Database documentation for the Ministry for Primary Industries Fisheries research trawl survey database

trawl

K. A. Mackay

NIWA Fisheries Data Management
Database Documentation Series

Revised September 2020

Revision History

Version	Change	Date	Responsible
1.0	Initial release as MAF Fisheries Internal Report No. 209.	1993	Kevin Mackay
1.1	Updated as NIWA Internal Report No. 16. Included addition of Other data section & tables t_subcatch and t_lgth_stage.	1998	Kevin Mackay
1.2	Added Business Rules and Appendix 1 Reference code tables, as NIWA Internal Report No. 73	2000	Kevin Mackay
1.3	Added t_stratum_defn, t_trip_stratum tables and associated ERD and descriptions. Added paragraph re potting surveys to section 2.1.2	20 Jun 2006	Fred Wei David Fisher
1.4	Minor corrections to business rules, for t_lgth.no_a. Added no_m9 & no_f9 to this document in table t_lgth_stage. These were added to the database in ~2004.	15 Jan 2007 2 Feb 2007	David Fisher
1.5	Added biomass_flag in t_station.	10 Jul 2007	Fred Wei
1.6	Added more business rules	Oct 2007	David Fisher
1.7	Added reference codes from rdb to Appendix 1, for short code lists	Nov 2010	David Fisher
1.8	Refined business rules re positions, dates and times.	Jan 2011	David Fisher
1.9	Added species codes section 2.3	April 2012	David Fisher
1.10	Added table t_site	July 2013	David Fisher
1.11	Added codes for stomach state and condition to section 5 t_fish_bio, and reference to marlin for curr_spp.	26 Mar2014	David Fisher
2.0	Postgres version	Jan 2015	D Fisher, F Wei
2.1	Changed t_catch and t_subcatch weights to numeric(9,3) (i.e., kg to 3 d.p.; was 1 d.p.). Added comments to t_catch and t_subcatch.	7 Jun 2019 12 Jun 2019	J Yeoman
2.2	Changed t_fish_bio.comments to text character varying(10)), and t_station.secchi to numeric(4,1) (was smallint)	9 Aug 2019	J Yeoman
2.3	Added min_depth & max_depth to t_stratum	2 Sep 2019	D Fisher
2.4	Changed t_stratum.descrptn from character varying(256) to text data type	30 Jul 2020	J Yeoman

Contents

List of Figures

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 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 provides an introduction to the trawl survey database **trawl**, and is a part of the database documentation series produced by NIWA. It supersedes the previous documentation by Mackay (1998) on this 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 fit in together, and their relationships to other databases.

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

2 Trawl survey database

2.1 Data sources

2.1.1 Trawl survey data

The **trawl** database is the major fisheries research database. It results from data collected by research trawl surveys on research vessels and chartered commercial fishing vessels.

Trawl surveys are a major tool used by research scientists for stock assessment. They are used to estimate basic parameters of commercial fish populations, including biomass, sex ratio, and the proportion of sexually mature fish, and the distribution of ages and lengths in the population. These parameters may be used in estimating mortality and growth rates.

The method for estimating the parameters from a trawl survey has been well documented in other publications (Francis 1981, 1984) and can be described in four basic steps:

1. The geographical area to be surveyed is defined and area calculated.
2. A number of points are picked at random within the survey area.
3. At each random location a trawl[1] is carried out and the catch rate is calculated.
4. The estimated biomass is calculated as the average catch rate multiplied by the area[2].

[1] Also known as a ‘station’.

[2] The trawl survey analysis program for biomass calculations is available on NIWA’s **neptune** computer.

The above four-step procedure is refined to take into account knowledge about where fish are most likely to be found. Dividing the survey area into sub-areas (called strata) does this so that known areas of low fish density are in different strata from areas of high density. A higher density of trawls is then allocated to strata where high catch rates are expected. The four-step procedures are then carried out separately for each stratum.

In addition to stratification, a further refinement is added to trawl surveys in the form of a two-phase design. In these surveys the catch rate information gathered in the first phase are used to allocate additional trawls to strata, which were found to have been under-sampled.

Sometimes trawl surveys are carried out by fishing at positions on a regularly spaced grid rather than at random locations. This may be done because there can be logistical gains in efficiency from having the same distance between consecutive trawls. This normally would ensure the survey is representative of the area.

The whole catch for each trawl is sorted by species, and individual species weights and a total weight are calculated.

For certain species from the catch (depending on the objectives of the trawl survey), fish are taken as a sample for further measurements. The amount of fish depends on the measurements to be taken. Ideally, all fish of any one species are measured for a length frequency, but for larger catches approximately 200 fish suffice. Length frequency measurements require the length and sex to be recorded for each fish.

Further biological examination may require up to another 20 fish. This examination at the least determines for each fish the sexual maturity of the fish (allocating a stage number to the gonad). A more detailed analysis includes determining individual fish weight, gonad weight, and the condition of the stomach and contents. These biological analyses are only taken on the most important of the target species.

In some instances, the whole catch can be divided in to subcatches for length frequency and biological analysis. For example, in a large catch, comparisons may be needed between the size ranges of fish caught at the beginning of the catch to those caught at the end. Another common case for multiple subcatches is where there are two distinctive size classes for one species. A subcatch is taken from each size class. A third case for subcatches is where the trawl gear has multiple codends, as with scampi trawls, so each codend will produce a subcatch of a species and the sum of all the codends will produce the whole catch.

2.1.2 Other types of data

While trawl survey data constitutes the bulk of the data held in **trawl**, it by no means represents all the data. The database design allows for any data to be stored from a trip that has one or more stations that deploy some sort of gear. Examples of such data include camera equipment, CTD probes, plankton nets, handlines and pots. For the most part, data from gear deployment other than trawling gear gets included into **trawl** if it is a part of a trawl survey.

Data collected from potting surveys such blue cod potting surveys are also included in **trawl**. The design of these potting surveys is similar to trawl surveys where the survey area may be divided into strata and a pot set and subsequent pot lift are represented as a station. These surveys typically have a group of pots at each location or site, but each individual pot set / lift should be treated the same as a trawl station and assigned a unique station number. The same concepts of catch, length frequencies and biological analyses apply to potting surveys as referred to for trawl surveys above.

The advantage of such a generic database design is that it allows for other surveys to use the trawl survey analysis applications, such as the biomass and scaled length frequency tool.

2.1.3 Soviet Trawl Survey Data

In 2001, the Ministry of Fisheries acquired Soviet trawl survey data from the New Zealand region collected from 52 trawl surveys covering the period from 1964 to 1987. Nothing is known about the sampling strategy employed by the Soviets during these surveys. No stratum information was given, so they are assumed to be non-stratified. These data are of dubious quality and were collected to unknown standards, and hence are held separate from all other trawl survey data.

2.2 Trip, cruise, or voyage?

Over the years, trawl surveys have been labeled many things. In the last few years research surveys have been called “trips”, “cruises” or “voyages”, but all represent the same thing.

As a consequence, while the trawl database labels all trawl surveys and associated tables with the word “trip”, the words “cruise” or “voyage” can just as easily be substituted.

2.3 Species codes

The term ‘species’ in this document generally refers to the three letter species codes as referenced in rdb.species_master, or the taxa this code represents. While most of these codes represent a single species, these species codes do not all refer to the taxonomic level of species, with some codes representing multiple species, other levels of taxa typically genus, or family, or occasionally inorganic material such as rocks or anthropogenic material such as various classifications of rubbish, etc.

2.4 Data validation

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

This process includes instructions for data recording[3], simple data validation using the **checkq**[4] validation program language, followed by loading of data into a loading database, and more stringent error checking with Empress C routines[5]. Note that all trawl survey data collected from RV Tangaroa and more recently RV Kaharoa have been collected using an on-line data acquisition system that collects, checks, and loads

[3] Currently located at https://one.niwa.co.nz/display/FMDS/trawl_instructions

[4] See local Unix manual page on **checkq**

[5] Marine Research Computing: Trawl survey data entry. *User Note 10.*

data directly into a loading database.

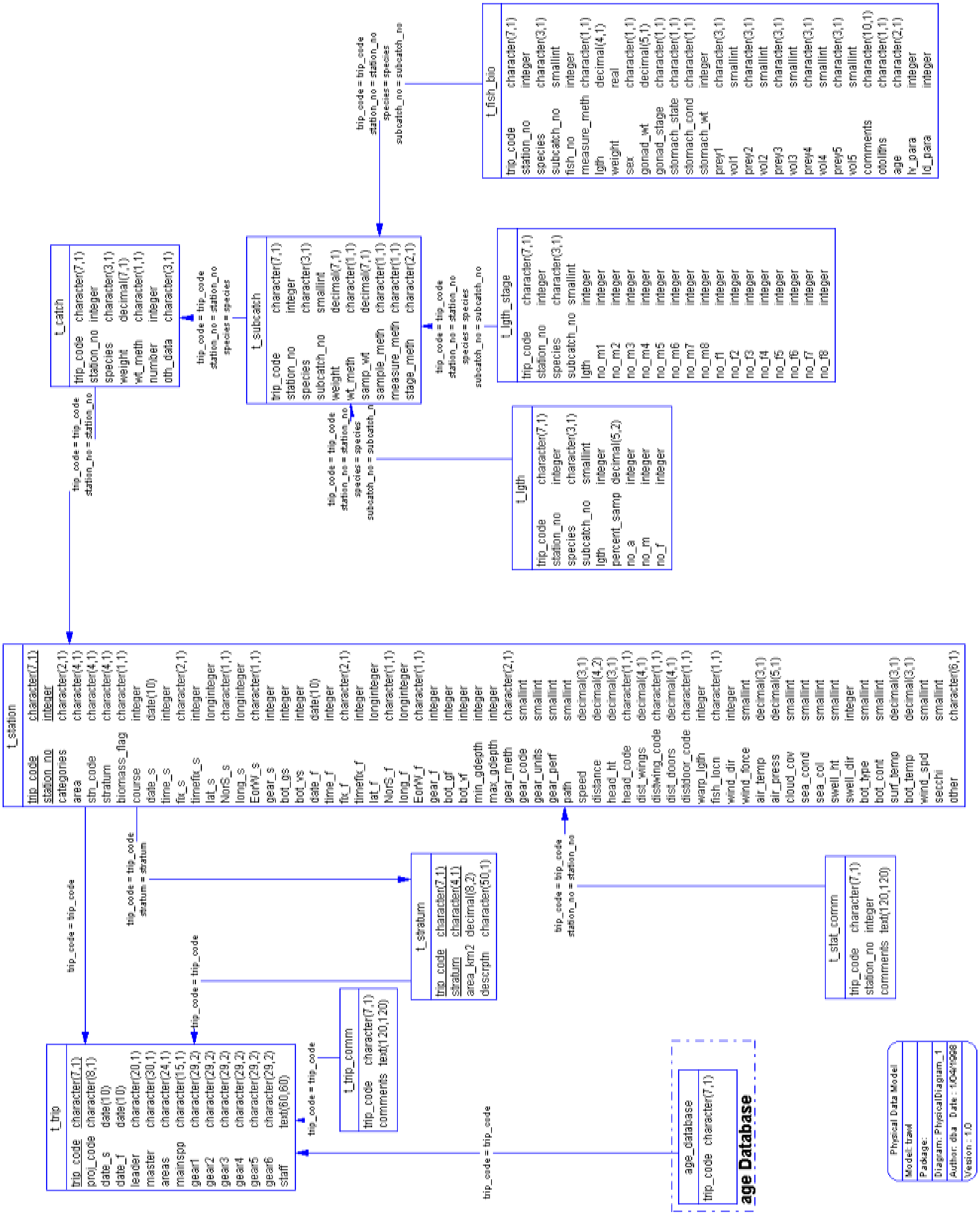


Figure 1: Entity Relationship Diagram (ERD) of the trawl database.

3 Data structures

3.1 Introduction

One of the primary influences on the trawl database design is the ability to scale length frequency data up to the whole catch. The following structures achieve this by creating a table for each tier of the sampling strategy.

3.2 Database description

This database contains several tables. The ERD for **trawl** (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 and tables in other databases. All of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary key[6]. 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 has been selected to be represented in the database. Each *attribute* of a table is a defining property or quality of the table.

Note that Figure 1 shows the main tables only. Most of the tables in the **trawl** database have some attributes, called foreign keys[7], which contain standard NIWA fisheries codes, such as *species* and *meth_codes*. These attributes provide links to the **rdb** (research database) database, which contains the definitive list of standard codes. Therefore, an expanded ERD for these tables will follow (Figures 2 - 5).

Section 5 shows a listing of all the **trawl** tables as implemented by the Postgres DBMS. As can be seen in the listing of the tables, most tables have a primary key. Primary keys are generally listed using the format:

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

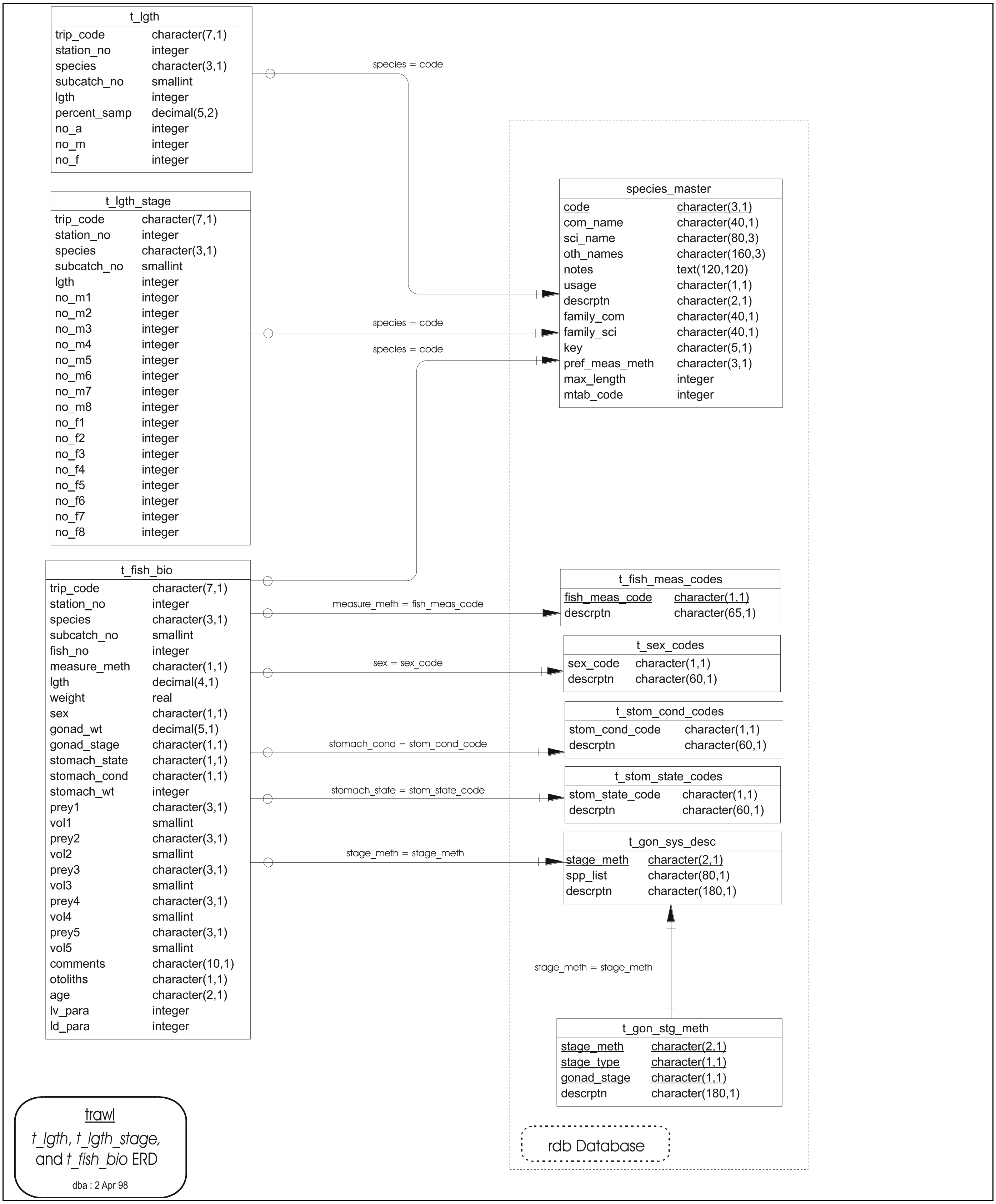
where the attribute(s) make up the primary key and the index name is the primary key name. The primary key forms a unique index and prevents records with duplicate key values from being inserted into the table, e.g., a trip with an existing trip code.

As reflected by the ERD, the highest level of a trawl survey is a research trip. Details for each trip are held in the table *t_trip* (Table 1). Each trip is uniquely identified by a trip code, stored as the attribute *trip_code*.

Comments for a trip are held in a separate table *t_trip_comm* (Table 2), but have the same attribute, trip_code, to identify records as *t_trip*. This means that one trip may have one or more than one comment associated with it, but it is also possible to have none at all.

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

[7] 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.



t_lgth	
trip_code	character(7,1)
station_no	integer
species	character(3,1)
subcatch_no	smallint
lgth	integer
percent_samp	decimal(5,2)
no_a	integer
no_m	integer
no_f	integer

t_lgth_stage	
trip_code	character(7,1)
station_no	integer
species	character(3,1)
subcatch_no	smallint
lgth	integer
no_m1	integer
no_m2	integer
no_m3	integer
no_m4	integer
no_m5	integer
no_m6	integer
no_m7	integer
no_m8	integer
no_f1	integer
no_f2	integer
no_f3	integer
no_f4	integer
no_f5	integer
no_f6	integer
no_f7	integer
no_f8	integer

t_fish_bio	
trip_code	character(7,1)
station_no	integer
species	character(3,1)
subcatch_no	smallint
fish_no	integer
measure_meth	character(1,1)
lgth	decimal(4,1)
weight	real
sex	character(1,1)
gonad_wt	decimal(5,1)
gonad_stage	character(1,1)
stomach_state	character(1,1)
stomach_cond	character(1,1)
stomach_wt	integer
prey1	character(3,1)
vol1	smallint
prey2	character(3,1)
vol2	smallint
prey3	character(3,1)
vol3	smallint
prey4	character(3,1)
vol4	smallint
prey5	character(3,1)
vol5	smallint
comments	character(10,1)
otoliths	character(1,1)
age	character(2,1)
lv_para	integer
ld_para	integer

species_master	
code	character(3,1)
com_name	character(40,1)
sci_name	character(80,3)
oth_names	character(160,3)
notes	text(120,120)
usage	character(1,1)
descrptn	character(2,1)
family_com	character(40,1)
family_sci	character(40,1)
key	character(5,1)
pref_meas_meth	character(3,1)
max_length	integer
mtab_code	integer

t_fish_meas_codes	
fish_meas_code	character(1,1)
descrptn	character(65,1)

t_sex_codes	
sex_code	character(1,1)
descrptn	character(60,1)

t_stom_cond_codes	
stom_cond_code	character(1,1)
descrptn	character(60,1)

t_stom_state_codes	
stom_state_code	character(1,1)
descrptn	character(60,1)

t_gon_sys_desc	
stage_meth	character(2,1)
spp_list	character(80,1)
descrptn	character(180,1)

t_gon_stg_meth	
stage_meth	character(2,1)
stage_type	character(1,1)
gonad_stage	character(1,1)
descrptn	character(180,1)

Figure 2: ERD showing the relationships between *t_station* and the master code tables in the rdb database.

The fundamental relationship between tables that is repeated throughout the database is the *one-to-many* relationship[8]. This is shown in the ERD by connecting a single line (indicating 'many') from the child table (e.g., *t_trip_comm*) to the parent table (e.g., *t_trip*) with an arrow-head (indicating 'one') pointing to the parent.

Every relationship has a mandatory or optional aspect to it. That is, if a relationship is mandatory, then it has to occur and least once, while an optional relationship might not occur at all. For example, in Figure 1, consider that relationship between the table *t_trip* and it's child table *t_trip_comm*.

The symbol "O" by the child *t_trip_comm* means that a trip record can have zero or many trip comments, while the bar by the parent *t_trip* means that for every trip comment there must be a matching trip record.

For stratified trawl surveys, stratum details, such as stratum code and area (in square kilometres) are stored in the table *t_stratum* (Table 3). Notice that there is an optional link from *t_trip* to *t_stratum*; this means that not all trips have to have strata, i.e., unstratified trawl surveys.

Any one trip also relates to many stations. This is a mandatory relationship: a trip has to have at least one station before it can be entered into the database. Generally, a station is the location at which the trawl gear was towed. Details for the station, such as start and finish location, time, depth, gear performance and environment parameters are stored in the table *t_station* (Table 4). Many of the attributes in this table represent codes to explain how other attributes where derived and what methods were used. As shown in Figure 2, each code is a foreign key to a table in the **rdb** database that provides an explanation for the code used.

Note that a station may or may not occur within a stratum (*t_station* contains the attribute *stratum*) and that one stratum may or may not contain stations. Therefore, there is a two-way optional many-to-one relationship between *t_station* and *t_stratum*.

Like the table *t_trip*, *t_station* has its own comments table *t_stat_comm* (Table 5).

Each station in a trawl survey may produce a catch of several species of fish, and other organisms. A catch from any one station is broken down into the different species, with each species being an individual record in the table *t_catch* (Table 6). Each record contains the species code, catch weight and other flags to indicate if a sample was taken for further measurement. The attributes *species* and *wt_meth* are codes that are foreign keys to tables in the **rdb** database (Figure 3) that provides explanations for the codes used. Not every station will produce a catch of fish, so again there is an optional one-to-many relationship between *t_station* and *t_catch*.

To cater for the instances where there are subcatches, the table *t_subcatch* (Table 7) stores information including subcatch weight, the method by which fish were selected for sampling from the subcatch, the weight of the fish used for sampling, and the fish measurement method used. Each subcatch for a given trip, station and species is identified by the attribute *subcatch_no*.

[8] A one-to-many relationship is where one record in a table (the *parent*) relates to one or many records in another table (the *child*).

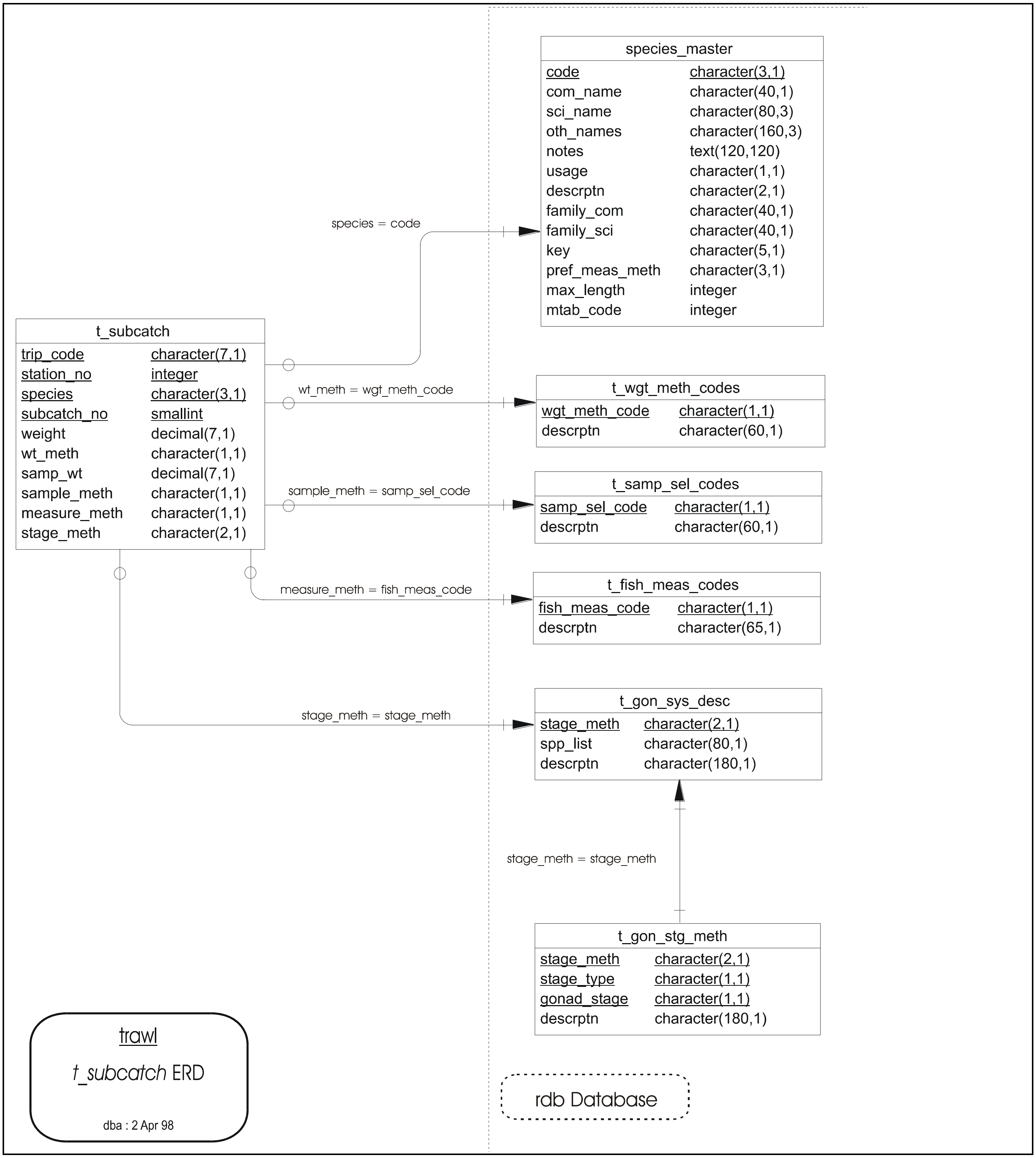


Figure 3: ERD showing the relationships between *t_catch* and the master code tables in the rdb database.

Note that when subcatches are not used, the whole catch becomes one subcatch, and the attribute *subcatch_no* is equal to 1. Therefore, every record in *t_catch* has a one-to-many relationship to *t_subcatch*.

From a subcatch, a sample of fish may be taken for length frequency measurements. Length frequency data are stored in the table *t_lgth* (Table 8). Length class is stored at record level in this table, not individual fish. For a length class, the number of males, females, and total fish is stored. Note that the attribute *percent_samp* stores the percent of the <u>subcatch</u> that was sampled for length frequency, not the percent of the whole catch of the species.

Some catches may be subdivided into subcatches. Subcatches may be distinguished by the attribute *subcatch_no*. For example, consider the scenario of a catch with two distinct size classes - a few large adults, and the remainder juveniles.

All the adults can then represent subcatch 1 and the juveniles represent subcatch 2. In this scenario, all the adults are measured for a length frequency giving a percent sampled of 100% of subcatch 1. While only half the juveniles were measured, giving a percent sampled of 50% of subcatch 2.

For relevant species a length frequency is required by gonad stage. This is especially necessary for pre-spawning and spawning trawl surveys. These length frequency data are held in the table *t_lgth_stage*. This is basically an extended version of *t_lgth* with counts of each gonad stage for males and females recorded for each length class. The 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 attribute *stage_meth* and relates to a full description as recorded in the *t_gon_stg_meth* and *t_gon_sys_desc* tables in the **rdb** database (Figures 4 & 5).

For the most part, the staging is carried out on the gonads of males and/or females. The exception to this is for scampi, a deepwater lobster. Scampi, like most crustacea, produce eggs in the ovaries, but store them under the tail while the eggs develop. Each female is apportioned two codes, the gonad code and the egg development code. This exception has resulted in scampi having their own view on the t_lgth_stage table, v_scampi, which caters for these differences.

In addition, some of the main species in a survey, up to 20 fish are randomly selected from the whole catch for a more detailed biological analysis.

Biological data are stored in the table *t_fish_bio* (Table 10). Records within this table contain information for individual fish, including fish weight, gonad stage and weight, stomach contents and condition. Each fish within this table is assigned a sequential *fish_no*. This attribute is combined with *trip_code*, *station*, and *species* to produce the primary key for this table.

Three views extend from this table. Each view is a ‘window’ into the records of *t_fish_bio* for a particular species only. The views *HOK_bio*, *ORH_bio*, and *SNA_bio* access data for the species hoki, orange roughy and snapper respectively. Note that these three views represent subsets of the t_fish_bio table and are not entities in their own right. Therefore, they are not shown on the ERD.

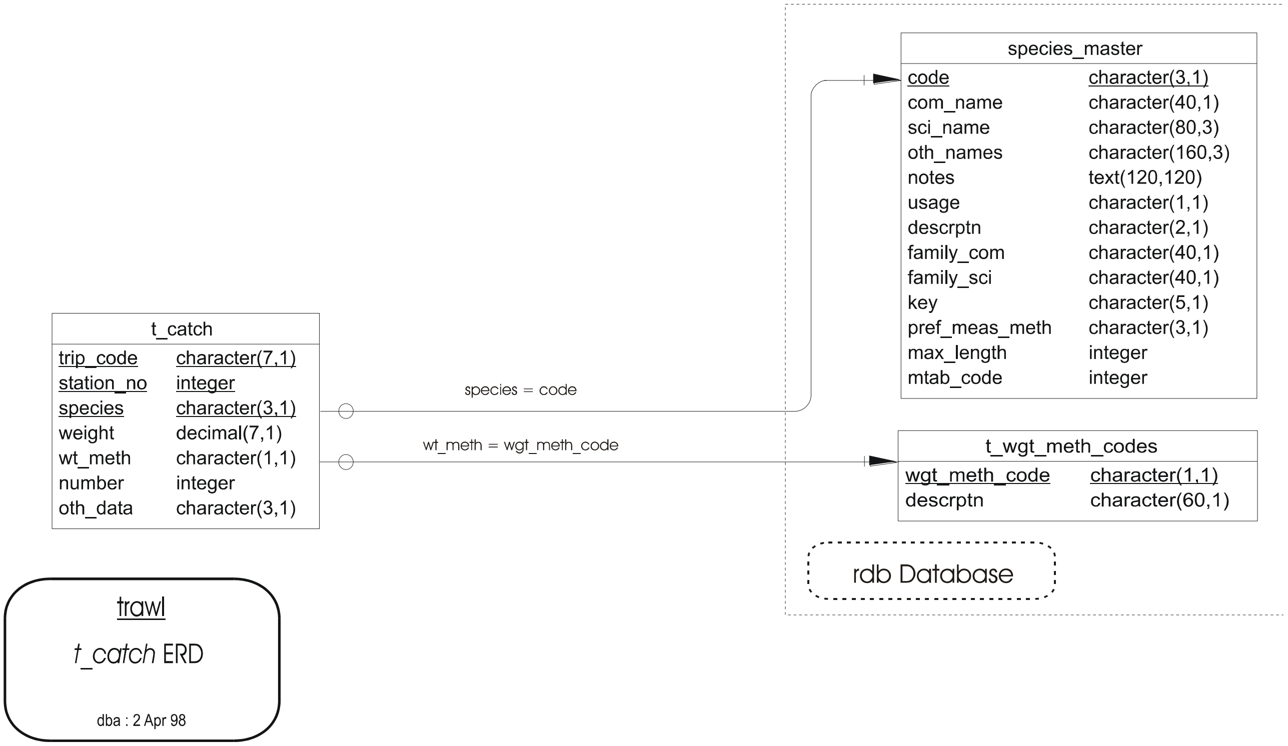


Figure 4: ERD showing the relationships between *t_subcatch* and the master code tables in the rdb database.

These last five tables (*t_catch*, *t_subcatch*, *t_lgth*, *t_lgth_stage*, and *t_fish_bio*) contain foreign keys, which link these tables to tables in the **rdb** database (Figure 5). Links to the **rdb** database are enforced by foreign key constraints[9]. 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 that 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 : constraint name FOREIGN KEY (attribute) REFERENCES parent
                          table (attribute)
```

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

```
Foreign-key constraints:
   "fk_t_trip_comm_t_trip_1" FOREIGN KEY (trip_code) REFERENCES
                             trawl.t_trip(trip_code)
```

This means that the value of the attribute *trip_code* in a *t_trip_comm* record must already exist in the parent table *t_trip* or the record will be rejected and the error message "invalid trip code" will be displayed.

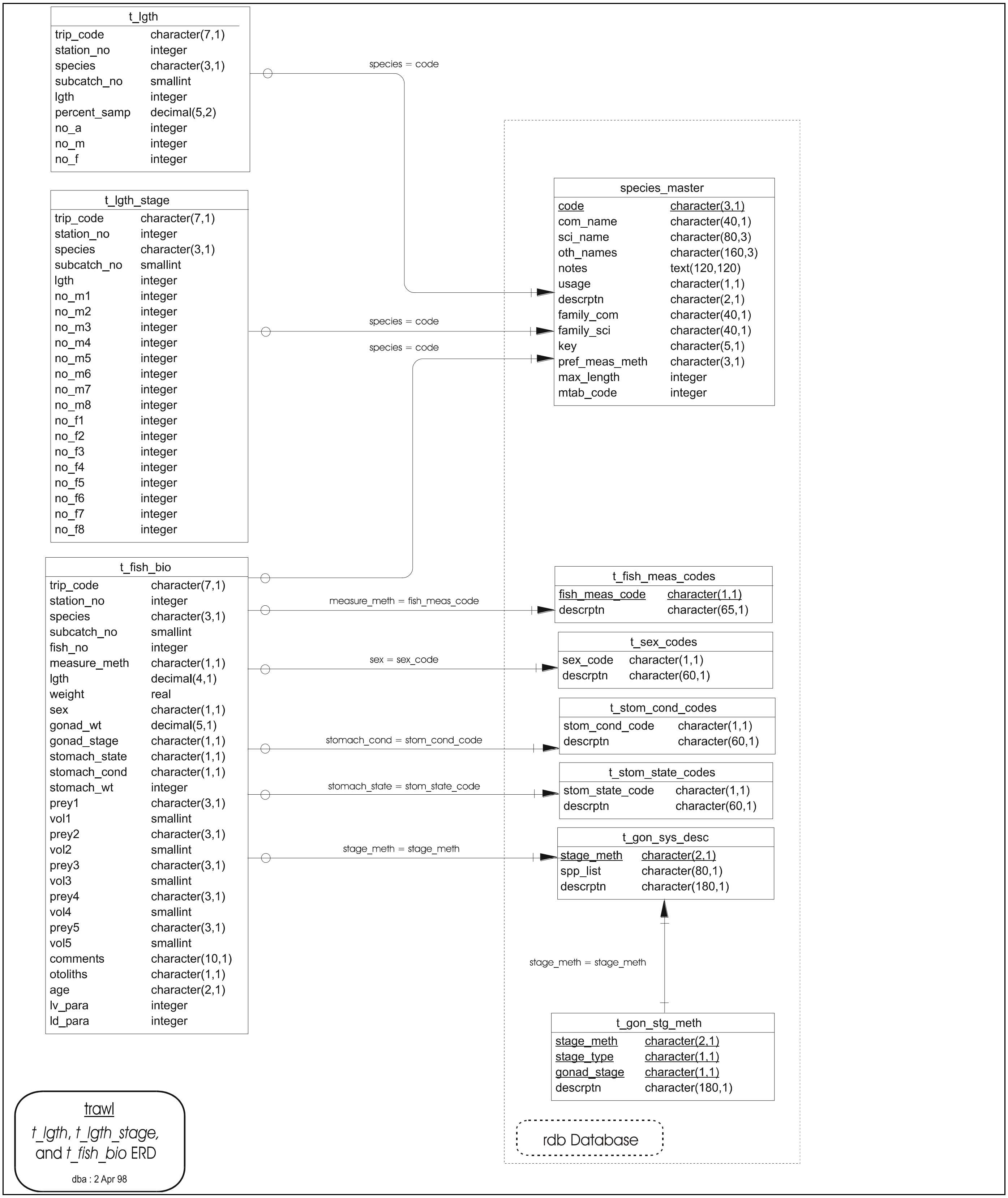
All tables in this database are indexed. That is, attributes that are most likely to be used as a searching

[9] Also known as integrity checks.

key have like values linked together to speed up searches. These indices are listed using the following format:

```
Indices:    index_name btree (attribute[, attribute])
```

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



t_lgth	
trip_code	character(7,1)
station_no	integer
species	character(3,1)
subcatch_no	smallint
lgth	integer
percent_samp	decimal(5,2)
no_a	integer
no_m	integer
no_f	integer

t_lgth_stage	
trip_code	character(7,1)
station_no	integer
species	character(3,1)
subcatch_no	smallint
lgth	integer
no_m1	integer
no_m2	integer
no_m3	integer
no_m4	integer
no_m5	integer
no_m6	integer
no_m7	integer
no_m8	integer
no_f1	integer
no_f2	integer
no_f3	integer
no_f4	integer
no_f5	integer
no_f6	integer
no_f7	integer
no_f8	integer

t_fish_bio	
trip_code	character(7,1)
station_no	integer
species	character(3,1)
subcatch_no	smallint
fish_no	integer
measure_meth	character(1,1)
lgth	decimal(4,1)
weight	real
sex	character(1,1)
gonad_wt	decimal(5,1)
gonad_stage	character(1,1)
stomach_state	character(1,1)
stomach_cond	character(1,1)
stomach_wt	integer
prey1	character(3,1)
vol1	smallint
prey2	character(3,1)
vol2	smallint
prey3	character(3,1)
vol3	smallint
prey4	character(3,1)
vol4	smallint
prey5	character(3,1)
vol5	smallint
comments	character(10,1)
otoliths	character(1,1)
age	character(2,1)
lv_para	integer
ld_para	integer

species_master	
code	character(3,1)
com_name	character(40,1)
sci_name	character(80,3)
oth_names	character(160,3)
notes	text(120,120)
usage	character(1,1)
descrptn	character(2,1)
family_com	character(40,1)
family_sci	character(40,1)
key	character(5,1)
pref_meas_meth	character(3,1)
max_length	integer
mtab_code	integer

t_fish_meas_codes	
fish_meas_code	character(1,1)
descrptn	character(65,1)

t_sex_codes	
sex_code	character(1,1)
descrptn	character(60,1)

t_stom_cond_codes	
stom_cond_code	character(1,1)
descrptn	character(60,1)

t_stom_state_codes	
stom_state_code	character(1,1)
descrptn	character(60,1)

t_gon_sys_desc	
stage_meth	character(2,1)
spp_list	character(80,1)
descrptn	character(180,1)

t_gon_stg_meth	
stage_meth	character(2,1)
stage_type	character(1,1)
gonad_stage	character(1,1)
descrptn	character(180,1)

Figure 5: ERD showing the relationships between *t_lgth*, *t_lgth_stage*, and *t_fish_bio* and the master code tables in the rdb database.

3.3 Stratum definition tables

A project was set up in June 2006 to upgrade the trawl database to capture data defining strata for trawl surveys in the form of polygons. The following ERD shows two new tables that were added: *t_trip_stratum* and *t_stratum_defn*. The attribute *stratum_def* in table *t_stratum_defn* contains actual strata polygons which are defined as a long text string in the format of WKT (Well Known Text) specified by OGC (Open Geospatial Consortium)'s SFS (Simple Feature Specification). Table *t_trip_stratum* links a trip's stratum to its definition in *t_stratum_defn*.

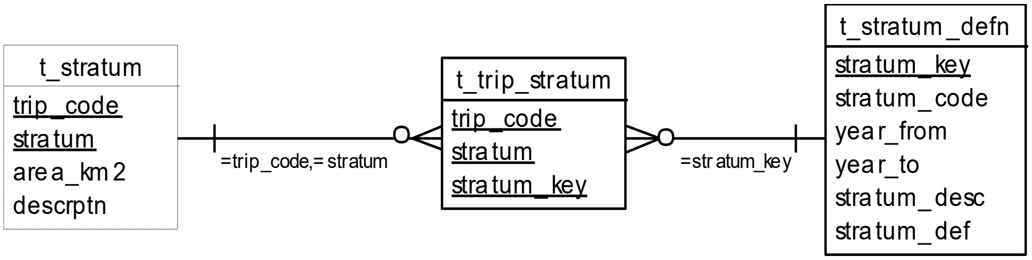


Figure 6: ERD showing the relationship between t_stratum and t_stratum_defn tables.

3.4 Site table

The table *t_site* was developed in the latter half of 2012. This table was implemented to store the locations of sites identified for blue cod (BCO) potting surveys, where a site is a geographical location near to which sampling may take place during a survey.

The design for these blue cod potting surveys has typically included a number of fixed sites identified before the first survey, with a subset of these sites selected for each survey. More recent surveys have incorporated random sites either in conjunction with fixed sites or all sites for the survey have been randomly generated. For sites that are selected for a survey, stations (i.e., pots) are typically placed around the site, or along a section of coastline near the site. This table is designed to capture data on the sites from these surveys, both fixed and random, including any fixed sites not yet surveyed (ie that pots have not been set around).

The sites are identified by the *stn_code* (station code) attribute. Each site is within a stratum, and the labelling convention for the fixed sites is for the station code to be composed of the stratum number followed by a letter, eg 1A, 1B, are *stn_code* values for sites in stratum 1. This labelling convention allows the t_site and t_station tables to be joined for a survey or survey series on *stn_code*, to check that stations are within a reasonable distance from the site etc. Station code values must be unique for a trip. The attribute *site_type* in the *t_site* table documents if the site is a fixed or random site.

The attribute *initial_trip* follows the format of the trip_code attribute in the t_trip, t_station and t_stratum tables etc. For any new survey series this attribute should contain the trip code for the first trip in the series. For existing survey series, this attribute may instead contain the value of trip code against which the sites were first loaded, eg lhr1001 for the Marlborough sounds survey series.

This t_site table could also be used to record the location selected for trawls in standard trawl surveys but has not been populated with these data to date.

3.5 Soviet trawl survey database description

The data schema for the Soviet trawl survey data was inherited from the original MS Access database that the data arrived in from the Ministry of Fisheries. While the original table names are still in use the **trawl** database, the attributes have all been renamed to provide a degree of consistency between the Soviet and all other trawl survey data. The Soviet data are in four main tables, with eight other lookup tables providing details of the various codes used. Tables containing Soviet data all have uppercase names.

The top-level table is *TSH* (Table 13), which conceptually represents the station form. Basically, the details recorded are similar to *t_station* (Table 4), however, there are some important differences:

- Gear methods are identified as a prefix to the name of gear deployed, as recorded in the *trawl_type* attribute, rather than as gear method code;
- Wind direction and speed as combined into one attribute, *wind*;
- Gear performance is determined solely by the amount of damage sustained to the gear and hence loss of catch, as recorded in the damage attribute, rather than a more holistic approach to gear performance using such other indicators as door spread and headline height.

The species composition of the catch is recorded in the table *TSP* (Table 14). This table provides a processing record of the sampling done to the catch as it stores multiple entries for species for each station as they were sampled and processed. *TSP* does not provide total species catch weights and or numbers for each station, although in the majority of stations these can be calculated by summing the attributes *num_fish* and *weight* by trip key and station number.

Length frequency data are held in the *TMS* table (Table 15). This is compatible with *t_lgth* (Table 8), with each record containing a fish length, sex a, and frequency (*c.f. t_lgth* where each record contains a male frequency, a female frequency, and a total frequency for each fish length).

Individual fish biological data are recorded in the *TFI* table (Table 16). This is very similar to *t_fish_bio* (Table 10), recording fish length, sex, weight, sexual maturity and stomach contents. However, there are two fish length methods and two fish weights that can be recorded: *lgth_fork* recorded fork length; *lgth_standard* records standard length; *wgt_total* records total whole fish weight; and *wgt_gutted* records the gutted fish weight.. The sexual maturity and stomach contents codes are based on a different coding system that usual. The Soviets also recorded the state of fatty tissue in fish, as recorded in the *fatness_code* attribute.

Of the eight lookup tables, six simply provide descriptions of the various codes employed in the four main tables. These are: *FISHCOD* (Table 17) for fish species identification numbers, some of which contain matching 3-character NIWA species codes; *SEXCOD* (Table 18) for sex codes; *STOMACHCOD* (Table 19) for stomach contents codes; *FATCOD* (Table 20) for fish fatness codes; *DAMAGECOD* (Table 21) for net damage and performance codes; and *WAVECOD* (Table 22) for wave and swell codes. The remaining two lookup tables are used to help decode attributes in the *TSH* table. They are:

TRAWLCOD (Table 23) for describing the gear method; and *WINDCOD* (Table 24) for describing the characteristics of the Beaufort Scale for wind force.

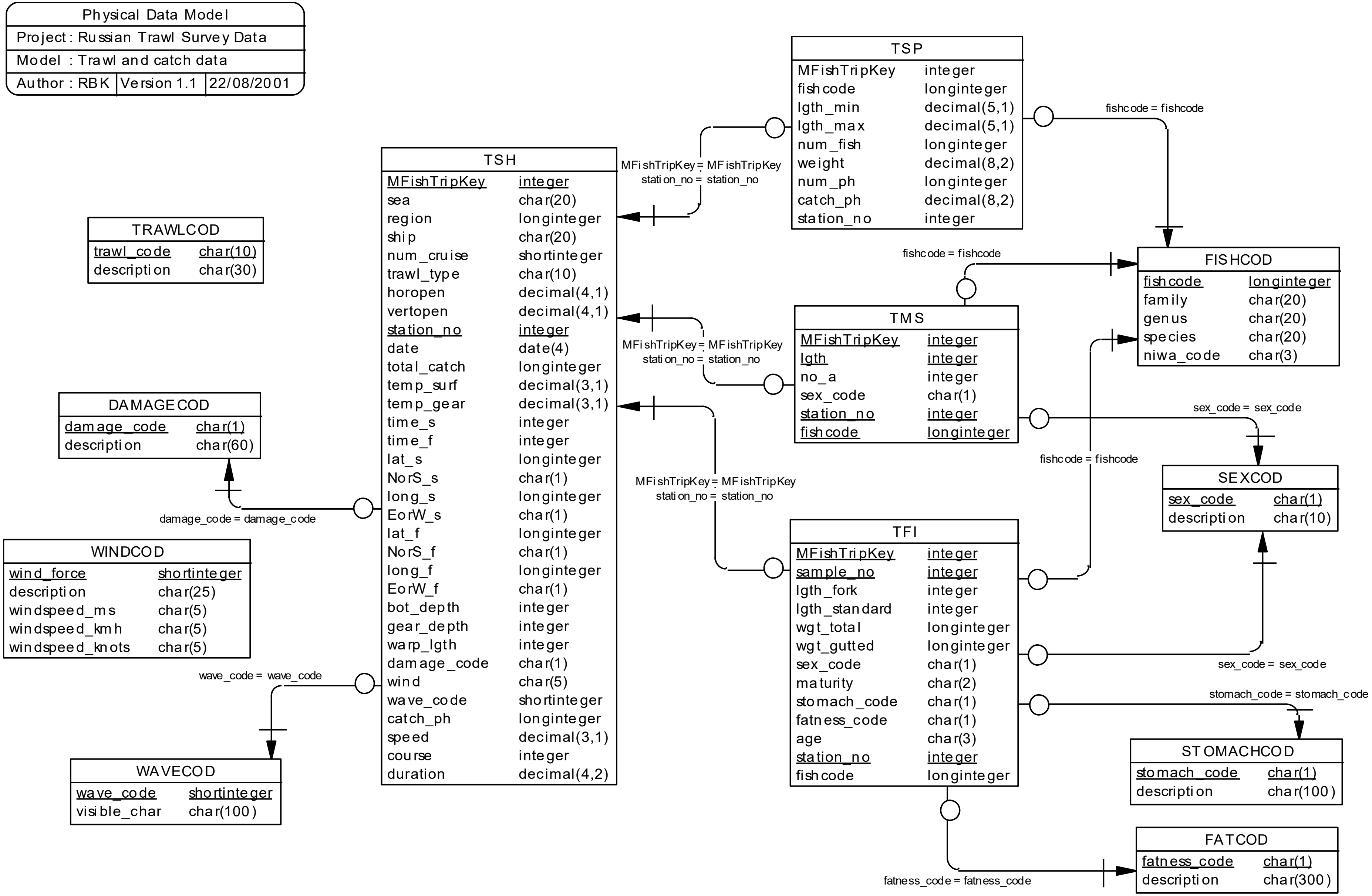


Figure 7: ERD of the Soviet trawl survey data.

3.6 Standards for fisheries databases

The **trawl** database was created in 1988. In 1993, a set of standards was set in place (Ng 1992) for all fisheries databases. The most significant effect of these standards was the requirement to add the prefix "t_" to the table names and "v_" to view names. However, this raised some potentially serious issues at the time. The **trawl** database represents a central part of fisheries stock assessment, and therefore has numerous scripts, programs, and applications linked to it. These range from the data checking and data loading routines, through to biomass calculations. Any changing of table names was considered to have a very significant flow-on effect to all relevant fisheries applications at that time.

As a compromise, views were created on all the tables, where the view name is the same as the original table for that view. This allowed all pre-1993 software to work with the database standard. The following table lists the original table name with the appropriate new table name and view.

Pre-1993	**Post 1993 Now**	
Original Table Name	**New Table Name**	**View Name**
trip	t_trip	trip
trip_comm	t_trip_comm	trip_comm
stratum	t_stratum	stratum
station	t_station	station
stat_comm	t_stat_comm	stat_comm
catch	t_catch	catch
lgth	t_lgth	lgth
fish_bio	t_fish_bio	fish_bio

Note that the standards for fisheries databases also require that the views *HOK_bio*, *ORH_bio* and *SNA_bio* on the table *t_fish_bio* should all be prefixed by "v_". Renaming these database views in order to conform to these standards would have the same adverse flow-on effect as renaming the tables. Rather than creating another set of views that were named to standards; i.e., creating the views *v_HOK_bio*, *v_ORH_bio*, and *v_SNA_bio*, it was decided to leave them unchanged, and hence they do not conform to the standard naming conventions.

Since the introduction of these standards, the tables *t_subcatch* and *t_lgth_stage*, and the view *v_scampi* were created in the database in accordance with the standards. Hence, there are no special views on these, which have the prefixes removed.

These views such as 'trip' on table t_trip were not retained in the change to Postgres about 2015, and may have been dropped earlier.

None of the Soviet trawl survey tables comply with this 1993 naming standard. Rather, these table names were inherited directly from the original Russian data extracts.

4 Table summaries

The **trawl** database has ten tables containing trawl survey data and four views showing species-specific data. An additional four tables contain Soviet trawl survey data of the New Zealand region, with eight associated lookup tables.

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

1. **t_trip :** contains profile information on all trips.
2. **t_trip_comm :** contains comments for a particular trip.
3. **t_stratum :** contains details of strata surveyed for a trip.
4. **t_station :** contains data on location, gear used and environment at each station within a trip.
5. **t_stat_comm :** contains comments for a station in a trip.
6. **t_catch :** contains information (weight, number caught etc) on all species caught at each station on a trip.
7. **t_subcatch :** contains information for each subcatch of each species caught at each station on a trip.
8. **t_lgth :** contains length frequency data on sampled species in a trip by station.
9. **t_lgth_stage :** contains length frequency data by gonad stage.
   a) **v_scampi :** contains length frequency data by gonad stage and egg development stage for female scampi.
10. **t_fish_bio :** contains biological data (gonad staging, stomach contents etc) on any species sampled in a trip by station. From this table, three views for the major species emanate. They are:
    a) **HOK_bio :** contains data from *t_fish_bio* for hoki only.
    b) **ORH_bio :** contains data from *t_fish_bio* for orange roughy only.
    c) **SNA_bio :** contains data from *t_fish_bio* for snapper only.

11. **t_stratum_defn :** contains definition information for strata.
12. **t_trip_stratum :** links trip's strata with their definition data in table *t_stratum_defn*.
13. **t_site :** Data on location, for sites selected for stations. (e.g., for blue cod potting surveys).
14. **t_trip_site :** Table to join t_site to t_station.
15. **t_gear :** Details of gear used on a trip (yet to be formally implemented).

The following are the tables for the Soviet trawl survey data contained in **trawl**:

16. **TSH** : trawl shot details table, including location, time, speed, depth, and total catch.
17. **TSP** : records the species composition of the catches. Often includes sampling and weighing of totals for each sample or entire catches species for samples or entire catches.
18. **TMS** : contains length frequency data by species for different trawls. This is occasional sampling rather than complete sampling.
19. **TFI** : contains details of biological analyses of individual fish from the trawl; e.g., length, weight, sex, maturity, stomach contents, fatness.
20. **FISHCOD** : contains fish species identification codes.
21. **SEXCOD** : contains fish sex codes.
22. **STOMACHCOD** : contains fish stomach contents and fullness codes.
23. **FATCOD** : contains fish fatness codes.

24. **DAMAGECOD** : contains codes of gear performance and damage to gear that may affect trawl catchability. Synonymous with *gear_perf* in the *t_stations* table.
25. **WAVECOD** : contains details of codes used to denote sea surface, swell and wave characteristics.
26. **TRAWLCOD** : contains details of general types of trawl gear used.
27. **WINDCOD** : contains descriptive data for the beaufort wind force scale.

5 trawl tables

The following are listings of the tables in the **trawl** database, including attribute names, data types (and any check constraints), and comments.

5.1 Table 1: t_trip

Table t_trip

Comment: Profile information on all trips held in this database.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code - 3 char vessel name, 2 digit year and 2 digit trip number.
proj_code	character varying(10)		Project or programme code for this trip as in the management database.
date_s	date		Start date for the trip.
date_f	date		Finish date for the trip.
leader	character varying(40)		Name of trip leader.
master	character varying(40)		Name of trip master(s).
areas	character varying(24)		Codes of area(s) surveyed separated by commas.
mainspp	character varying(31)		Target species code(s) separated by commas.
gear1	character varying(52)		Gear_meth code, codend, liner & cover mesh sizes (mm), ground rope length and height, sweep & bridle lengths (m), plus default values for headline height, headline ht code, distance between wings, wing distance code, distance between doors and distance door code separated by commas for 1st gear code used.
gear2	character varying(52)		Gear_meth code, codend, liner & cover mesh sizes (mm), ground rope length and height, sweep & bridle lengths (m) etc, separated by commas, as for gear1 for 2nd gear code used.
gear3	character varying(52)		Gear_meth code, codend, liner & cover mesh sizes (mm), ground rope length and

		height, sweep & bridle lengths (m) etc, separated by commas, as for gear1 for 3rd gear code used.
gear4	character varying(52)	Gear_meth code, codend, liner & cover mesh sizes (mm), ground rope length and height, sweep & bridle lengths (m) etc, separated by commas, as for gear1 for 4th gear code used.
gear5	character varying(52)	Gear_meth code, codend, liner & cover mesh sizes (mm), ground rope length and height, sweep & bridle lengths (m) etc, separated by commas, as for gear1 for 5th gear code used.
gear6	character varying(52)	Gear_meth code, codend, liner & cover mesh sizes (mm), ground rope length and height, sweep & bridle lengths (m) etc, separated by commas, as for gear1 for 6th gear code used.
staff	text	Name(s) of all staff on the trip.

```
Indexes:
   "pk_t_trip" PRIMARY KEY, btree (trip_code)
Check constraints:
   "t_trip_proj_code_check" CHECK (proj_code::text ~ '^[A-Z]+[0-9]+[A-Z]*'::text)
```

5.2 Table 2: t_trip_comm

Comment: Comments for a particular trip.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as defined in the trip table.
comments	text	No	Any comments about this trip e.g. details about gear used apart from those recorded in the trip table.

```
Indexes:
   "nx_t_trip_comm_trip_code" btree (trip_code)

Foreign-key constraints:
   "fk_t_trip_comm_t_trip_1" FOREIGN KEY (trip_code) REFERENCES
      trawl.t_trip(trip_code)
```

5.3 Table 3: t_stratum

Comment: Table of strata surveyed in all trips.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
stratum	character varying(4)	No	Stratum code - unique within a trip.
area_km2	numeric(8,2)		Size of a stratum in sq. km. (km2) - must be greater than 0 sq. km.
lgth_coast	numeric(8,2)		length of coastline in km, as used for BCO potting surveys
min_depth	integer		Minimum depth of sea bottom for stratum, when stratum is defined by depth.
max_depth	integer		Maximum depth of sea bottom for stratum, when stratum is defined by depth.
descrptn	text		Short description of the stratum e.g. location and previously depths.

```
Indexes:
   "pk_t_stratum" PRIMARY KEY, btree (trip_code, stratum)
   "nx_t_stratum_area_km2" btree (area_km2)
Check constraints:
    "t_stratum_area_km2_check" CHECK (area_km2 > 0::numeric
     OR lgth_coast > 0::numeric)
    "t_stratum_depth_check" CHECK (min_depth <= max_depth)
Foreign-key constraints:
   "fk_t_stratum_t_trip_1" FOREIGN KEY (trip_code) REFERENCES
     trawl.t_trip(trip_code)
```

5.4 Table 4: t_station

Comment: Data on location, gear used and environment at each station on a trip.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as defined in the trip table.
station_no	integer	No	Station number - unique within a trip.
categories	character varying(2)		2 separate user-defined categories; definitions should be in trip comments.
area	character varying(4)		Code describing area, refer to rdb.area_codes or http://marlin.niwa.co.nz.
stn_code	character varying(4)		Code for a permanent station occupied repeatedly.
stratum	character varying(4)		Stratum number if trip is a stratified survey, else a transect code.
biomass_flag	character varying(1)		Code to indicate if the station is valid for biomass estimation, 0 = No, 1 = Yes, 2 or more = yes for non core strata and hence 1 is for core strata for this trip. See t_trip_comm for meaning of values >= 2.
course	integer		Course of vessel during the shot (course-made-good).
date_s	date		Starting date of the shot (dd Mmm yy format).
time_s	integer		Starting time (24hr,NZST) of the shot (hhmm format).
fix_s	character varying(2)		Method of fixing position at start of tow, refer rdb.t_fix_meth_codes or Appendix 1 of the database documentation.
timefix_s	integer		Time (in minutes) elapsed since last position fix at the start of tow.
lat_s	integer		Latitude of vessel at start of tow (ddmmmm format, d=deg., m=min. to 2 implied dec. pl.).
nors_s	character varying(1)		Tow start position hemisphere.
long_s	integer		Longitude of vessel at start of tow

		(dddmmmm format, d=deg., m=min. to 2 implied dec. pl.).
eorw_s	character varying(1)	Tow start position meridian.
gear_s	integer	Depth of lowest part of gear (groundrope) at start of the tow (m).
bot_gs	integer	Depth of sea bottom at gear position at start of the tow (m).
bot_vs	integer	Depth of sea bottom at vessel position at start of the tow (m).
date_f	date	Finishing date of the shot.
time_f	integer	Finishing time (24hr,NZST) of shot (hhmm format).
fix_f	character varying(2)	Method of fixing position at end of tow, refer rdb.t_fix_meth_codes or Appendix 1 of the database documentation.
timefix_f	integer	Time (in minutes) elapsed since last position fix at end of the tow.
lat_f	integer	Latitude of vessel at end of tow (ddmmmm format, d=deg., m=min. to 2 implied dec. pl.).
nors_f	character varying(1)	Tow finish position hemisphere.
long_f	integer	Longitude of vessel at end of tow (dddmmmm format, d=deg., m=min. to 2 implied dec. pl.).
eorw_f	character varying(1)	Tow finish position meridian.
gear_f	integer	Depth of lowest part of gear (groundrope) at end of the tow (m).
bot_gf	integer	Depth of sea bottom at gear position at end of tow (m).
bot_vf	integer	Depth of sea bottom at vessel position at end of tow (m).
min_gdepth	integer	Minimum depth of lowest part of gear (groundrope) during the tow (m).
max_gdepth	integer	Maximum depth of lowest part of gear (groundrope) during the tow (m).

gear_meth	character varying(2)	Gear method code, refer rdb.meth_codes.
gear_code	smallint	Code for set of gear used, details in trip record.
gear_units	smallint	Number of units of gear used in the tow.
gear_perf	smallint	Code for performance of gear during the tow, refer to Appendix 1 of the database documentation.
path	smallint	Code describing configuration of path of shot, refer to Appendix 1 of the database documentation.
speed	numeric(3,1)	Average speed through water during shot (knots).
distance	numeric(4,2)	Distance of gear over bottom (nautical miles).
head_ht	numeric(3,1)	Average headline height (m).
head_code	character varying(1)	Code showing how headline height was determined, refer to rdb.t_headline_codes or Appendix 1 of the database documentation.
dist_wings	numeric(4,1)	Average distance between wings (m).
distwing_code	character varying(1)	Code to indicate how distance between the wings was determined for this tow, refer rdb.t_wing_dist_codes or Appendix 1 of the database documentation.
dist_doors	numeric(4,1)	Average distance between doors of gear (m).
distdoor_code	character varying(1)	Code to indicate how the distance between the doors was determined for this tow, refer rdb.t_door_dist_codes or Appendix 1 of the database documentation.
warp_lgth	integer	Length of warp during the tow (m).
fish_locn	character varying(1)	Code to indicate the location of the fish at the net mouth during the shot as observed on the net sonde, refer rdb.t_fish_obs_codes or Appendix 1 of the database documentation.

wind_dir	integer	Wind direction (degrees true), 999=No wind.
wind_force	smallint	Wind force on Beaufort scale.
air_temp	numeric(3,1)	Air temperature (degrees C).
air_press	numeric(5,1)	Air pressure (millibars).
cloud_cov	smallint	Code describing cloud cover during tow in eighths.
sea_cond	smallint	Code describing condition of sea, refer to Appendix 1 of the database documentation.
sea_col	smallint	Code describing colour of sea, refer to Appendix 1 of the database documentation.
swell_ht	smallint	Code describing height of swell, refer to Appendix 1 of the database documentation.
swell_dir	integer	Direction of the swell (degrees true).
bot_type	smallint	Code describing sea bottom type, refer to Appendix 1 of the database documentation.
bot_cont	smallint	Code describing sea bottom contour, refer to Appendix 1 of the database documentation.
surf_temp	numeric(3,1)	Surface temperature (degrees C).
bot_temp	numeric(3,1)	Temperature at bottom (degrees C).
wind_spd	smallint	Wind speed from anemometer (m/s) (1 knot=0.51 m/s).
secchi	numeric(4,1)	Depth at which Secchi disc becomes invisible (m).
other	character varying(6)	Any other details, should be fully commented.
dlat_s	numeric(7,5)	Latitude of vessel at start of tow in decimal degree.
dlon_s	numeric(8,5)	Longitude of vessel at start of tow in decimal degree.
dlat_e	numeric(7,5)	Latitude of vessel at end of tow in

		decimal degree.
dlon_e	numeric(8,5)	Longitude of vessel at end of tow in decimal degree.
startp	geometry	Position of vessel at start of tow as gis point type.
endp	geometry	Position of vessel at end of tow as gis point type.
track	geometry	Track line of vessel from start position to end position of tow as gis line type.

```
Indexes:
   "pk_t_station" PRIMARY KEY, btree (trip_code, station_no)
   "nx_t_station_endp" gist (endp)
   "nx_t_station_gear_meth" btree (gear_meth)
   "nx_t_station_max_gdepth" btree (max_gdepth)
   "nx_t_station_min_gdepth" btree (min_gdepth)
   "nx_t_station_startp" gist (startp)
   "nx_t_station_track" gist (track)
Check constraints:
   "enforce_dims_endp" CHECK (ndims(endp) = 2)
   "enforce_dims_startp" CHECK (ndims(startp) = 2)
   "enforce_dims_track" CHECK (ndims(track) = 2)
   "enforce_geotype_endp" CHECK (geometrytype(endp) = 'POINT'::text OR endp IS
                                           NULL)
   "enforce_geotype_startp" CHECK (geometrytype(startp) = 'POINT'::text OR startp
                                           IS NULL)
   "enforce_geotype_track" CHECK (geometrytype(track) = 'LINESTRING'::text OR
                                           track IS NULL)
   "enforce_srid_endp" CHECK (srid(endp) = 4326)
   "enforce_srid_startp" CHECK (srid(startp) = 4326)
   "enforce_srid_track" CHECK (srid(track) = 4326)
   "t_station_bot_cont_check" CHECK (bot_cont >= 0 AND bot_cont <= 5)
   "t_station_bot_type_check" CHECK (bot_type IS NULL OR bot_type >= 0 AND
                                           bot_type <= 12)
   "t_station_cloud_cov_check" CHECK (cloud_cov >= 0 AND cloud_cov <= 8)
   "t_station_course_check" CHECK (course IS NULL OR course >= 0 AND course <=
                                           359)
   "t_station_eorw_f_check" CHECK (eorw_f::text ~ '[EW]'::text)
   "t_station_eorw_s_check" CHECK (eorw_s::text ~ '[EW]'::text)
   "t_station_gear_perf_check" CHECK (gear_perf IS NULL OR gear_perf >= 1 AND
                                           gear_perf <= 4)
   "t_station_lat_f_check" CHECK (lat_f::text ~ '[1-8][0-9][0-5][0-9][0-9][0-
                                           9]'::text)
   "t_station_lat_s_check" CHECK (lat_s IS NULL OR lat_s::text ~ '[1-8][0-9][0-
                                           5][0-9][0-9][0-9]'::text)
   "t_station_long_f_check" CHECK (long_f IS NULL OR long_f >= 0 AND long_f <=
                                           1800000)
   "t_station_long_s_check" CHECK (long_s IS NULL OR long_s >= 0 AND long_s <=
                                           1800000)
   "t_station_nors_f_check" CHECK (nors_f::text ~ '[NS]'::text)
   "t_station_nors_s_check" CHECK (nors_s::text ~ '[NS]'::text)
   "t_station_path_check" CHECK (path IS NULL OR path >= 1 AND path <= 8)
   "t_station_sea_col_check" CHECK (sea_col >= 1 AND sea_col <= 8)
   "t_station_sea_cond_check" CHECK (sea_cond IS NULL OR sea_cond >= 0 AND
                                           sea_cond <= 9)
   "t_station_swell_dir_check" CHECK (swell_dir IS NULL OR swell_dir >= 0 AND
                                           swell_dir <= 359 OR swell_dir = 999)
   "t_station_swell_ht_check" CHECK (swell_ht >= 1 AND swell_ht <= 3)
   "t_station_time_f_check" CHECK (time_f >= 0 AND time_f <= 2359)
   "t_station_time_s_check" CHECK (time_s >= 0 AND time_s <= 2359)
   "t_station_wind_dir_check" CHECK (wind_dir IS NULL OR wind_dir >= 0 AND
                                           wind_dir <= 359 OR wind_dir = 999)
   "t_station_wind_force_check" CHECK (wind_force >= 0 AND wind_force <= 12)

Foreign-key constraints:
   "fk_t_station_area_codes_9" FOREIGN KEY (area) REFERENCES rdb.area_codes(code)
   "fk_t_station_meth_codes_4" FOREIGN KEY (gear_meth)
    REFERENCES rdb.meth_codes(code)
   "fk_t_station_t_door_dist_codes" FOREIGN KEY (distdoor_code)
    REFERENCES rdb.t_door_dist_codes(door_code)
   "fk_t_station_t_fish_obs_codes_8" FOREIGN KEY (fish_locn)
    REFERENCES rdb.t_fish_obs_codes(fish_obs_code)
   "fk_t_station_t_fix_meth_codes_2" FOREIGN KEY (fix_s)
    REFERENCES rdb.t_fix_meth_codes(fix_meth_code)
   "fk_t_station_t_fix_meth_codes_3" FOREIGN KEY (fix_f)
    REFERENCES rdb.t_fix_meth_codes(fix_meth_code)
   "fk_t_station_t_headline_codes_5" FOREIGN KEY (head_code)
    REFERENCES rdb.t_headline_codes(headline_code)
   "fk_t_station_t_trip_1" FOREIGN KEY (trip_code)
    REFERENCES trawl.t_trip(trip_code)
   "fk_t_station_t_wing_dist_codes" FOREIGN KEY (distwing_code)
    REFERENCES rdb.t_wing_dist_codes(wing_dist_code)
```

5.5 Table 5: t_stat_comm

Comment: Comments for a station in a trip.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
comments	text	No	Comments for this station - should include comments about catch & LF data or any special action taken during tow.

```
Indexes:
   "nx_t_stat_comm_station_no" btree (station_no)
   "nx_t_stat_comm_trip_code" btree (trip_code)

Foreign-key constraints:
   "fk_t_stat_comm_t_station" FOREIGN KEY (trip_code, station_no)
    REFERENCES trawl.t_station(trip_code, station_no)
```

5.6 Table 6: t_catch

Comment: Information (weight, number caught etc) on all species caught at each station on a trip.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
species	character(3)	No	Species code, refer to rdb.curr_spp or https://marlin.niwa.co.nz/species_codes
weight	numeric(9,3)		Weight of the species caught at that station (kg).
wt_meth	character varying(1)		Code of method used to determine weight of catch, refer rdb.t_wgt_meth_codes or Appendix 1 of the database documentation.
number	integer		Counted or estimated number of this species.
oth_data	character varying(3)		Col. 1=L/F?, Col. 2=Biologicals, Col. 3=Otoliths. In each column, 1=Yes and 0 or blank=No.
comments	text		Any comments about this catch entry.

```
Indexes:
   "nx_t_catch_species" btree (species)
   "nx_t_catch_station_no" btree (station_no)
   "nx_t_catch_trip_code" btree (trip_code)

Foreign-key constraints:
   "fk_t_catch_t_station" FOREIGN KEY (trip_code, station_no)
    REFERENCES trawl.t_station(trip_code, station_no)
   "fk_t_catch_t_wgt_meth_codes" FOREIGN KEY (wt_meth)
    REFERENCES rdb.t_wgt_meth_codes(wgt_meth_code)
```

5.7 Table 7: t_subcatch

Comment: Information (weight, sample weight etc) on each subcatch for each species sampled. Generally, the subcatch is identical to the whole catch for any species sampled.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
species	character(3)	No	Species code, refer to rdb.curr_spp.
subcatch_no	smallint	No	Sequential number to identify each subcatch of a species taken from the whole catch for that species.
weight	numeric(9,3)		Weight (kg) of the subcatch of the species caught at that station.
wt_meth	character varying(1)		Code of method used to determine weight of subcatch, refer rdb.t_wgt_meth_codes or Appendix 1 of the database documentation.
samp_wt	numeric(9,3)		Weight (kg) of the sample of fish used for measuring.
sample_meth	character varying(1)		Code of method used in sampling LFs (if done), refer rdb.t_samp_sel_codes or Appendix 1 of the database documentation.
measure_meth	character varying(1)		Code of method used to measure fish lengths (if LFs done), refer rdb.t_fish_meas_codes or Appendix 1 of the database documentation.
stage_meth	character varying(2)		Code for gonad staging method used, refer rdb:t_gon_sys_desc.
comments	text		Any comments about this subcatch entry.

```
Indexes:
   "nx_t_subcatch_species" btree (species)
   "nx_t_subcatch_station_no" btree (station_no)
   "nx_t_subcatch_trip_code" btree (trip_code)
Foreign-key constraints:
   "fk_t_subcatch_2" FOREIGN KEY (wt_meth)
    REFERENCES rdb.t_wgt_meth_codes(wgt_meth_code)
   "fk_t_subcatch_3" FOREIGN KEY (sample_meth)
    REFERENCES rdb.t_samp_sel_codes(samp_sel_code)
   "fk_t_subcatch_4" FOREIGN KEY (measure_meth)
    REFERENCES rdb.t_fish_meas_codes(fish_meas_code)
```

```
"fk_t_subcatch_t_gon_sys_desc_5" FOREIGN KEY (stage_meth)
 REFERENCES rdb.t_gon_sys_desc(stage_meth)
```

5.8 Table 8: t_lgth

Comment: Length frequency data on sampled species in a trip.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
species	character(3)	No	Species code, refer to rdb.curr_spp or https://marlin.niwa.co.nz/species_codes
subcatch_no	smallint	No	Subcatch number as in subcatch table.
lgth	integer	No	Measured length of the fish (cm), except for scampi (SCI) which is measured in mm.
percent_samp	numeric(5,2)		Sampling percentage associated with this subcatch.
no_a	integer		Number of all measured fish at this length in this subcatch.
no_m	integer		Number of all measured male fish at this length in this subcatch.
no_f	integer		Number of all measured female fish at this length in this subcatch.

```
Indexes:
   "nx_t_lgth_species" btree (species)
   "nx_t_lgth_station_no" btree (station_no)
   "nx_t_lgth_trip_code" btree (trip_code)
Check constraints:
   "t_lgth_percent_samp_check" CHECK
    (percent_samp IS NULL OR percent_samp >= 0.00 AND percent_samp <= 100.00)
```

5.9 Table 9: t_lgth_stage

Comment: Table to store staged length frequency data.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
species	character(3)	No	Species code, refer rdb.curr_spp.
subcatch_no	smallint	No	Subcatch number as in subcatch table.
lgth	integer	No	Measured length (cm) of the fish.
no_m1	integer		Number of all Stage 1 males sampled at this length.
no_m2	integer		Number of all Stage 2 males sampled at this length.
no_m3	integer		Number of all Stage 3 males sampled at this length.
no_m4	integer		Number of all Stage 4 males sampled at this length.
no_m5	integer		Number of all Stage 5 males sampled at this length.
no_m6	integer		Number of all Stage 6 males sampled at this length.
no_m7	integer		Number of all Stage 7 males sampled at this length.
no_m8	integer		Number of all Stage 8 males sampled at this length.
no_m9	integer		Number of all Stage 9 males sampled at this length.
no_f1	integer		Number of all Stage 1 females sampled at this length.
no_f2	integer		Number of all Stage 2 females sampled at this length.
no_f3	integer		Number of all Stage 3 females sampled at this length.
no_f4	integer		Number of all Stage 4 females sampled at this length.

no_f5	integer	Number of all Stage 5 females sampled at this length.
no_f6	integer	Number of all Stage 6 females sampled at this length.
no_f7	integer	Number of all Stage 7 females sampled at this length.
no_f8	integer	Number of all Stage 8 females sampled at this length.
no_f9	integer	Number of all Stage 9 females sampled at this length.

```
Indexes:
   "nx_t_lgth_stage_species" btree (species)
   "nx_t_lgth_stage_station_no" btree (station_no)
   "nx_t_lgth_stage_trip_code" btree (trip_code)
```

5.9.1 View v_scampi

```
Comment: View containing length frequency data by gonad stage and egg development
         stage for female scampi.
```

Column	**Type**
trip_code	character varying(7)
station_no	integer
species	character(3)
subcatch_no	smallint
lgth	integer
egg0	integer
egg1	integer
egg2	integer
egg3	integer
egg4	integer
gonad1	integer
gonad2	integer
gonad3	integer
gonad4	integer
gonad5	integer
gonad8	integer

View definition:

```
SELECT t_lgth_stage.trip_code, t_lgth_stage.station_no,
t_lgth_stage.species, t_lgth_stage.subcatch_no, t_lgth_stage.lgth,
t_lgth_stage.no_m8 AS egg0, t_lgth_stage.no_m1 AS egg1,
t_lgth_stage.no_m2 AS egg2, t_lgth_stage.no_m3 AS egg3,
t_lgth_stage.no_m4 AS egg4, t_lgth_stage.no_f1 AS gonad1,
t_lgth_stage.no_f2 AS gonad2, t_lgth_stage.no_f3 AS gonad3,
t_lgth_stage.no_f4 AS gonad4, t_lgth_stage.no_f5 AS gonad5,
t_lgth_stage.no_f8 AS gonad8
 FROM trawl.t_lgth_stage
 WHERE t_lgth_stage.species::text = 'SCI'::text;
```

5.10 Table 10: t_fish_bio

Comment: Biological data (gonad staging, stomach contents etc) on all fish species.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
species	character(3)	No	Species code, refer to rdb.curr_spp or https://marlin.niwa.co.nz/species_codes
subcatch_no	smallint	No	Subcatch number as in subcatch table.
fish_no	integer	No	Unique fish number within a station.
measure_meth	character varying(1)	No	Code of method used to measure fish lengths, refer rdb.t_fish_meas_codes.
lgth	numeric(4,1)		Measured length (decimal cm) of the fish.
weight	real		Measured weight (grams) of the fish.
sex	character varying(1)		Sex code. 1=male, 2=female, 3=immature or unable to determine, refer rdb.t_sex_codes.
gonad_wt	numeric(5,1)		Weight of fish gonad. May be left blank intentionally.
gonad_stage	character varying(1)		1 digit code for the stage of gonad development, refer rdb.t_gon_stg_meth.
stomach_state	character varying(1)		Code used to describe the state of the stomach fullness, 0-Empty, 1=Trace, 2=Part full(1/4-3/4), 3=Full, 8=Regurgitated/Eviserated, 9=Everted, refer rdb.t_stom_state_codes.
stomach_cond	character varying(1)		Code used to describe the digestion condition of the stomach contents, 1=Fresh, 2=1/2 digested, 3=Digested, 4=Mixed digestion states. Refer rdb.t_stom_cond_codes.
stomach_wt	integer		Weight of fish stomach in grams.
prey1	character varying(3)		Code for 1st species found in stomach, may also be MINITAB code, refer rdb.curr_spp.

vol1	smallint	Percentage volume of 1st species to total stomach content.
prey2	character varying(3)	Code for 2nd species found in stomach, may also be MINITAB code, refer rdb.curr_spp.
vol2	smallint	Percentage volume of 2nd species to total stomach content.
prey3	character varying(3)	Code for 3rd species found in stomach, may also be MINITAB code, refer rdb.curr_spp.
vol3	smallint	Percentage volume of 3rd species to total stomach content.
prey4	character varying(3)	Code for 4th species found in stomach, may also be MINITAB code, refer rdb.curr_spp.
vol4	smallint	Percentage volume of 4th species to total stomach content.
prey5	character varying(3)	Code for 5th species found in stomach, may also be MINITAB code, refer rdb.curr_spp.
vol5	smallint	Percentage volume of 5th species to total stomach content.
comments	text	
otoliths	character varying(1)	Were otoliths taken from this fish (Y = Yes; N = No).
age	character varying(2)	Age read from otoliths - 2-digit age or b=broken otolith, u=unreadable otolith. Now recorded in the age database.
lv_para	integer	Parasite count on left ventral muscle tissue.
ld_para	integer	Parasite count on left dorsal muscle tissue.

```
Indexes:
   "nx_t_fish_bio_species" btree (species)
   "nx_t_fish_bio_trip_code" btree (trip_code)
Check constraints:
   "t_fish_bio_age_check" CHECK (age::text ~ '[0-9bu]{1,}'::text)
   "t_fish_bio_gonad_stage_check" CHECK (gonad_stage::text >= 1::text AND
                                        gonad_stage::text <= 9::text)
   "t_fish_bio_otoliths_check" CHECK (otoliths::text ~ '[YN]'::text)
```

```
Foreign-key constraints:
   "fk_t_fish_bio_6" FOREIGN KEY (stomach_cond)
    REFERENCES rdb.t_stom_cond_codes(stom_cond_code)
   "fk_t_fish_bio_t_fish_means_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)
```

5.10.1 View hok_bio

```
Comment: View of all hoki (HOK) biological data.

Column          Type

trip_code       character varying(7)
station_no      integer
species         character(3)
subcatch_no     smallint
fish_no         integer
measure_meth    character varying(1)
lgth            numeric(4,1)
weight          real
sex             character varying(1)
gonad_wt        numeric(5,1)
gonad_stage     character varying(1)
stomach_state   character varying(1)
stomach_cond    character varying(1)
stomach_wt      integer
prey1           character varying(3)
vol1            smallint
prey2           character varying(3)
vol2            smallint
prey3           character varying(3)
vol3            smallint
prey4           character varying(3)
vol4            smallint
prey5           character varying(3)
vol5            smallint
comments        character varying(10)
otoliths        character varying(1)
age             character varying(2)
lv_para         integer
ld_para         integer
View definition:
                SELECT t_fish_bio.trip_code, t_fish_bio.station_no,
                t_fish_bio.species, t_fish_bio.subcatch_no, t_fish_bio.fish_no,
                t_fish_bio.measure_meth, t_fish_bio.lgth, t_fish_bio.weight,
                t_fish_bio.sex, t_fish_bio.gonad_wt, t_fish_bio.gonad_stage,
                t_fish_bio.stomach_state, t_fish_bio.stomach_cond,
                t_fish_bio.stomach_wt, t_fish_bio.prey1, t_fish_bio.vol1,
                t_fish_bio.prey2, t_fish_bio.vol2, t_fish_bio.prey3,
                t_fish_bio.vol3, t_fish_bio.prey4, t_fish_bio.vol4,
                t_fish_bio.prey5, t_fish_bio.vol5, t_fish_bio.comments,
                t_fish_bio.otoliths, t_fish_bio.age, t_fish_bio.lv_para,
                t_fish_bio.ld_para
                 FROM trawl.t_fish_bio
                 WHERE t_fish_bio.species ~~ 'HOK'::text;
```

5.10.2 View orh_bio

Comment: View of all orange roughy (ORH) biological data.

Column	Type	Null?	Description
trip_code	character varying(7)		
station_no	integer		
species	character(3)		
subcatch_no	smallint		
fish_no	integer		
measure_meth	character varying(1)		
lgth	numeric(4,1)		
weight	real		
sex	character varying(1)		
gonad_wt	numeric(5,1)		
gonad_stage	character varying(1)		
stomach_state	character varying(1)		
stomach_cond	character varying(1)		
stomach_wt	integer		
prey1	character varying(3)		
vol1	smallint		
prey2	character varying(3)		
vol2	smallint		
prey3	character varying(3)		
vol3	smallint		
prey4	character varying(3)		
vol4	smallint		
prey5	character varying(3)		
vol5	smallint		
comments	character varying(10)		
otoliths	character varying(1)		
age	character varying(2)		
lv_para	integer		
ld_para	integer		

View definition:

```
SELECT t_fish_bio.trip_code, t_fish_bio.station_no,
t_fish_bio.species, t_fish_bio.subcatch_no, t_fish_bio.fish_no,
t_fish_bio.measure_meth, t_fish_bio.lgth, t_fish_bio.weight,
t_fish_bio.sex, t_fish_bio.gonad_wt, t_fish_bio.gonad_stage,
t_fish_bio.stomach_state, t_fish_bio.stomach_cond,
t_fish_bio.stomach_wt, t_fish_bio.prey1, t_fish_bio.vol1,
t_fish_bio.prey2, t_fish_bio.vol2, t_fish_bio.prey3,
t_fish_bio.vol3, t_fish_bio.prey4, t_fish_bio.vol4,
t_fish_bio.prey5, t_fish_bio.vol5, t_fish_bio.comments,
t_fish_bio.otoliths, t_fish_bio.age, t_fish_bio.lv_para,
t_fish_bio.ld_para
 FROM trawl.t_fish_bio
 WHERE t_fish_bio.species ~~ 'ORH'::text;
```

5.10.3 View sna_bio

Comment: View of all snapper (SNA) biological data.

Column	Type
trip_code	character varying(7)
station_no	integer
species	character(3)
subcatch_no	smallint
fish_no	integer
measure_meth	character varying(1)
lgth	numeric(4,1)
weight	real
sex	character varying(1)
gonad_wt	numeric(5,1)
gonad_stage	character varying(1)
stomach_state	character varying(1)
stomach_cond	character varying(1)
stomach_wt	integer
prey1	character varying(3)
vol1	smallint
prey2	character varying(3)
vol2	smallint
prey3	character varying(3)
vol3	smallint
prey4	character varying(3)
vol4	smallint
prey5	character varying(3)
vol5	smallint
comments	character varying(10)
otoliths	character varying(1)
age	character varying(2)
lv_para	integer
ld_para	integer

View definition:

```
SELECT t_fish_bio.trip_code, t_fish_bio.station_no,
t_fish_bio.species, t_fish_bio.subcatch_no, t_fish_bio.fish_no,
t_fish_bio.measure_meth, t_fish_bio.lgth, t_fish_bio.weight,
t_fish_bio.sex, t_fish_bio.gonad_wt, t_fish_bio.gonad_stage,
t_fish_bio.stomach_state, t_fish_bio.stomach_cond,
t_fish_bio.stomach_wt, t_fish_bio.prey1, t_fish_bio.vol1,
t_fish_bio.prey2, t_fish_bio.vol2, t_fish_bio.prey3,
t_fish_bio.vol3, t_fish_bio.prey4, t_fish_bio.vol4,
t_fish_bio.prey5, t_fish_bio.vol5, t_fish_bio.comments,
t_fish_bio.otoliths, t_fish_bio.age, t_fish_bio.lv_para,
t_fish_bio.ld_para
 FROM trawl.t_fish_bio
 WHERE t_fish_bio.species ~~ 'SNA'::text;
```

5.11 Table 11: t_stratum_defn

Comment: Strata definition table containing polygon definition strings in WKT format for strata.

Column	Type	Null?	Description
stratum_key	integer	No	Unique integer as primary key.
stratum_code	character varying(20)	No	User defined 4 character long code for a strata corresponding to those in t_stratum table.
year_from	integer		Year a strata defined.
year_to	integer		Year a strata is valid to.
stratum_desc	character varying(256)		Any text description about a stratum.
stratum_def	character varying(400000)		Long text strings in WKT format defining strata polygons.
poly	geometry		Stratum polygon as gis multipolygon type.

```
Indexes:
   "pk_t_stratum_defn" PRIMARY KEY, btree (stratum_key)
   "nx_t_stratum_defn_poly" gist (poly)
Check constraints:
   "enforce_dims_poly" CHECK (ndims(poly) = 2)
   "enforce_geotype_poly" CHECK
             (geometrytype(poly) = 'POLYGON'::text
             OR geometrytype(poly) = 'MULTIPOLYGON'::text OR poly IS NULL)
   "enforce_srid_poly" CHECK (srid(poly) = 4326)
```

5.12 Table 12: t_trip_stratum

Comment: A table linking trip strata to their polygon definition.

Column	Type	Null?	Description
trip_code	character varying(7)	No	7 char long trip code identifying a trip.
stratum	character varying(4)	No	4 char long code together with trip_code uniquely identifying a stratum.
stratum_key	integer		Foreign key to t_stratum_defn uniquely identifying a stratum.

```
Indexes:
   "pk_t_trip_stratum" PRIMARY KEY, btree (trip_code, stratum)

Foreign-key constraints:
   "fk_t_trip_s_fk_t_trip_t_stratu" FOREIGN KEY (trip_code, stratum)
    REFERENCES trawl.t_stratum(trip_code, stratum)
   "fk_t_trip_stratum_1" FOREIGN KEY (stratum_key)
    REFERENCES trawl.t_stratum_defn(stratum_key)
```

5.13 Table 13: t_site

Comment: Data on location, for sites selected for stations.

Column	Type	Null?	Description
site_key	integer	No	Unique identifier for t_site record.
initial_trip	character varying(7)	No	Trip code as defined in trip table for first trip to define this site.
stn_code	character varying(4)		Code for a permanent station occupied repeatedly.
area	character varying(4)		Code describing area, refer to rdb.area_codes or https://marlin.niwa.co.nz/area_codes/
stratum	character varying(4)		Stratum code or stratum number.
lat	integer		Latitude of site position (ddmmmm format, d=deg., m=min. to 2 implied decimal place).
nors	character varying(1)		Site position hemisphere, N or S.
long	integer		Longitude of site position (dddmmmm format, d=deg., m=min. to 2 implied decimal place).
eorw	character varying(1)		Site position meridian, E or W.
site_type	character varying(8)		Code to record type of site, including if a fixed or random site.
dlat	numeric(7,5)		Latitude of site position in decimal degrees.
dlon	numeric(8,5)		Longitude of site position in decimal degrees.
position	geometry		Position of the site as gis point type.

```
Indexes:
   "pk_t_site" PRIMARY KEY, btree (site_key)
   "ui_t_site" UNIQUE, btree (initial_trip, stn_code)
Check constraints:
   "enforce_dims_geom" CHECK (ndims("position") = 2)
   "enforce_geotype_geom" CHECK (geometrytype("position") = 'POINT'::text OR
                                             "position" IS NULL)
   "enforce_srid_geom" CHECK (srid("position") = 4326)
```

5.14 Table 14: t_trip_site

Comment: Table to join t_site to t_station.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as defined in the trip table.
station_no	integer	No	Station number as in station table.
site_key	integer	No	Site key as in t_site table.

```
Indexes:
   "pk_t_trip_site" PRIMARY KEY, btree (trip_code, station_no)
   "nx_t_trip_site_trip_code" UNIQUE, btree (trip_code, station_no)

Foreign-key constraints:
   "fk_t_trip_site_t_site_1" FOREIGN KEY (site_key)
    REFERENCES trawl.t_site(site_key)
   "fk_t_trip_site_t_station_3" FOREIGN KEY (trip_code, station_no)
    REFERENCES trawl.t_station(trip_code, station_no)
   "fk_t_trip_site_t_trip_2" FOREIGN KEY (trip_code)
    REFERENCES trawl.t_trip(trip_code)
```

5.15 Table 15: t_gear

Comment: Gear information used in a trip.

Column	Type	Null?	Description
gear_key	integer	No	Unique identifier for t_gear record.
trip_code	character varying(7)	No	Trip code as in trip table.
gear_code	numeric(1,0)		Code to identify the gear used, usually from 1 to 6.
gear_method	character varying(2)		Gear method code. Refer rdb.meth_codes.
codend_mesh	integer		Size of cod end mesh for the gear (mm).
liner_mesh	integer		Size of liner mesh for the gear (mm).
cover_mesh	integer		Size of cover mesh for the gear (mm).
gr_lgth	integer		Length of ground rope for the gear (m).
gr_ht	numeric(4,1)		Height of ground rope from the bottom for the gear (m).
sweep_lgth	integer		Length of sweeps for the gear (m).
bridle_lgth	integer		Length of bridle for the gear (m).
default_head_ht	numeric(4,1)		Default headline height for the gear (m).
head_code	character varying(4)		Code to describe how the default headline height was derived. Refer rdb.t_headline_codes.
default_wing_dist	numeric(4,1)		Default distance between wings for the gear (m).
wing_code	character varying(4)		Code to descibe how the default distance between the wings was derived. Refer rdb.t_wing_dist_codes.
default_door_dist	numeric(4,1)		Default distance between doors for the gear (m).
door_code	character varying(2)		Code to describe how the default door distance code was derived. Refer rdb.t_door_dist_codes.
gear_text	character varying(52)		Original data from gear1 - gear6 attribute, as the original record may not parse into the separate attributes

```
                                          correctly.

Indexes:
   "pk_t_gear" PRIMARY KEY, btree (gear_key)
   "ui_t_gear" UNIQUE, btree (trip_code, gear_code)

Foreign-key constraints:
   "fk_t_gear_t_trip" FOREIGN KEY (trip_code)
    REFERENCES trawl.t_trip(trip_code)
```

5.16 Table 16: tsh

Comment: Soviet trawl shot details table, including location, time, speed, depth, and total catch.

Column	Type	Null?	Description
mfishtripkey	integer	No	Unique code for each trip.
sea	character varying(20)		Sea or ocean name.
region	integer		Area code.
ship	character varying(20)		Vessel name.
num_cruise	smallint		Cruise number of this ship (this is not unique across ships).
trawl_type	character varying(10)		Gear type (see table TRAWLCOD for details).
horopen	numeric(4,1)		Trawl horizontal opening in metres.
vertopen	numeric(4,1)		Trawl vertical opening in metres.
station_no	integer	No	Haul number.
date	date		Date.
total_catch	integer		Total catch in kilograms.
temp_surf	numeric(3,1)		Water surface temperature in degrees celcius.
temp_gear	numeric(3,1)		Haul depth temperature in degrees celcius.
time_s	integer		Start time (24-hour) Using the local time zone.
time_f	integer		Haul back (24-hour) time Using the local time zone.
lat_s	integer		Start Latitude to 0.1 minute accuracy.
nors_s	character varying(1)		Tow start position hemisphere.
long_s	integer		Start Longitude to 0.1 minute accuracy.
eorw_s	character varying(1)		Tow start position meridian.
lat_f	integer		Haul back latitude to 0.1 minute accuracy.

nors_f	character varying(1)	Tow finish position hemisphere.
long_f	integer	Haul back longitude to 0.1 minute accuracy.
eorw_f	character varying(1)	Tow finish position meridian.
bot_depth	integer	Bottom depth in metres.
gear_depth	integer	Gear depth in metres.
warp_lgth	integer	The length of wire out where the trawl is fixed in metres.
damage_code	character varying(1)	Gear performance code. Refer to the DAMAGECOD table.
wind	character varying(5)	Wind direction and speed (beaufort scale). Refer to the WINDCOD table.
wave_code	smallint	Swell (value of 1..10). Refer to the WAVECOD table.
catch_ph	integer	Total catch per hour in kg/hour.
speed	numeric(3,1)	Vessel speed in knots.
course	integer	Vessel course in degrees.
duration	numeric(4,2)	Tow duration in hours.

```
Indexes:
   "pk_tsh" PRIMARY KEY, btree (mfishtripkey, station_no)
```

5.17 Table 17: tsp

Comment: Records the species composition of the Soviet catches. Often includes sampling and weighing of totals for each species for samples or entire catches.

Column	Type	Null?	Description
mfishtripkey	integer		Unique code for each trip (note: TSP records are unique on MFishTripKey, station_no, fishcode, lgth_min, lgth_max, weight).
fishcode	integer		Species code. Refer to the FISHCOD table.
lgth_min	numeric(5,1)		Minimum length of fish (fork length in cm.).
lgth_max	numeric(5,1)		Maximum length of fish (fork length in cm.).
num_fish	integer		Catch of species in number.
weight	numeric(8,2)		Catch of species in weight (kg).
num_ph	integer		Species catch numbers per hour (CPUE).
catch_ph	numeric(8,2)		Species catch weight per hour (CPUE).
station_no	integer		Haul number.

```
Indexes:
   "nx_tsp_mfishtripkey" btree (mfishtripkey)
   "nx_tsp_station_no" btree (station_no)
```

5.18 Table 18: tms

Comment: Contains length frequency data by species for different Soviet trawls. This is occasional sampling rather than complete sampling.

Column	Type	Null?	Description
mfishtripkey	integer	No	Unique code for each trip.
lgth	integer	No	Length frequency length (mm).
no_a	integer		Frequency.
sex_code	character varying(1)		Sex code. Refer to the SEXCOD table.
station_no	integer	No	Haul number.
fishcode	integer	No	Species code. Refer to the FISHCOD table.

5.19 Table 19: tfi

Comment: Contains details of biological analyses of individual fish from the Soviet trawl; e.g., length, weight, sex, maturity, stomach contents, fatness.

Column	Type	Null?	Description
mfishtripkey	integer	No	Unique code for each trip.
sample_no	integer	No	Sample number.
lgth_fork	integer		Fork length (mm).
lgth_standard	integer		Standard length (mm).
wgt_total	integer		Total weight of the whole fish (g).
wgt_gutted	integer		Weight of the gutted fish (g).
sex_code	character varying(1)		Sex code. Refer the SEXCOD table.
maturity	character varying(2)		Maturity code. Main stages have a leading 0. Transitional stages as adjacent stages code combination.
stomach_code	character varying(1)		Code for the stomach content scale. Refer to the STOMACHCOD table.
fatness_code	character varying(1)		Fish fatness code. Refer to the FATCOD table.
age	character varying(3)		Fish age - a count of otolith annuli rings. Presence or absence of growth increments is indicated by a - or a +.
station_no	integer	No	Haul number.
fishcode	integer		Species code. Refer to the FISHCOD table.

5.20 Table 20: fishcod

Comment: Soviet fish species identification codes. NIWA 3-character species codes are populated where known for linking to the curr_spp table in the rdb database.

Column	Type	Null?	Description
fishcode	integer	No	Unique identification number for each species.
family	character varying(20)		Scientific family name.
genus	character varying(20)		Genus.
species	character varying(20)		Species.
niwa_code	character varying(3)		3-character NIWA species code. Refer to rdb.curr_spp or https://marlin.niwa.co.nz/species_codes

Indexes:
   "pk_fishcod" PRIMARY KEY, btree (fishcode)

5.21 Table 21: sexcod

Comment: Soviet codes and descriptions for fish sexes.

Column	Type	Null?	Description
sex_code	character varying(1)	No	Unique 1-character code for the sex of a fish.
description	character varying(10)		Description of the sex code.

Indexes:
   "pk_sexcod" PRIMARY KEY, btree (sex_code)

5.22 Table 22: stomachcod

Comment: Soviet codes and descriptions for fish stomach contents and fullness.

Column	Type	Null?	Description
stomach_code	character varying(1)	No	Code for the stomach content scale.
description	character varying(100)		Description of the stomach content scale.

```
Indexes:
   "pk_stomachcod" PRIMARY KEY, btree (stomach_code)
```

5.23 Table 23: fatcod

Comment: Soviet codes and descriptions for fatness of fish.

Column	Type	Null?	Description
fatness_code	character varying(1)	No	Code for the fatness of fish.
description	character varying(300)		Description of the fatness of the fish.

```
Indexes:
   "pk_fatcod" PRIMARY KEY, btree (fatness_code)
```

5.24 Table 24: damagecod

Comment: Soviet codes of gear performance and damage to gear that may affect trawl catchability. Synonymous with gear_perf in the t_stations table.

Column	Type	Null?	Description
damage_code	character varying(1)	No	Code for the type of damage to the gear performance.
description	character varying(60)		Description of the damage to the performance of the trawl (c.f. gear_perf).

```
Indexes:
   "pk_damagecod" PRIMARY KEY, btree (damage_code)
```

5.25 Table 25: wavecod

Comment: Soviet codes used to denote sea surface and wave characteristics.

Column	Type	Null?	Description
wave_code	smallint	No	Unique number for each sea surface scale unit.
visible_char	character varying(100)		Wave and sea surface characteristic description.

```
Indexes:
   "pk_wavecod" PRIMARY KEY, btree (wave_code)
```

5.26 Table 26: trawlcod

Comment: Contains details of general types of trawl gear used in Soviet trawl surveys.

Column	Type	Null?	Description
trawl_code	character varying(10)	No	Code for the type of trawl.
description	character varying(30)		Description for the type of trawl.

```
Indexes:
   "pk_trawlcod" PRIMARY KEY, btree (trawl_code)
```

5.27 Table 27: windcod

Comment: Contains descriptive details for the beaufort wind force scale.

Column	Type	Null?	Description
wind_force	smallint	No	Wind force (Beaufort Scale).
description	character varying(25)		Wind force description.
windspeed_ms	character varying(5)		Wind force average wind speed (m/s).
windspeed_kmh	character varying(5)		Wind force average wind speed (km/h).
windspeed_knots	character varying(5)		Wind force average wind speed (knots).

```
Indexes:
   "pk_windcod" PRIMARY KEY, btree (wind_force)
```

6 **trawl** business rules

6.1 Introduction to business rules

The following are a list of business rules pertaining to the **trawl** database. A business rule is a written statement specifying what the information system (i.e., any system that is designed to handle trawl 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. 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.

Being a closed dataset, the Soviet trawl survey data have no business rules recorded.

6.2 Summary of rules

Trawl survey trip details (t_trip)

trip_code Trip code, must be unique. Trip codes are in the following format: 3 character vessel code (which should be in the *t_vessels* table in the **rdb** database); 2 digit year (e.g., 99 = 1999, 00 = 2000); 2 digit sequential trip number for each vessel each year.

proj_code Project code must be a valid code within the NIWA project management system.

date_s The start date of the trip must be a legitimate date, and should have a value.

date_f The start date of the trip must be a legitimate date, and should have a value.

Multiple column checks on date:
The start date must not be later than the finish date.

areas Each of the listed area codes must be a valid code as listed in the *area_codes* table in the **rdb** database.

mainspp Each of the listed species codes must be a valid code as listed in the *curr_spp* table in the **rdb** database.

gear1 – **gear6** Gear descriptions. The following describe the format, and where applicable, the business rules for the description of gear used during a trip:

gear number Must be a unique, sequential number from 1 to 6 to identify each unit of gear.

gear method Must be a valid code as listed in the *meth_codes* table in the **rdb** database.

codend mesh

liner mesh

cover mesh

ground rope length

ground rope height

sweep length

bridle length

default headline height

headline height code Must be a valid code as listed in the *t_headline_codes* table in the **rdb** database

default wing distance

wing distance code Must be a valid code as listed in the *t_wing_dist_codes* table in the **rdb** database

default door distance

door distance code Must be a valid code as listed in the *t_door_dist_codes* table in the **rdb** database

Trawl survey trip comments (t_trip_comm)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

Trawl survey stratum details (t_stratum)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

Trawl survey stratum definition details (t_stratum_defn)

stratum_code Must have a value and should be equal to a *stratum* code in the *t_stratum* table.

stratum_def Must have a value, and should be in the WKT format for defining polygons.

Trawl survey station details (t_station)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

area Area code must be a valid code as listed in the *area_codes* table in the **rdb** database.

course Course must be within the range of 0 – 359 degrees.

date_s The date at the start of a station must be a legitimate date, and should have a value.

Multiple column checks on start date:
The date must fall within the range of the trip start and finish dates.

time_s Start time of the station must be a valid 24-hour time and fall within the range of 0 – 2359 hours.

fix_s }
fix_f } The method of position fix code must be valid code as listed in the *t_fix_meth_codes* table in the **rdb** database.

lat_s Must be a valid latitude

NorS_s Northern or Southern Hemisphere at station start, must be equal to either “N” or “S”.

long_s Must be a valid longitude.

EorW_s Longitude east or west at station start, must be equal to either “E” or “W”.

bot_gs Depth of sea bottom must not be less than depth of gear

date_f The date at the finish of a station must be a legitimate date.

Multiple column checks on finish date:
The date must fall within the range of the trip start and finish dates.

time_f Finish time of the station must be a valid 24-hour time and fall within the range of 0 – 2359.

Multiple columns checks on date and time:
The start date must not be later than the finish date and within a reasonable time period.

lat_f Must be a valid latitude

NorS_f Northern or Southern Hemisphere at station finish, must be equal to either “N” or “S”.

long_f Must be a valid longitude.

EorW_f Longitude east or west at station finish, must be equal to either "E" or "W".

Multiple columns checks on position:
The finish position should be within a reasonable distance from the start position for the gear type used.
Where a latitude column has a value the corresponding NorS column should have a valid value. Where a longitude column has a value the corresponding EorW column should have a valid value.

Multiple column and record checks on positions, dates and times:
The speed calculated as the distance between consecutive station positions divided by the time difference, should be reasonable for the vessel, eg 12 knots or less for trawlers.

bot_gf Depth of sea bottom must not be less than depth of gear.

min_gdepth Minimum gear depth must be less than or equal to the depth of gear at the start and finish of the station.

max_gdepth Maximum gear depth must be greater than or equal to the minimum gear depth and the depth of gear at the start and finish of the station

gear_meth Gear method code must be a valid code as listed in the *meth_codes* table in the **rdb** database.

gear_code Must be within the range 1 – 6 to relate to gear details in *gear1* to *gear6* respectively in the *t_trip* table.

Multiple columns check on: gear_meth and gear_code
The combination of *gear_meth* and *gear_code* should exist in one of the gear1 to gear6 attributes of the *t_trip* table.

gear_perf The gear performance code must be valid code as listed in Appendix 1.

path The path code must be valid code as listed in Appendix 1.

speed The vessel's recorded speed during the station should be within the range 0 – 5 knots and be reasonable for the gear method.

distance The distance traveled during the station should be reasonable for the gear method.

**Multiple columns check on: distance; start and finish positions;
and speed and start/finish times:**

The distance traveled during a station as calculated by (1) the difference between start and finish positions; (2) speed * elapsed time; and (3) recorded distance should be in approximate agreement.

head_code Headline height code must be a valid code as listed in the *t_headline_codes* table in the **rdb** database.

distwing_code Distance between trawl wings code must be a valid code as listed in the *t_wing_dist_codes* table in the **rdb** database.

distdoor_code Distance between trawl doors code must be a valid code as listed in the *t_door_dist_codes* table in the **rdb** database.

fish_locn Must be a valid code as listed in the *t_fish_obs_codes* table in the **rdb** database.

wind_dir Wind direction must fall within the range of 0-359, 999.

wind_force Wind force must fall within the range of 0 – 12.

air_temp Air temperature should fall within the reasonable range of 5 – 30.

air_press Air pressure should fall within the reasonable range of 960 to 1040.

cloud_cov Cloud cover must fall within the range of 0-8.

sea_cond The sea condition code must be valid code as listed in Appendix 1.

sea_col The sea colour code must be valid code as listed in Appendix 1.

swell_ht The swell height code must be valid code as listed in Appendix 1.

swell_dir Wind direction must fall within the range of 0-359, 999.

bot_type The bottom type code must be valid code as listed in Appendix 1.

bot_cont The bottom contour code must be valid code as listed in Appendix 1.

surf_temp Sea surface temperature should fall within the reasonable range of 5 – 28.

bot_temp Sea bottom temperature should fall within the reasonable range of 3 – 25.

wind_spd Wind speed should fall within the reasonable range of 0 - 30.

secchi Secchi disc distance should fall within the reasonable range of 0 – 40.

Trawl survey station comments (t_stat_comm)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

Multiple columns check on trip code and station number:
The combination of trip code and station number must exist in the *t_station* table.

Trawl survey catch details (t_catch)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

Multiple columns check on trip code and station number:
The combination of trip code and station number must exist in the *t_station* table.

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

weight Must be a valid number greater then 0

wt_meth Must be a valid code as listed in the *t_wgt_meth_codes* table in the **rdb** database.

oth_data Must be up to 3 characters long, with each character being a "1" (meaning presence), "0" (meaning absence), or " " (meaning not recorded).

Trawl survey subcatch details (t_subcatch)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

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

Multiple columns check on trip code, station number, and species:
The combination of trip code, station number, and species must exist in the *t_catch* table.

subcatch_no Must be a unique number within a single trip code, station number, and species.

weight Must be a valid number greater then 0

wt_meth Must be a valid code as listed in the *t_wgt_meth_codes* table in the **rdb** database.

sample_meth Must be a valid sample selection method code as listed in the *t_samp_sel_codes* table in the **rdb** database.

measure_meth Must be a valid fish measurement method code as listed in the *t_fish_meas_codes* table in the **rdb** database.

Multiple columns check on species and measure_meth:
The fish measurement method code must be valid for the species sampled.

stage_meth Must be a valid gonad stage method code as listed in the *t_gon_sys_desc* table in the **rdb** database.

Multiple columns check on species and stage_meth:
The gonad stage method code must be valid for the species sampled.

Trawl survey length frequency details (t_lgth)

Multiple columns check on trip code, station number, species, and subcatch number:
The combination of trip code, station number, species, and subcatch number must exist in the *t_subcatch* table.

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

lgth 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.

percent_samp Must be a valid percentage up to 100%

Multiple columns check on percentage sampled and *t_subcatch:sample_meth*:
The sample selection method code must valid with the percentage sampled

no_m} Must be a valid integer greater than 0
no_f}
no_a} **Multiple columns check on *no_a*, *no_m*, and *no_f*:**
The number in *no_a* must be equal to or greater than the sum of *no_m* and *no_f*.

Trawl survey gonad staged length frequency details (t_lgth_stage)

Multiple columns check on trip code, station number, species, and subcatch number:
The combination of trip code, station number, species, and subcatch number must exist in the *t_subcatch* table.

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

lgth
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.

no_m1 – no_m8}
no_f1 – no_f8}
Must be a valid integer greater than or equal to 0.

Trawl survey fish biology details (t_fish_bio)

Multiple columns check on trip code, station number, species, and subcatch number:
The combination of trip code, station number, species, and subcatch number must exist in the *t_subcatch* table.

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

Multiple columns check on trip code, station number, species, subcatch number and fish number:
The combination of trip code, station number, species, subcatch number and fish number should be unique.

lgth 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 **Multiple columns check on species and weight:**
The fish weight should be less than a reasonable maximum fish weight for the species. Some reasonable maximum fish weights for some major species are given in Appendix 1.

sex Must be a valid sex code as listed in the *t_sex_codes* table in the **rdb** database.

gonad_wt Should not be more than $\frac{1}{3}$ of the total fish weight.

gonad_stage **Multiple column check on species, gonad stage, sex, and *t_subcatch*:stage_meth:**
Must be a valid gonad stage for the species, sex, and gonad staging method code as listed in the *t_gon_stg_meth* table in the **rdb** database.

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.

Blue cod potting survey site location details (t_site)

site_key Must be unique (system generated)

initial_trip Should be equal to a trip code as listed in the *t_trip* table.

stn_code The station code must be unique for the trip, ie trip series.

area The area code should be a valid code as listed in the *area_codes* table of the **rdb** database.

stratum Should be equal to a stratum code in the *t_stratum* table for the initial trip.

lat Must be a valid latitude.

nors Must be equal to either ‘N’ or ‘S’.

long Must be a valid longitude.

eorw Must be equal to either ‘E’ or ‘W’.

Multiple columns checks on lat, nors, long, eorw and stratum:
The resulting position should be within the stratum boundary for the stratum.

7 Acknowledgements

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

8 References

1. Mackay, K. 1998: Marine Research database documentation. 6. trawl. *NIWA Greta Point Internal Report No. 16.* 40p.

2. Francis, R. I. C. C. 1981: Stratified random trawl surveys of deepwater demersal fish stocks around New Zealand. *Fisheries Research Division Occasional Publication No. 32*. 28p.

3. Francis, R. I. C. C. 1984: An adaptive strategy for stratified random trawl surveys. *N.Z. Journal of Marine and Freshwater Research 18*: 59-71.

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

Appendix 1 – Reference Code Tables

Gear performance code

1.	Excellent
2.	Satisfactory, catch unlikely to be reduced by performance
3.	Unsatisfactory, catch probably reduced by malfunction or damage
4.	Unsatisfactory, catch reduced by malfunction or damage

Path code

1.	Horizontal straight line
2.	Vertical straight line
3.	Closed circle or loop
4.	Closed triangle or square
5.	Zigzag
6.	U-bend
7.	Contour at constant depth
8.	Retrack on straight line

Cloud cover code

0	Clear sky
1-8	Number of eighths of sky covered

Sea condition code

0	Calm, glassy	0m
1	Calm	0 – 0.1m
2	Smooth	0.1 – 0.5m
3	Slight	0.5 – 1m
4	Moderate	1 – 2.5m
5	Rough	2.5 – 4m
6	Very rough	4 – 6m
7	High	6 - 10m
8	Very high	10 – 15m
9	Huge	over 15m

Sea colour code

01	Deep blue
02	Blue
03	Light blue
04	Greeny blue
05	Bluey green
06	Deep green
07	Green
08	Yellow green

Swell height code

1	Low	0 – 2m
2	Moderate	2 – 4m
3	Heavy	over 4m

Bottom contour code

0	Unknown
1	Smooth/flat
2	Undulating
3	Hillocky
4	Rugged
5	Very rugged

Bottom type code

0	Unknown
1	Mud or ooze
2	Mud with some sand
3	Sand
4	Sand/gravel and shells
5	Shells (broken)
6	Gravel
7	Rock
8	Coral
9	Stone
10	Live shell beds
11	Mud with broken shells
12	Sponge beds

NB These codes listed above can also be found in the Trawl Instructions document.

Position fix method codes from the rdb database table t_fix_meth_codes
(used for t_station fix_s and fix_f attributes)

fix_meth_code

01	Radar
02	Dead reckoning
03	Astrofix
04	Transect marks
05	Radio (RDF)
06	Radar and RDF
07	SatNav
08	Global Positioning System satelites (GPS)
09	Local knowledge
10	GPX
11	Estimated position of sampling gear derived from GPS

Headline codes from the rdb database table t_headline_codes

head_code

0	From net plan
1	From tank test
2	Assumed from previous trips
3	Expected height from net sounder for current trip
4	From net sounder output
5	From scanmar output
6	Expected height from scanmar readings for current trip

Wing distance codes from the rdb database table t_wing_dist_codes

distwing_code

0	From the net plan
1	From tank tests
2	Assumed from previous trip
3	Calculated from current trip from warp angle, length etc.
4	Measured from current trip by wingtip transducers
5	Calc. from current trip by door mounted transducers
6	From scanmar output

Door distance codes from the rdb database table t_door_dist_codes

distdoor_code

0	From gear plan
1	From tank tests
2	Assumed from previous trips
3	Calc. from current trip using measured warp angle etc.
4	Calc. from current trip from wingtip transducers
5	Measured from current trip by door transducers
6	Average of measured doorspreads for current trip
7	Average of measured doorspreads for this depth, current trip
8	Estimated from warp_lgth with parameters from other voyages

Fish location codes from the rdb database table t_fish_obs_codes

fish_locn

0	Not noted for this tow
1	No fish where observed on the net sonde
2	All fish observed between headline and ground rope
3	Some fish observed above the headline
4	Some fish observed below the ground rope
5	Some fish observed above headline & below ground rope

Commonly used gear method codes from the rdb database table meth_codes,
refer to rdb.meth_codes for the complete list of codes, which contains over 60 codes.

gear_meth

01	Bottom trawl
03	High opening bottom trawl
05	Prawn trawl (scampi)
06	Midwater trawl
51	Pots (general)
72	Photography/Camera gear (still)
82	CTD (Conductivity, Temperature & Depth)

Weight method codes from the rdb database table t_wgt_meth_codes

wt_meth

1	Weighed
2	Scaled up from sub_sample weight
3	Calc. from LF (& and scaled up from sub-samples)
4	Number of cases * mean case weight
5	Estimated by eye
6	Calc. from processed weight and conversion factors
7	Combination of weighing methods
8	Number of fish * mean fish weight

Sample selection method codes from the rdb database table t_samp_sel_codes

sample_meth

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

Fish measurement method codes from rdb table t_fish_meas_codes (subset frequencly used in trawl)

fish_meas_code

1	Fork Length
2	Total Length
3	Standard Length
4	Mantle Length (squid)
5	Pelvic Length (rays)
B	Carapace Length - Orbit to Carapace notch (scampi)
G	Tip of snout to posterior end of dorsal fin (Ghost sharks)

Maximum fish weights (grams)

BOE	1,600
HAK	30,000
HOK	6,000
LIN	35,000
ORH	3,000
SNA	12,000
SSO	4,500