_catch(landing_no, species)
"fk_t_stratum_t_fish_meas_codes" FOREIGN KEY (fish_meas_method)
REFERENCES rdb.t_fish_meas_codes(fish_meas_code)
"fk_t_stratum_t_gon_sys_desc" FOREIGN KEY (stage_meth)
REFERENCES rdb.t_gon_sys_desc(stage_meth)
"fk_t_stratum_t_samp_sel_codes" FOREIGN KEY (sampling_method)
REFERENCES rdb.t_samp_sel_codes(samp_sel_code)

5.4 Table 4: t_cluster

Comment: Table describing each cluster unit taken from a landing from which a sample was taken; i.e., a description of each box, bin or sack that was sampled from.

Column	Type	Null?	Description
landing_no	integer	No	Landing number, composed of a 4 digit year followed by a number for each sample, unique per year. Prior to 2000 year was identified by 2 digits.
catch_spp	character varying(3)	No	Species code to link to t_catch, typically identical to species attribute, except where catch recorded as a mix of species e.g. JMA.
species	character(3)	No	3 character species code, refer rdb.curr_spp.
stratum_no	integer	No	Identifying number for each stratum in the catch.
cluster_no	smallint	No	Sequential number for each cluster unit in the stratum (e.g. box number).
cluster_wt	numeric(6,1)		Weight (decimal kg) of the cluster unit.
percent_samp	integer		Percentage of the cluster unit sampled.
fish_count	integer		Number of fish in this cluster unit.
description	character varying(60)		

Indexes:

```
"pk_t_cluster" PRIMARY KEY, btree  
(landing_no, species, stratum_no, cluster_no)
```

Foreign-key constraints:

```
"fk_t_cluster_curr_spp" FOREIGN KEY (species)  
REFERENCES rdb.curr_spp(code)  
"fk_t_cluster_species_master" FOREIGN KEY (catch_spp)  
REFERENCES rdb.species_master(code)  
"fk_t_cluster_t_stratum" FOREIGN KEY (landing_no, catch_spp, stratum_no)  
REFERENCES market.t_stratum(landing_no, species, stratum_no)
```

5.5 Table 5: t_length

Comment: Table containing length frequency and gonad staging data.

Column	Type	Null?	Description
landing_no	integer	No	Landing number, composed of a 4 digit year followed by a number for each sample, unique per year. Prior to 2000 year was identified by 2 digits.
stratum_no	integer	No	Identifying number for each stratum in the catch.
cluster_no	smallint	No	Sequential number for each cluster unit in the stratum (e.g. box number).
species	character(3)	No	3 character species code, refer rdb.curr_spp.
lgth	integer	No	Length class, for finfish measured in whole cm truncated e.g. 32.8 = 32 cm. For shellfish including paua rounded to the nearest whole mm.
no_m	integer		Number of males of this length.
no_f	integer		Number of females of this length.
no_t	integer	No	Total number of "fish" of this length.
no_m1	integer		Number of males with stage 1 gonads of this length (number immature for freshwater eels).
no_m2	integer		Number of males with stage 2 gonads of this length.
no_m3	integer		Number of males with stage 3 gonads of this length.
no_m4	integer		Number of males with stage 4 gonads of this length.
no_m5	integer		Number of males with stage 5 gonads of this length.
no_f1	integer		Number of females with stage 1 gonads of this length.
no_f2	integer		Number of females with stage 2 gonads of this length.

no_f3	integer	Number of females with stage 3 gonads of this length.
no_f4	integer	Number of females with stage 4 gonads of this length.
no_f5	integer	Number of females with stage 5 gonads of this length.

Indexes:

```
"pk_t_length" PRIMARY KEY, btree
(landing_no, stratum_no, cluster_no, species, lgth)
```

Foreign-key constraints:

```
"fk_t_length_t_cluster" FOREIGN KEY (landing_no, species, stratum_no,
cluster_no)
REFERENCES market.t_cluster(landing_no, species, stratum_no,
cluster_no)
```

```
"fk_t_length_curr_spp" FOREIGN KEY (species)
REFERENCES rdb.curr_spp(code)
```

5.5.1 View: v_eel_lf

Comment: View of all freshwater eel length frequency data;
i.e. long-finned (=LFE) & short-finned (=SFE) eels.

Column	Type
landing_no	integer
stratum_no	integer
species	character(3)
lgth	integer
no_m	integer
no_f	integer
no_i	integer
no_t	integer

View definition:

```
SELECT t_length.landing_no, t_length.stratum_no, t_length.species,  
       t_length.lgth, t_length.no_m, t_length.no_f, t_length.no_m1 AS no_i,  
       t_length.no_t  
FROM market.t_length  
WHERE t_length.species::text = 'LFE'::text OR t_length.species::text =  
       'SFE'::text;
```

Refer to table t_length for column Descriptions

5.6 Table 6: t_fish_bio

Comment: This table holds records for fish selected for biologicals. Information includes gonad staging, stomach contents etc.

Column	Type	Null?	Description
landing_no	integer	No	Landing number, composed of a 4 digit year followed by a number for each sample, unique per year. Prior to 2000 year was identified by 2 digits.
species	character(3)	No	3 character species code, refer rdb.curr_spp.
stratum_no	integer	No	Identifying number for each stratum in the catch.
cluster_no	smallint	No	Sequential number for each cluster unit in the stratum, e.g., box number; -1 = not used.
fish_no	integer	No	Sequential number for each fish from a sample.
measure_meth	character varying(1)		Code of method used to measure fish lengths in column lgth, refer rdb.t_fish_meas_codes.
measure_meth2	character varying(1)		Code of method used to measure fish lengths in column lgth2, refer rdb.t_fish_meas_codes.
lgth	numeric(4,1)		Length (decimal cm) of the individual fish.
lgth2	numeric(4,1)		Second length (decimal cm) of the individual fish.
weight	real		Weight (grams) of the fish.
sex	character varying(1)		1 digit sex code. 1=male, 2=female, 3=immature etc refer rdb.t_sex_codes.
gonad_wt	numeric(5,1)		Weight (grams) of the gonads.
gonad_stage	character varying(1)		1 digit code for the reproductive stage of gonad.
stomach_state	character varying(1)		1 digit code to describe the stomach fullness, refer rdb.t_stom_state_codes.
stomach_cond	character varying(1)		1 digit code for the digestion condition of the stomach, refer rdb.t_stom_cond_codes.

prey1	character varying(3)	3 character uppercase alphabetic species code for the first prey.
vol1	smallint	Percentage volume in the stomach taken by prey1.
prey2	character varying(3)	3 character uppercase alphabetic species code for the second prey.
vol2	smallint	Percentage volume in the stomach taken by prey2.
prey3	character varying(3)	3 character uppercase alphabetic species code for the third prey.
vol3	smallint	Percentage volume in the stomach taken by prey3.
prey4	character varying(3)	3 character uppercase alphabetic species code for the fourth prey.
vol4	smallint	Percentage volume in the stomach taken by prey4.
prey5	character varying(3)	3 character uppercase alphabetic species code for the fifth prey.
vol5	smallint	Percentage volume in the stomach taken by prey5.
comments	character varying(60)	

Indexes:

```
"pk_t_fish_bio" PRIMARY KEY, btree
(landing_no, species, stratum_no, cluster_no, fish_no)
```

Check constraints:

```
"t_fish_bio_vol1_check" CHECK (vol1 >= 0 AND vol1 <= 100)
"t_fish_bio_vol2_check" CHECK (vol2 >= 0 AND vol2 <= 100)
"t_fish_bio_vol3_check" CHECK (vol3 >= 0 AND vol3 <= 100)
"t_fish_bio_vol4_check" CHECK (vol4 >= 0 AND vol4 <= 100)
"t_fish_bio_vol5_check" CHECK (vol5 >= 0 AND vol5 <= 100)
```

Foreign-key constraints:

```
"fk_t_fish_bio_curr_spp" FOREIGN KEY (species)
REFERENCES rdb.curr_spp(code)
"fk_t_fish_bio_t_fish_meas_codes" FOREIGN KEY (measure_meth)
REFERENCES rdb.t_fish_meas_codes(fish_meas_code)
"fk_t_fish_bio_t_sex_codes" FOREIGN KEY (sex)
REFERENCES rdb.t_sex_codes(sex_code)
"fk_t_fish_bio_t_stom_state_codes" FOREIGN KEY (stomach_state)
REFERENCES rdb.t_stom_state_codes(stom_state_code)
"fk_t_fish_bio_t_stom_cond_" FOREIGN KEY (stomach_cond)
REFERENCES rdb.t_stom_cond_codes(stom_cond_code)
```

6 market business rules

6.1 Introduction to business rules

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

There are three recognised 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. Formula and Validation rules are implemented by referential constraints, range checks, and algorithms both in the database and during validation.

Validation rules may be part of the preloading checks on the data as opposed to constraints or checks imposed by the database. These rules sometimes state that a value should be within a certain range. All such rules containing the word 'should' are conducted by preloading software. The use of the word 'should' in relation to these validation checks means that a warning message is generated when a value falls outside this range and the data are then checked further in relation to this value.

6.2 Summary of rules

Landing details (t_landing)

landing_no	Landing number, must be unique.
vessel_name	Must have a value entered.
vessel_no	Must be an integer greater than zero.
port	Must have a value entered
shed	Fish processing shed name must have a value entered which is checked against a valid list.
trip_start_date	Must be a valid date.
trip_end_date	Must be a valid date.

Multiple column checks on trip dates:

The trip start date must not be greater than the trip end date.

min_depth	Minimum depth must be an integer greater than zero and should be within a reasonable range of 1 - 1300.
max_depth	Maximum depth must be an integer greater than zero and should be within a reasonable range of 1 - 1300.

Multiple column checks on depths.

The maximum depth must be greater than or equal to the minimum depth.

Multiple column checks on fishing method and depths.

The minimum depth and the maximum depth should be within a reasonable range for the fishing method. Reasonable depth ranges for the major methods are given in Appendix 2.

landing_date	The date the fish was landed must have a value entered and must be a valid date.
---------------------	--

Multiple column checks on landing and trip dates:

The date landed must be on or after to the trip end date.

target_spp	The target species code must be a valid code as listed in the <i>curr_spp</i> table in the rdb database.
species	Must have a value. Each species code in the list must be a valid code as listed in the <i>curr_spp</i> table in the rdb database.
landing_weight	The total landed weight of all species must be greater than zero and should be within a reasonable range of 10 - 250000.

fishing_method	The fishing method code must be a valid code as listed in the <i>meth_codes</i> table in the rdb database.
area	Each area code in the list must be a valid code as listed in area code tables in the rdb database, except for paua which from 1 October 1999 may also use the zones numbered from 1 up to 99 as documented in ‘PAUA LOGBOOK 1999 – 2000 Logbook, instructions, examples, maps and descriptions. Contract for New Zealand Paua Management Company, September 1999’ produced by NIWA.
proj_code	Must be a valid Ministry of Fisheries Project Code as contained in the master list supplied to the Data Manager.

Catch details (t_catch)

landing_no	Must be equal to a landing number in the <i>t_landing</i> table.
species	Must have a value and be a valid code as listed in the <i>curr_spp</i> table in the rdb database.
species_wt	The species weight must be a number greater than zero and should not exceed a reasonable limit of 250 000. Multiple column check on species and species_wt: The species weight should be within reasonable limits for the species. Reasonable weight ranges for the major species are listed in Appendix 2. Multiple column check on landing_wt, species_wt: Species weight must be less than or equal to the landing weight.
no_cluster_units	The number of cluster units landed must be an integer greater than zero and should be within a reasonable range of 1 – 7 500.

Stratum details (t_stratum)

Multiple column check on landing_no and species:

The combination of landing number and species must exist in the *t_catch* table.

species	Must have a value and be a valid code as listed in the <i>curr_spp</i> table in the rdb database.
stratum_no	Stratum number must have a value and be a unique integer within a landing and species.
stratum_wt	Stratum weight must be a number greater than zero and should not exceed a reasonable limit of 250000.

Multiple column check on species_wt and stratum_wt:

Stratum weight must be less than or equal to species weight for each landing_no, species combination.

Multiple column check on stratum_wt and species:

The stratum weight should be within reasonable limits for the species. Reasonable weight ranges for the major species are listed in Appendix 2.

no_bins_in_strat	Must be an integer greater than zero and should be within a reasonable range of 1 – 7500.
no_clusters_sampled	Must be an integer greater than zero and should be within a reasonable range of 1 - 69.

Multiple column check on no_bins_in_strat and no_clusters_sampled

The number of clusters sampled must be less than or equal to the number of clusters (or bins) in the stratum.

boxes_or_bins	Container code must be a valid code as listed in Appendix 2.
per_strat_samp	The percentage of the stratum sampled must be between 0 and 100.
area	The area code must be a valid code as listed in the <i>area_codes</i> table in the rdb database.
sampling method	Method of selecting the sample code which must be a valid code as listed in the <i>t_samp_sel_codes</i> table of the rdb database.
fish_samp_method	The code for the method of selecting fish from a cluster must either be a value of 1 or 2.
fish_meas_method	The fish measurement method code must be a valid code as listed in the <i>t_fish_meas_codes</i> of the rdb database.
stage_meth	Gonad staging method code must be a valid code as listed in the <i>t_gon_sys_desc</i> table of the rdb database.

gonads_staged	The code to indicate if the gonads were staged must be in the range 1 to 5, as listed in Appendix 2.
otoliths_kept	The code to indicate if otoliths or other material were collected for ageing analysis must be in the range of 1 to 2.
sample_wt	The sample weight must be a number greater than zero, and should not exceed a reasonable limit of 1500.
	Multiple column check on stratum_wt and sample_wt. Sample weight must be less than or equal to the stratum weight.
no_fish_meas	The number of fish measured must be an integer greater than zero.

Cluster unit details (t_cluster)

Multiple column check on landing number, species and stratum number:

The combination of *landing_no*, *catch_spp* and *stratum_no* must exist in the *t_stratum* table as *landing_no*, *species* and *stratum_no*.

catch_spp Must be a valid code as listed in the *curr_spp* table in the **rdb** database.

species Must be a valid code as listed in the *curr_spp* table in the **rdb** database.

cluster_no Cluster number must have a value and be a unique integer within a stratum for each species.

cluster_wt Cluster weight must be a number greater than zero.

Multiple column check on cluster_wt and boxes_or_bins:

The cluster weight should be within a reasonable range for the type of cluster unit where clusters are recorded separately. Reasonable weight limits for the common cluster units are listed in Appendix 2.

percent_samp Percentage of the cluster sampled must be between 0 and 100.

Length frequencies (t_length)

Multiple column check on landing number, species, stratum number and cluster number:

The combination of *landing_no*, *species*, *stratum_no* and *cluster_no* must exist in the *t_cluster* table.

species Must be a valid code as listed in the *curr_spp* table in the **rdb** database.

lgth The fish length must be an integer greater than zero and should be within the reasonable range of 5 - 200.

Multiple column check on species and length:

The fish length should be less than the maximum-recorded fish length for the species as recorded in the *curr_spp* table in the **rdb** database.

no_m The number of male fish must be an integer greater than or equal to zero.

no_f The number of female fish must be an integer greater than or equal to zero.

no_t The total number of fish must be an integer greater than zero.

no_m1 – no_m5 Must be an integer greater than or equal to zero.

no_f1 – no_f5 Must be an integer greater than or equal to zero.

Multiple column checks on number of males and number of males of each gonad stage.

The total number of males must be greater than or equal to the sum of the number of males of each stage.

Multiple column checks on number of females and number of females of each gonad stage.

The number of females must be greater than or equal to the sum of the number of females of each stage.

Multiple column check on number of males, females and total fish:

The total number of fish must be greater than or equal to the number of males plus the number of females.

Fish biological details (t_fish_bio)

Multiple columns check on landing number, stratum number, species, and cluster number:

The combination of *landing_no*, *stratum_no*, *species*, and *cluster_no* must exist in the *t_cluster* table unless *cluster_no* = -1 then the combination of *landing_no*, *stratum_no*, *species* must exist in the *t_stratum* table.

fish_no	Must be a unique number within each cluster.
species	Must be a valid species code as listed in the <i>curr_spp</i> table in the rdb database.
measure_meth	The fish measurement method code must be a valid code as listed in the <i>t_fish_meas_codes</i> table of the rdb database.
measure_meth2	The second fish measurement method code must be a valid code as listed in the <i>t_fish_meas_codes</i> table of the rdb database.
lgth the	The fish length must be an integer greater than zero and should be within the reasonable range of 5 - 200
lgth2	The second fish length must be an integer greater than zero and should be within the reasonable range of 5 - 200

Multiple columns check on species and length:

The fish length should be less than the maximum-recorded fish length for the species as recorded in the *curr_spp* table in the **rdb** database.

weight	The weight of the fish must be a number greater than zero.
---------------	--

Multiple columns check on species and weight:

The fish weight should be less than a reasonable maximum fish weight for the species. Reasonable maximum fish weights for the major species are given in Appendix 2.

sex	Must be a valid sex code as listed in the <i>t_sex_codes</i> table in the rdb database.
gonad_wt	The gonad weight must be a number greater than zero.
gonad_stage	Multiple column check on species, gonad stage, sex, and <i>t_stratum.stage_meth</i>: Must be a valid gonad stage for the species, sex, and gonad staging method code as listed in the <i>t_gon_stg_meth</i> table in the rdb database. See also Appendix 2.

- stomach_state** Must be a valid stomach state code as listed in the *t_stom_state_codes* table in the **rdb** database.
- stomach_cond** Must be a valid stomach condition code as listed in the *t_stom_cond_codes* table in the **rdb** database.
- prey1 – prey5** Must be a valid species code as listed in the *curr_spp* table in the **rdb** database.
- vol1 – vol5** Must be a valid percentage within the range 0 – 100.

Multiple columns checks on prey volumes:
The sum of vol1 – vol5 must equal 100.

Appendix 1 – Data Integrity

Data collection and data processing.

Data are collected on the standardised forms, which are shown in Appendix 3. All samples should have a Stock Monitoring Programme Landing Record (2000 Edition) form filled out. Length data are recorded on a choice of one of the following Stock Monitoring Programme forms:

- Length Frequency and Gonad Staging (2000 Edition) for sexed length frequencies with optional gonad staging currently used for HOK, etc.
- Snapper Length Frequency (2001 Edition) for unsexed length frequencies, currently used for SNA and TRE.
- Length Weight (2000 Edition) used for freshwater eels.

If any new market sampling is proposed, it should be discussed with the market data administrator, or one of the authors of this report. Most sampling can be recorded on the existing forms, but new forms can be designed in consultation with the market data administrator and fisheries Data Management staff, at NIWA, Greta Point.

Data entry, process, and definitions.

This section outlines the flow of paper recorded data, for market sampling data from field collection through to its availability to researchers for stock assessment analyses, and defines the separate tasks that are required to do this.

In this example, samplers working in fish processing facilities on shore collect hand written data. These data are recorded on paper and written in pencil. Each sample is discrete, and is given a unique landing number that is later linked to each fish or length class measured in that sample.

At the completion of each sample the sampler/recorder ensures that all pages are in order and sequentially numbered and that all required data fields have been correctly filled. The data are then forwarded to a project team member (the client) responsible for checking the data prior to keypunching.

There are 5 clear steps in the data flow following its collection. These are listed and then discussed individually in detail.

1. Pre-key punching, checking and batching.
2. Key punching data entry.
3. Electronic transfer of raw data flat files in disk and paper copy to client (i.e., project team).
4. Data error checking (manual and computer), validation, and grooming.
5. "Groomed", validated data loaded to database. Now available for analysis.

1. Pre-key punching, visual checking and batching:

The paper forms from each sample are checked for obvious gross errors or omissions and corrected if necessary. Samples are placed with other samples of the same species into batches and allocated a unique file name; e.g., lryyspp1, where lr =landing record, yy = year e.g. 98, spp = species code (e.g. hok for hoki), and 1 is a sequential number for that species in that year. The batches of raw data are sent for keypunching.

2. Key punching data entry:

At keypunching, the batches of raw data are digitised and verified by trained data entry operators. Verification simply means that the data are digitised twice and the two resulting files are crosschecked for mismatches. Operator errors are corrected at this point, and the completed digitised file is ready for transfer for the error checking process. At no point in this process are changes or interpretations made to the raw data. NIWA uses the KEYS Data Emulator for data entry.

3. Electronic transfer of raw data flat files in disk and paper copy to client:

The digitised data file is transferred for error checking, along with the original raw data file. At this point the data are now in a format that is compatible with the data processing routines.

4. Data error checking, validation, and grooming:

Data files are put through a number of computer error checking (validation) routines that look for inaccuracies and inconsistencies within samples. Any errors detected are corrected. Data are then passed through these error-checking routines until the data reach a satisfactory standard that will allow them to be inserted in the appropriate database.

In some instances, data may be inserted into “working tables” in a database. This is often done to check the integrity of the data by taking advantage of relational databases ability to manipulate, match, and compare related sets of data.

5. "Groomed", validated data loaded to database. Available for analysis:

The clean, groomed, and validated data are inserted into the appropriate database and now become available for analysis.

The clean digitised data files and raw paper data are then archived for safekeeping.

Appendix 2 – Reference code tables

Container capacity codes (boxes_or_bins)

1	boxes (capacity approx. 20 – 50 kg)
2	bins (capacity approx. 250 kg – 1 t)
3	other (please specify)
4	trays (capacity approx less than or equal to 20 kg)
5	wire cage (capacity approx 400kg)
6	sacks
7	strings

NB Codes 4, 6 & 7 were implemented on the 1st of January 2000.

“Were the gonads staged ?” codes (gonads_staged)

1	Yes
2	No
3	Males only staged
4	Females only staged
5	Both sexes; i.e. all staged

NB Codes 3, 4 & 5 were implemented on the 1st of January 2000.

Gonad staging codes (stage_meth = ‘SM’)

Males

1	Immature/Resting
2	Maturing
3	Mature/Ripe
4	Running ripe
5	Spent

Females

1	Immature/Resting (Ovary translucent or pink, with no eggs visible)
2	Maturing (Eggs visible, opaque/coloured, but not hyaline)
3	Mature/Ripe (hyaline (clear) eggs present)
4	Running Ripe (Large clear eggs flow freely)
5	Spent (Ovary flaccid and bloody)

Unsexed or immature

1	Immature (unable to determine sex)
---	------------------------------------

This staging method (SM) is currently used for all species except eels (SFE & LFE).

Reasonable depth ranges for *min_depth* and *max_depth* by fishing method.

Fishing method	Code(s)	Reasonable depth range (metres)
Trawling	01, 06, 09	10 - 1300
Pair Trawling	07	10 - 100
Danish Seine	16, 17	10 - 200
Dredge	21	10 - 100
Bottom Long Line	31	10 - 600
Drop / Dahn Line	35	30 - 700
Diving	71	1 - 20

Reasonable ranges for species weight and stratum weight by species.

species code	minimum weight	maximum weight
BNS	90	45 000
BYS	100	70 000
EMA	10000	250 000
EPT	200	120 000
GMU	30	1 000
GUR	300	2 500
HOK	700	230 000
JMA	2000	300 000
KAH	50	250 000
LFE	1	1 500
ORH	200	200 000
PAI	10	2 000
PAU	10	2 000
RBV	50	60 000
RCO	200	40 000
SFE	1	3 000
SKI	90	75 000
SNA	5	50 000
TAR	60	20 000
TRE	20	80 000

Approximate reasonable limits for cluster weight by container code.

Container code weight range kg

Boxes	1	20 – 50
Bins	2	250 – 1000
Trays	4	<= 20

NB. Container (capacity) code is stored in the attribute *boxes_or_bins*. These codes are documented on the bottom of the Landing Record form as Container capacity code, see Appendix 3.

Reasonable maximum fish weights (grams)

BNS	20 000
JMD	2 000
JMM	2 400
JMN	800
LFE	4 000
SFE	2 500
ORH	3 000
RBV	2 500
RCO	3 700
SKI	12 000
SNA	10 000
TAR	3 000
TRE	4 000

Reference code lists from the rdb database as used in the market database.

Codes from table t_fish_meas_codes

fish_meas_code	description
1	Fork Length
2	Total Length
3	Standard Length
4	Mantle Length (squid)
5	Pelvic Length (rays)
6	Carapace Width
7	Shell Height
8	Shell Length - maximum dimension of the shellfish (except oysters)
A	Snout-Anus length, from tip of the snout to the anus, eg for Macrourus spp.
P	Basal or aperture length being the longest measurement along the anterior-posterior axis of the base of the shell. (Paua)
S	Length from anterior margin of dorsal fin origin to posterior margin of caudal fin. (STA)

These codes are used in table t_stratum for attribute fish_meas_method and table t_fish_bio attributes measure_meth and measure_meth2.

Codes from table t_samp_sel_codes

samp_sel_code	description
1	Rigorous simple random sample (SRS)
4	Stratified sample
5	Approx. simple random sample
8	Non-random, not representative of whole catch
9	Whole catch

These codes are used in table t_stratum for attribute sampling_method. Typically only codes 5 and 9 are used, or 4 for stratified samples.

Codes from table t_stom_state_codes and t_stom_cond_codes

stom_state_code	descriptn	stom_cond_code	description
0	Empty		
1	Trace	1	Fresh
2	Part full (1/4-3/4)	2	1/2 digested
3	Full	3	Digested
		4	Mixed digestion states
8	Regurgitated/Eviscerated		
9	Everted		

These codes are used in table t_fish_bio attributes stomach_state and stomach_cond.

