Database documentation: l_line

K. A. Mackay and L. Griggs

NIWA Fisheries Data Management
Database Documentation Series

Updated - 10 September 2001

Table of Contents

List of Figures

1 Database Document Series

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

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

This document is a brief introduction to the l_line sampling database **l_line**, and is a part of the database documentation series produced by NIWA.

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

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

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

2 Observer Longline Tuna Programme

From 1987 New Zealand placed fishery observers on selected foreign-licensed and some domestic-chartered Japanese vessels in the southern bluefin tuna fishery. Initial coverage was very low (fewer than 1% of sets made in 1987 and 1988) and confined to the East Cape area (see Appendix A) in June-July. Since 1989 the geographic and temporal coverage has more even, with the addition of observers south of New Zealand in areas 3 and 4.

At time of publication this programme is on going, with emphasis on all species of tuna as well as billfish, sharks, birds, and seal catches.

3 Data Structures

3.1 Table Relationships

This database encompasses several tables. The ERD for **l_line** (Figure 1) shows the physical data model structure[1] of the database and it's entities (each entity is implemented as a database *table*) and relationships between these tables. Each table represents an object, event, or concept in the real

[1] Also known as database *schema*

world that has been represented in the database. Each *attribute* of a table is a defining property or quality of the table.

All of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary key[2]. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed. Most of the tables in the **l_line** database also contain special attributes, called foreign keys[3].

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

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

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

The **l_line** database is implemented as a relational database. That is, each table is a special case of the mathematical construct known as a *relation* and hence elementary relation theory is used to deal with the data within tables and the relationships between them. All relationships in **l_line** are of the type *one-to-many*[4]. This is shown in the ERD by connecting a single line (indicating 'many') from the child table; e.g., *t_line_set*, to the parent table; e.g., *t_trip*, with an arrowhead (indicating 'one') pointing to the parent. For example, consider the relationship between the tables' *t_trip* (the parent table) and *t_line_set* (the child table). Any one trip in *t_trip* can have one or more longline sets in *t_line_set*, but any one longline set can only be a part of one trip. Note that the word 'many' applies to the possible number of records another is associated with. For a given instance, there might be zero, one, two, or more associated records, but if it is ever possible to have more than one, we use the word 'many' to describe the association.

Note that the one-to-many relationships can be either mandatory or optional. The optional relationship, denoted in the ERD by the symbol "O" at one or both ends of the relationship line, means that a record does not have to have any associated records. Conversely, the mandatory

[2] A primary key is an attribute or a combination of attributes that contains an unique value to identify that record.
[3] 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.
[4] A one-to-many relationship is where one record (the *parent*) in a table relates to one or many records (the *child*) in another table; e.g., one survey in *t_trip* can have many longline sets in *t_line_set* but any one longline set can only come from one trip.

relationship denoted in the ERD by a bar symbol across the relationship line, means that a record has to have at least one associated record. For example, if we consider again the one-to-many relationship between the tables ***t_trip*** and ***t_line_set***, which has a mandatory ‘one’ and an optional ‘many’. This means that one trip record can have zero or more (many) longline sets within it, but one longline set must have one, and only one, associated record in the trip table.

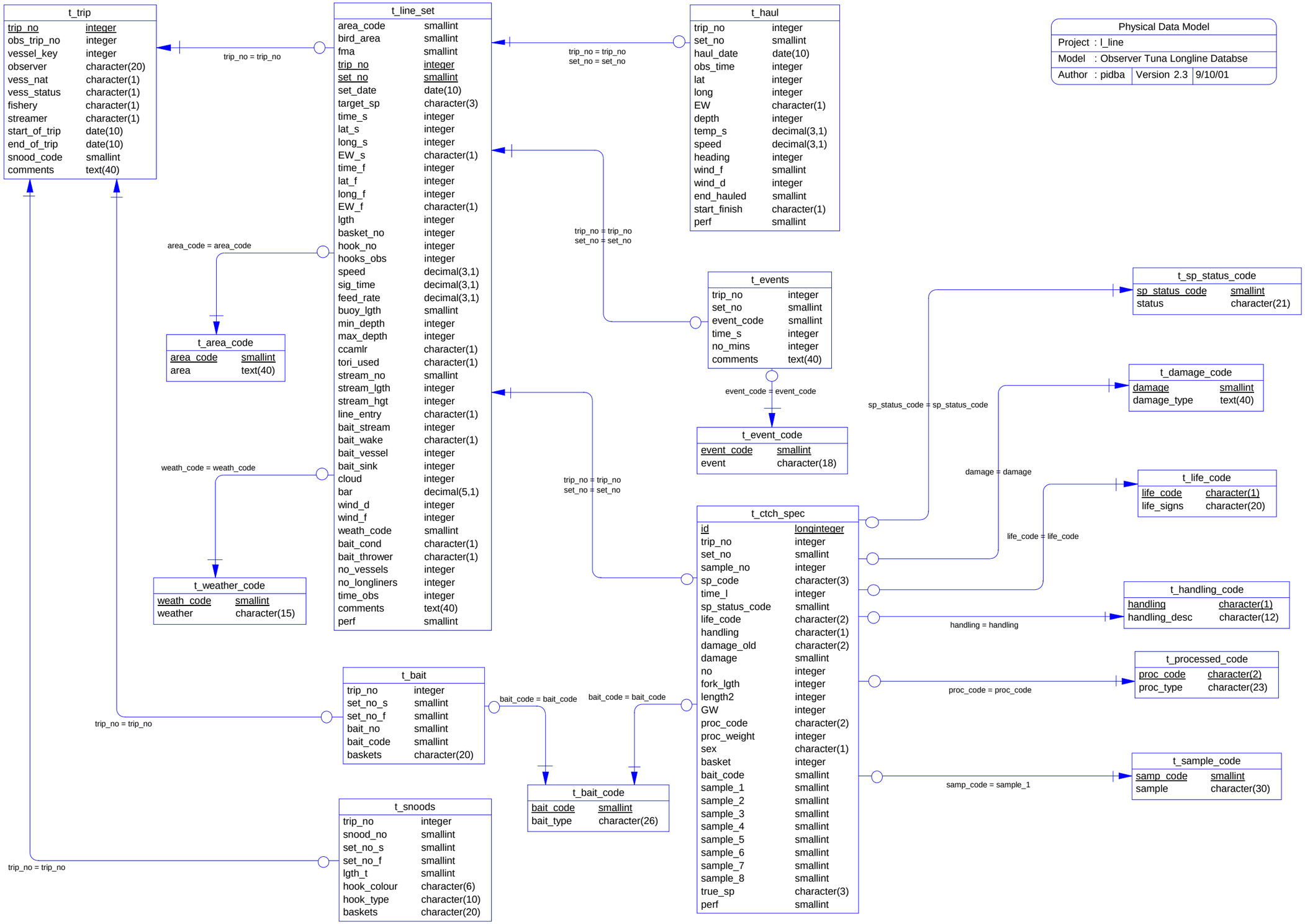

Figure 1: ERD for the l_line database.

These relationships are enforced in the database by the use of referential constraints[5]. 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:

i. a parent record is deleted;
ii. the parent record is altered so the relationship is lost;
iii. or a child record is entered without a parent record.

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

```
Referential:      error message (attribute[, attribute]) |INSERT|
                                                         |DELETE|
                  parent table (attribute[, attribute])
```

Items stacked between vertical lines | | are options of which one must be chosen.

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

```
Referential:      INVALID TRIP_NO (trip_no) INSERT t_trip (trip_no)
```

This means that the value of the attribute *trip_no* in the current longline set record must already exist in the parent table *t_trip* or the record will be rejected and the following message will be displayed:

```
*** User Error: insert constraint 'INVALID TRIP_NO' violation
```

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

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

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

3.2 Database Design

At the top level in this database, there is the table *t_trip* (Table 1). This contains the data from the **Trip Log** that the observers complete. Data in this table pertain to details about the trip as a whole, such as fishing gear arrangement, and observer's name. The attribute *call_sign* records the vessels radio callsign and is used to link to the table *t_vessel* in the tuna database (Figure 2). This table contains specifications about the longline vessel, such as vessel name, registered length, gross weight, freezer and hold capacities.

During the course of a fishing trip the vessel sets its long lines several times, with the observers job to record details about each set on the **Set Log**. Data from this log stored in the table *t_line_set*

[5] Also known as integrity checks.

(Table 2). Information, such as time of set, weather conditions, use of tori (bird) poles, and the physical dimensions of the line are recorded in this table.

There are three further tables to detail the hauling and the catch from the longline sets.

Just as each line is set, in must be hauled, with the observers recording hourly observations of the line haul in the **Hourly Haul Log**. This information is recorded in the table *t_haul* (Table 3). This hourly log allows changes in positions and conditions during hauling to de recorded. Hence, approximate positions of individual items caught can be calculated. The table also details the direction and speed of the vessel during the haul, which can be used to plot the haul if positions are unavailable.

Often, there are interruptions to the setting or hauling of a line. These interruptions are recorded by the observer into the **Events Log** and stored in the table *t_events* (Table 4). These events are stored as a code and are linked by a referential constraint to descriptions for each code as stored in the table *t_event_code* (Table 12). Note that the attribute *no_mins* recorded the duration of the event, but prior to 1991 it recorded the duration of the whole set.

The actual catch from the longline is recorded by the observers using the **Deck Log** and stored in the table *t_ctch_spec* (Table 5). This table holds details of individual item (not only fish but also sharks, birds, seals etc) caught, as identified by the attribute id, including trip and set number, species code, time landed, any damage to the specimen, length, weight, bait type caught on, and basket number. Given that each basket holds six hooks, the attribute *basket_no* helps to determine which hook caught the individual item. Positions of the individual items caught can be calculated by using the attribute *time_l* to extrapolate from records in the hour haul log table *t_haul*. Note that several of the attributes are codes, which are linked by referential constraints to specific code tables (Figure 3).

For each individual specimen, there may be up to eight samples taken for further analysis. Types of samples taken are also noted in *t_ctch_spec* by the attributes *sample_1* through to *sample_8*.

In addition to these five tables, there are two other tables that are used to record different strategies used by the vessel while fishing: snood length; and bait.

Snood length (length of line from the main line to an individual hook) for a range of sets are recorded into the Snood log and stored in the table *t_snoods* (Table 6). The hook colour is also recorded.

Similarly, the bait strategy used for a range of sets is recorded in the **Bait Log** and stored in the table *t_bait* (Table 7). The bait type is recorded as a code and is linked to a descriptive table *t_bait_code* (Table 9) by referential constraints.

The remaining ten tables (Tables 8 - 17) in the database contain descriptions for codes used for these seven main tables. These links to the main tables are shown in Figure 2 and Figure 3.

4 Table Summaries

This database can be broken down into seven main tables containing information on the longline tuna, ten other tables describing the various codes that are used.

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

1. **t_trip :** contains details pertaining to a longlining trip.
2. **t_line_set :** contains details of the setting of a longline.
3. **t_haul :** contains hourly records of vessel speed, position, and course as a longline is hauled.
4. **t_events :** contains records of interruptions to hauling a longline.
5. **t_ctch_spec :** contains record of individual items (fish, sharks, birds, etc) caught by the longline set and details any samples taken from them.
6. **t_snoods :** contains details of the snood arrangement used for a range of sets during a trip.
7. **t_bait :** contains details of the bait setting strategy used for a range of sets during a trip.

Also contained in the **l_line** database are numerous master tables containing all codes and their descriptions. Most of the codes used in the **l_line** database are checked against these master tables before being inserted. The following is a summary list of these master code tables:

8. **t_area_code :** contains numeric codes and definitions of the five main fishing areas. (See Appendix A)
9. **t_bait_code :** contains codes to describe the type of bait used.
10. **t_damage_code :** contains codes to describe the type of damage sustained to a landed specimen.
11. **t_event_code :** contains codes to describe the type of events that interrupt the hauling of a longline.
12. **t_handling_code :** contains codes to describe the nature of the handling of the specimen by the crew; i.e., what did they do with the specimen.
13. **t_life_code :** contains codes to describe the state of the life signs of a landed specimen.
14. **t_processed_code :** contains codes to describe the type of processing that was undertaken on a fish before the processed weight was taken.
15. **t_sample_code :** contains codes to describe what samples where taken from a landed specimen for further analysis.
16. **t_sp_status_code :** contains codes to describe the status of a species upon landing.
17. **t_weather_code :** contains codes to describe the weather conditions during a set or a haul of a longline.

5 l_line Tables

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

5.1 Table 1: t_trip

Comment: Profile information on all observed tuna longline trips.

Attributes	Data Type	Null?	Comment
trip_no	integer	No	Unique code assigned to each distinct observed trip
obs_trip_no	integer		Foreign key to the trip table in the observer database.
vessel_key	integer		Ministry of Fisheries generate number to uniquely identify each vessel
observer	character(20,1)		Name of the observer
vess_nat	character(1,1) smatch “[JKN]”		Code for the nationality of the observed vessel: J (Japan), K (Korea) or N (New Zealand)
vess_status	character(1,1) smatch “[CDF]”		Code for the fishing status of the observed vessel: F (foreign licensed), C (chartered) or D (domestic)
fishery	character(1,1) smatch “[SND]”		Fishery the vessel is licensed to fish S=Southern; N=Northern; D=Domestic
streamer	character(1,1) match “[YN]”		Indicates presence/absence of tori (bird) poles on the vessel
start_of_trip	date(5)		Date at start of the first set of trip
end_of_trip	date(5)		Date at end of the last haul of trip
snood_code	smallint range 1 or 2		Code describing pattern in snoods table, where 1=patterned and 2=random (for data up to 1992 inclusive)
comments	text(40,40,40,1)		Any information pertinent to the trip not included in the previous attributes that should be considered in analyses of data from this trip

Creator: pidba
Indices: UNIQUE BTREE ON (trip_no)
NORMAL (2, 15) BTREE ON (vess_status)

5.2 Table 2: t_line_set

Comment: Profile information on all observed sets of tuna longlines.

Attributes	Data Type	Null?	Comment
area_code	smallint		Code for area setting started in. Refer to the t_area_codes table
bird_area	smallint		Code for the bird area setting started in.
fma	smallint		Fisheries Management Area in which setting started in.
trip_no	integer	No	Unique code assigned to each distinct observed trip
set_no	smallint		Number assigned by observers to a distinct observed set
set_date	date(5) >= “19 June 1987”		Date at which the set started
mon	character(3,1)		Number of month of the year. Used to group data by calendar months.
fyr	integer		Fishing year. The amalgamation of two calender years. The fishing year runs from 1 Oct to 30 Sep.
target_sp	character(3,1)		Target species code. Refer to the curr_spp table in the rdb database
time_s	integer range 0 to 2359 match “*[0-5][0-9]”		Time at the start of the set
lat_s	integer range 2900 to 4900 match “*[0-5][0-9]”		Latitude at start of set (DDMM format)
long_s	integer range 16300 to 18000 match “*[0-5][0-9]”		Longitude at the start of the set
EW_s	character(1,1) smatch “[EW]”		long_s east/west of 180 line
time_f	integer range 0 to 2359 match “*[0-5][0-9]”		24-hour time at the finish of the set. If time_f is less that time_s it is assumed that the time_f is on the following day.
lat_f	integer range 2900 to 4900 match “*[0-5][0-9]”		Latitude at finish of set (DDMM format)

Attributes	Data Type	Null?	Comment
long_f	integer range 16300 to 18000 match “*[0-5][0-9]”		Longitude at the finish of the set
EW_f	character(1,1) smatch “[EW]”		long_f east/west of 180 line
lgth	integer <= 350		Length of line (kilometres)
basket_no	integer		Number of hooks on the line
hook_no	integer		Number of hooks on the line
hook_obs	integer		Number of hooks observed (generally less than hooks set where 12 hours haul duration is exceeded)
speed	decimal(3,1)		Speed of the vessel during the set (knots)
sig_time	decimal(2,1)		Snood signal time (seconds)
feed_rate	decimal(2,1)		Line feeder rate (m/sec)
buoy_lgth	smallint		Length between buoy at surface and connection to mainline below
min_depth	integer		Expected minimum depth of the line when set (metres)
max_depth	integer		Expected maximum depth of the line when set (metres)
ccamlr	character(1,1) match “[YN]”		Tori line design of ccamlr specifications
tori_used	character(1,1) match “[YN]”		Codes use/non-use of tori pole for set
stream_no	smallint		Number of streamers used in association with tori pole
stream_lgth	integer		Length of streamers used (metres)
stream_hgt	integer		Height of attachment above water (metres)
line_entry	character(1,1) match “[YN]”		Tori line over bait entry point
bait_stream	integer		Distance between bait landing point and tori line
bait_wake	character(1,1) match “[YN]”		Bait landing inside of vessel wake

Attributes	Data Type	Null?	Comment
bait_vessel	integer		Distance between bait landing point and vessel midline
bait_sink	integer		Distance behind vessel that bait sank (metres)
cloud	smallint range 0 to 100		Percentage cloud cover at start of the set
bar	decimal(5,1)		Barometer reading at start of the set
wind_d	integer range 0 to 359		Wind direction at start of the set (degrees)
wind_f	smallint range 0 to 12		Wind force at start of set (Beaufort scale)
weath_code	smallint		Code for prevailing weather conditions at start of set
bait_cond	character(1,1) smatch “[FT]”		Bait frozen or thawed
bait_thrower	character(1,1) match “[YN]”		Mechanical bait thrower used
no_vessels	integer		Number of vessels within 24nm radius
no_longliners	integer		Number of longliners within 24nm radius
time_obs	integer		Time of observation of set details
comments	text(40,40,40,1)		Other information pertinent to the set
perf	smallint		Performance flag for the line set: 1 = OK; 2 = Reject

Creator: pidba
Referential: INVALID TRIP_NO (trip_no) INSERT t_trip (trip_no)
INVALID AREA_CODE (area_code) INSERT t_area_code (area_code)
INVALID WEATHER (weath_code) INSERT t_weather_code (weath_code)
INVALID TARGET_SP (target_sp) INSERT rdb : curr_spp (code)
Indices: NORMAL (2, 15) BTREE ON (set_no)
UNIQUE BTREE ON (trip_no, set_no)
NORMAL (2, 15) BTREE lineset_perf_ndx ON (perf)

5.3 Table 3: t_haul

Comment: Hourly information of observed tuna longline hauls.

Attributes	Data Type	Null?	Comment
trip_no	integer	No	Unique code assigned to each distinct observed trip
set_no	smallint		Observed set to which the haul corresponds
haul_date	date(5) >= “19 June 1987”		Date on which haul was commenced
obs_time	integer range 0 to 2359 match “*[0-5][0-9]”		24-hour time of observation
lat	integer range 2900 to 4900 match “*[0-5][0-9]”		Latitude of vessel hauling at time of observation (DDMM format)
long	integer range 16300 to 18000 match “*[0-5][0-9]”		Longitude of vessel hauling at time of observation (DDDMM format)
EW	character(1,1) match “[EW]”		Longitude east or west of the 180 degree meridian
depth	integer		Depth of bottom at time of observation
temp_s	decimal(3,1)		Sea surface temperature at time of observation (Celsius to 1 decimal place)
speed	decimal(3,1)		Ships speed at time of observation (knots)
heading	integer range 0 to 359		Ships heading at time of observation (degrees)
wind_f	smallint range 0 to 12		Beaufort scale wind force at time of observation
wind_d	integer range 0 to 359		Wind direction at time of observation
end_hauled	smallint match “[12]”		Code describing at which end of the longline was hauled first: 1=the end that was set first; 2=the end that was set last

Attributes	Data Type	Null?	Comment
start_finish	character(1,1) match “[SF]”		1-char code to identify the first and last record for each haul: S=first record; F=final record
perf	smallint		Performance flag for the haul record: 1 = OK; 2 = Reject

Creator: pidba
Referential: INVALID SET_NO (trip_no, set_no) INSERT t_line_set (trip_no, set_no)
Indices: NORMAL (2, 15) BTREE ON (trip_no)
NORMAL (2. 15) BTREE ON (set_no)
NORMAL (2, 15) BTREE haul_perf_ndx ON (perf)

5.4 Table 4: t_events

Comment: Profile of events affecting haul/observations.

Attributes	Data Type	Null?	Comment
trip_no	integer	No	Unique code assigned to each distinct observed trip
set_no	smallint		Observed set to which the described event applies
event_code	smallint		Code for the described event started
time_s	integer range 0 to 2359 match "*[0-5][0-9]"		24-hr time at which the event range (NZST)
no_mins	integer		Number of minutes described event lasted for
comments	text(40,40,40,1)		Other relevant factors

Creator: pidba
Referential: INVALID SET_NO (trip_no, set_no) INSERT t_line_set (trip_no, set_no)
INVALID EVENT_CODE (event_code) INSERT t_event_code (event_code)
Indices: NORMAL (2, 15) BTREE ON (trip_no)
NORMAL (2, 15) BTREE ON (set_no)
NORMAL (2, 15) BTREE ON (event_code)

5.5 Table 5: t_ctch_spec

Comment: Description of catches of specimens (fish, birds, seals, etc) made by tuna longlines.

Attributes	Data Type	Null?	Comment
id	longinteger		Unique identification number assigned to each specimen
trip_no	integer	No	Unique code assigned to each distinct observed trip
set_no	smallint		Observed set to which the following data applies
sample_no	integer		Number assigned by observer to samples where taken
sp_code	character(3,1)		3-char species code for each specimen recorded. Refer rdb:curr_spp
time_l	integer range 0 to 2359 match "*[0-5][0-9]"		24-hour time observer recorded specimen as being landed (NZST)
sp_status_code	smallint		Status/condition of the specimen
life_code	character(1,1)		1-char code to denote the level of the specimens life signs (used from 1992)
handling	character(1,1)		1-char code to denote the crews handling of the specimen (used from 1992)
damage_old	character(1,1)		1-char code to describe the type severity of damage to a specimen (used up to the 1991 season, from 1992 refer to the attribute "damage")
damage	smallint		Numeric code for the type of damage (caused by driftnets, shark bites, etc) on specimens (used from 1992)
no	smallint		Tally of number caught for any given species when individuals not recorded separately
fork_lgth	integer		Fork length (cm) of the specimen. Except: SWO, STM, SSF = lower jaw to fork; THR, BET, POS = total length
length2	integer		SWO, STM, SSF = eye to fork; THR, BET, POS = standard length (cm)
GW	integer		Greenweight of specimen (kg)

Attributes	Data Type	Null?	Comment
proc_code	character(2,1)		2-char code to indicate type of processing done on the specimen
proc_weight	integer		Processed weight (kg) of the specimen
sex	character(1,1)		1-digit code for the sex of the specimen. Refer rdb:t_sex_codes
basket	integer		Basket (of hooks) number in which specimen was caught
bait_code	smallint		Code for type of bait specimen was caught on
sample_1	smallint		Code for 1st sample taken from specimen
sample_2	smallint		Code for 2nd sample taken from specimen
sample_3	smallint		Code for 3rd sample taken from specimen
sample_4	smallint		Code for 4th sample taken from specimen
sample_5	smallint		Code for 5th sample taken from specimen
sample_6	smallint		Code for 6th sample taken from specimen
sample_7	smallint		Code for 7th sample taken from specimen
sample_8	smallint		Code for 8th sample taken from specimen
true_sp	character(3,1)		3-char species code as identified by the Natural History Museum. Refer to the curr_spp table in the rdb database
type	smallint		Data type: 1=observed, 2=tallied, 3=prior to start of observations, 4=after end of observations, 5=missed at unknown time during haul
perf	smallint		Performance flag for the catch specimen record: 1 = OK; 2 = Reject

Creator: pidba

Referential:
INVALID SET_NO (trip_no, set_no) INSERT t_line_set (trip_no, set_no)
INVALID SP_CODE (sp_code) INSERT rdb : curr_spp (code)
INVALID STATUS (sp_status_code) INSERT t_sp_status_code (sp_status_code)
INVALID LIFE_CODE (life_code) INSERT t_life_code (life_code)
INVALID HANDLING (handling) INSERT t_handling_code (handling)
INVALID DAMAGE (damage) INSERT t_damage_code (damage)
INVALID PROC_CODE (proc_code) INSERT t_processed_code (proc_code)
INVALID BAIT_CODE (bait_code) INSERT t_bait_code (bait_code)
INVALID SEX (sex) INSERT rdb : t_sex_codes (sex_code)
INVALID SAMPLE 1 (sample_1) INSERT t_sample_code (samp_code)
INVALID SAMPLE 2 (sample_2) INSERT t_sample_code (samp_code)
INVALID SAMPLE 3 (sample_3) INSERT t_sample_code (samp_code)

```
                INVALID SAMPLE 4 (sample_4) INSERT t_sample_code (samp_code)
                INVALID SAMPLE 5 (sample_5) INSERT t_sample_code (samp_code)
                INVALID SAMPLE 6 (sample_6) INSERT t_sample_code (samp_code)
                INVALID SAMPLE 7 (sample_7) INSERT t_sample_code (samp_code)
                INVALID SAMPLE 8 (sample_8) INSERT t_sample_code (samp_code)
                INVALID TRUE_SP (true_sp) INSERT rdb : curr_spp (code)
Indices:        UNIQUE BTREE ON (id)
                NORMAL (2, 15) BTREE ON (sp_code)
                NORMAL (2, 15) BTREE ON (trip_no)
                NORMAL (2, 15) BTREE ON (set_no)
                NORMAL (2, 15) BTREE ctchspec_perf_ndx ON (perf)
```

5.6 Table 6: t_snoods

Comment: Profile on the snood arrangement strategy used on a range of tuna longline sets.

Attributes	Data Type	Null?	Comment
trip_no	integer	No	Unique code assigned to each distinct observed trip
snood_no	smallint range 1 to 30		Snood number to which the data applies, corresponds to bait_no in the bait table
set_no_s	smallint		Starting set to which snood arrangement applies
set_no_f	smallint		Final set to which snood arrangement applies
lgth_t	smallint range 6 to 50		Total length (m) of the identified snood
hook_colour	character(15,1)		Colour of the hook on the snood
hook_type	character(10,1)		Type of hook on the snood
baskets	character(20,1)		Brief description of the range of baskets to which arrangement applies, if blank: applies to all baskets

Creator: pidba
Referential: INVALID TRIP_NO (trip_no) INSERT t_trip (trip_no)
Indices: NORMAL (2, 15) BTREE ON (trip_no)

5.7 Table 7: t_bait

Comment: Profile on the bait strategy used on a range of tuna longline sets.

Attributes	Data Type	Null?	Comment
trip_no	integer	No	Unique code assigned to each distinct observed trip
set_no_s	smallint		Starting set for described bait strategy
set_no_f	smallint		Final set for the described bait strategy
bait_no	smallint range 1 to 30		Bait number from the start of the basket corresponds to snood_no from snoods table
bait_code	smallint		Code for type of bait used
baskets	character(20,1)		Brief description of the range of baskets to which arrangement applies, if blank: applies to all baskets

Creator: pidba
Referential: INVALID TRIP_NO (trip_no) INSERT t_trip (trip_no)
INVALID BAIT_CODE (bait_code) INSERT t_bait_code (bait_code)
Indices: NORMAL (2, 15) BTREE ON (trip_no)

5.8 Table 8: t_area_code

Comment: Area codes used in database t_line_set table to define main regions of fishing effort for southern bluefin and bigeye tuna.

Attributes	Data Type	Null?	Comment
area_code	smallint	No	Code used in main tables of database used to define main fishing areas
area	text(40,40,40,1)	No	Description of the main fishing areas defined by area_code

Creator: pidba
Indices: UNIQUE BTREE ON (area_code)

5.9 Table 9: t_bait_code

Comment: Bait codes used in database.

Attributes	Data Type	Null?	Comment
bait_code	smallint	No	Code for type of bait used
bait_type	character(26,1)	No	Description of bait code

Creator: pidba
Indices: UNIQUE BTREE ON (bait_code)

5.10 Table 10: t_damage_code

Comment: Damage codes used in database to describe cause of damage to individual catch specimens on landing.

Attributes	Data Type	Null?	Comment
damage	character(1,1)	No	Numeric code to denote the cause of damage to landed specimens
damage_type	character(20,1)	No	Description of cause of damage

Creator: pidba
Indices: UNIQUE BTREE ON (damage)

5.11 Table 11: t_event_code

Comment: Event codes used in database to describe interruptions to hauling and observations of the hauling.

Attributes	Data Type	Null?	Comment
event_code	smallint	No	Code used in main tables of database to describe interruptions to the haul and observations of the haul
event	character(18,1)	No	Description of events defined in event_code

Creator: pidba
Indices: UNIQUE BTREE ON (event_code)

5.12 Table 12: t_handling_code

Comment: Specimen handling codes used in database.

Attributes	Data Type	Null?	Comment
handling_code	character(1,1)	No	Code for the crews handling of a landed specimen
handling	character(20,1)	No	Description of handling code

Creator: pidba
Indices: UNIQUE BTREE ON (handling)

5.13 Table 13: t_life_code

Comment: Specimen life sign codes used in database.

Attributes	Data Type	Null?	Comment
life_code	character(1,1)	No	Code for life signs of specimen landed
life_signs	character(20,1)	No	Description of life signs code

Creator: pidba
Indices: UNIQUE BTREE ON (life_code)

5.14 Table 14: t_processed_code

Comment: Fish processed codes used in database.

Attributes	Data Type	Null?	Comment
proc_code	character(2,1)	No	Code for fish processed type that was weighed
proc_type	character(23,1)	No	Description of process code

Creator: pidba
Indices: UNIQUE BTREE ON (proc_code)

5.15 Table 15: t_sample_code

Comment: Sample codes used in database to describe the type of sample taken from a specimen in the t_ctch_spec table.

Attributes	Data Type	Null?	Comment
samp_code	smallint	No	Code used in main tables of database to describe type of sample taken from specimen in t_ctch_spec table
sample	character(20,1)	No	Description of sample taken

Creator: pidba
Indices: UNIQUE BTREE ON (samp_code)

5.16 Table 16: t_sp_status_code

Comment: Species status codes used in database.

Attributes	Data Type	Null?	Comment
sp_status_code	smallint	No	Code for the status/condition of specimen upon landing
status	character(21,1)	No	Description of status of specimen

Creator: pidba
Indices: UNIQUE BTREE ON (sp_status_code)

5.17 Table 17: t_weather_code

Comment: Weather codes used in database.

Attributes	Data Type	Null?	Comment
weath_code	smallint	No	Code for weather conditions
weather	character(15,1)	No	Description of weather code

Creator: pidba
Indices: UNIQUE BTREE ON (weath_code)

6 l_line business rules

6.1 Introduction to business rules

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

There are three recognised types of business rules:

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

Fact rules are shown on the ERD by the cardinality (e.g., one-to-many) of table relationships. Formula and Validation rules are implemented by referential constraints, range checks, and algorithms both in the database and during validation.

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

6.2 Summary of rules

Trip details (t_trip)

trip_no	Must be not null and an integer greater than zero.
obs_trip_no	Must be a valid observer trip number as listed in either the *observer_trip* or *new_observer_trip* tables in the **obs** database.
vessel_key	Must be a valid Ministry of Fisheries vessel key number.
observer	Must not be null.
vess_nat	Must be one character that is either a "J", "K" or "N".
vess_status	Must be one character that is either a "F", "C" or "D".
fishery	Must be one character that is either a "S", "N" or "D".
streamer	No longer used
start_of_trip	Must be a valid date on, or after, 19 June 1987 and can not exceed current date.
end_of_trip	Must be a valid date on, or after, 19 June 1987 and can not exceed current date.
	Multiple column checks on trip dates: The trip start date must not be greater than the trip end date.
snood_code	No longer used

Longline set table (t_line_set)

area_code Must be a valid area code as listed in the *t_area_code* table.

bird_area Must be an integer between the range of 1 to 4 inclusive.

fma Must be an integer between the range of 1 to 10 inclusive.

trip_no Must be a valid observer longline trip number as listed in the *t_trip* table.

set_no Must be an integer greater than zero.

set_date Must be a valid date on, or after, 19 June 1987 and can not exceed current date.

Multiple column checks on trip dates and set date:
Longline set date must be on or after the trip start date, and on or before the trip end date.

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

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

lat_s Must be an integer that represents a valid latitude with the implied value of minutes not exceeding 59 and should be within the reasonable range of 2400 to 4900.

long_s Must be an integer that represents a valid longitude with the implied value of minutes not exceeding 59 and should be within the reasonable range of 16300 to 18000.

EW_s Must be one character that is either a "E" or "W".

time_f Finish time of the longline set must be a valid 24-hour time and fall within the range of 0 – 2359 hours

lat_f Must be an integer that represents a valid latitude with the implied value of minutes not exceeding 59 and should be within the reasonable range of 2400 to 4900.

long_f Must be an integer that represents a valid longitude with the implied value of minutes not exceeding 59 and should be within the reasonable range of 16300 to 18000.

EW_f Must be one character that is either a "E" or "W".

lgth Must be an integer between the range of 0 to 350.

basket_no Must be an integer between the range of 1 to 800.

Longline set table (t_line_set) cont….

hook_no	Must be an integer between the range of 1 to 4000.
hooks_obs	Must be an integer between the range of 0 to 4000
	Multiple column checks on *hook_no* and *hooks_obs*: The number of hooks observed must be less than or equal to the total number of hooks in a longline set.
speed	Must be a number between 2 and 15.
sig_time	Must be a number between 3 and 15.
feed_rate	Must be a number between 2 and 10.
buoy_lgth	Must be a number between 5 and 60.
min_depth	Must be a number between 5 and 350
max_depth	Must be a number between 5 and 350.
	Multiple column checks on minimum and maximum longline depths: Minimum longline set depth must be less than or equal to the maximum longline set depth.
ccamlr	Must be one character that is either a "Y" or "N".
tori_used	Must be one character that is either a "Y" or "N".
stream_no	Must be an integer between 0 and 100.
tori_line_lgth	Must be an integer between 10 and 350.
tori_att_hgt	Must be an integer between 1 and 20.
line_entry	Must be one character that is either a "Y" or "N".
bait_stream	Must be an integer between 0 and 20.
bait_wake	Must be one character that is either a "Y" or "N".
bait_vessel	No longer used
bait_sink	No longer used
cloud	Must be an integer between the range of 0 to 100.
bar	Must be an integer between 935 and 1045.

Longline set table (t_line_set) cont….

wind_d	Must be an integer between the range of 0 to 359.
wind_f	Must be an integer between the range of 0 to 12.
weath_code	No longer used, refer to the attribute *l_line* in the table *t_weath_code*.
bait_cond	Must be one character that is either a "F", "T" or "I".
bait_thrower	Must be one character that is either a "Y" or "N".
no_vessels	Must be an integer between the range of 0 to 20.
no_longliners	Must be an integer between the range of 0 to 20.
time_obs	Time of observation must be a valid 24-hour time and fall within the range of 0 – 2359 hours **Multiple column checks on longline set start time and observation time:** Time of observation must be on or after the start time of the longline set.
perf	Must be equal to either "0" or "1".

Longline haul table (t_haul)

	Multiple column checks on *trip_no* and *set_no*: The combination of *trip_no* and *set_no* must exist in the *t_line_set* table.
haul_date	Must be a valid date on, or after, 19 June 1987 and can not exceed current date. **Multiple column checks on trip dates and haul date:** Longline haul date must be on or after the trip start date, and on or before the trip end date. **Multiple column checks on longline set date and haul date:** Longline haul date must be on or after the longline set date.
obs_time	Time of observation must be a valid 24-hour time and fall within the range of 0 – 2359 hours.
lat	Must be an integer that represents a valid latitude with the implied value of minutes not exceeding 59 and should be within the reasonable range of 2400 to 4900.
long	Must be an integer that represents a valid longitude with the implied value of minutes not exceeding 59 and should be within the reasonable range of 16300 to 18000.
EW	Must be one character that is either a "E" or "W".
depth	Must be an integer between 50 and 6000.
temp_s	Must be a number between 5 and 27.
speed	Must be a number between 0 and 15.
heading	Must be an integer between 0 and 359.
wind_f	Must be an integer between 0 and 12.
wind_d	Must be an integer between 0 and 359.
end_hauled	Must be equal to either "0" or "1".
start_finish	Must be one character that is either a "S", "F", "O" or "L".
perf	Must be equal to either "0" or "1".

Events table (t_events)

Multiple column checks on *trip_no* and *set_no*:
The combination of *trip_no* and *set_no* must exist in the *t_line_set* table.

event_code Must be a valid event code as listed in the *t_event_code* table.

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

no_mins Must be an integer greater than or equal to zero and should fall within the reasonable range of 1 to 1440.

Catch and specimen table (t_ctch_spec)

id Must be an unique not null integer greater than zero.

Multiple column checks on *trip_no* and *set_no*:
The combination of *trip_no* and *set_no* must exist in the *t_line_set* table.

sample_no Must be an integer greater than zero

Multiple column checks on *trip_no* and *sample_no*:
Sample numbers should be unique within each trip.

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

time_l Time specimen landed on the deck must be a valid 24-hour time and fall within the range of 0 – 2359 hours.

sp_status_code No longer used, pre-1992 only, refer to the *t_sp_status_code* table.

life_code Must be a valid life code as listed in the *t_life_code* table.

handling Must be a valid handling code as listed in the *t_handling_code* table.

damage_old No longer used

damage Must be a valid damage code as listed in the *t_damage_code* table.

no integer, greater than 0

fork_lgth Must be an integer between the range of 1 and 800.

Multiple column checks on species code and fork length:
The fork length should be less than the maximum length of the species as listed in the *curr_spp* table in the **rdb** database.

length2 Must be an integer between the range of 1 and 800.

Multiple column checks on species code and *length2*:
Other specimen lengths should be less than the maximum length of the species as listed in the *curr_spp* table in the **rdb** database.

GW Must be an integer between the range of 1 and 450.

Multiple column checks on species code and green weight:
The green weight should be within the reasonable limits for the species code as listed in Appendix 2.

proc_code Must be a valid processing code as listed in the *t_processed_code* table.

proc_weight Must be an integer between 1 and 280.

Catch and specimen table (t_ctch_spec) cont…

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

basket No longer used.

bait_code No longer used, refer to the *t_bait_code* table.

sample_1-8 Must be a valid sample code as listed in the *t_sample_code* table.

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

perf Must be equal to either “0” or “1”.

Snoods strategy table (t_snoods)

trip_no Must exist and be a valid observer longline trip number as listed in the *t_trip* table.

snood_no Must be an integer between the range of 1 to 30.

set_no_s Must be an integer greater than zero.

set_no_f Must be an integer greater than zero.

Multiple column checks on *set_no_s* and *set_no_f*:
The finish set number must be the same as, or after, the start set number.

lgth_t Must be an integer between the range of 6 to 50.

Bait strategy table (t_bait)

trip_no Must exist and be a valid observer longline trip number as listed in the *t_trip* table.

set_no_s Must be an integer greater than zero.

set_no_f Must be an integer greater than zero.

Multiple column checks on *set_no_s* and *set_no_f*:
The finish set number must be the same as, or after, the start set number.

bait_no Must be an integer between the range of 1 to 30.

bait_code Must be a valid bait code as listed in the *t_bait_code* table.

Area codes table (t_area_code)

area_code Must exist and be an integer between 1 and 127.

area Must exist.

Bait codes table (t_bait_code)

bait_code Must exist and be an integer between 1 and 127.

bait_type Must exist.

Damage codes table (t_damage_code)

damage_code Must exist, unique, and only 1 character in length..

damage_type Must exist.

Event codes table (t_event_code)

event_code Must exist and be an integer between 1 and 127.

event Must exist.

Handling codes table (t_handling_code)

handling_code Must exist, unique, and only 1 character in length..

handling Must exist.

Life sign codes table (t_life_code)

life_code Must exist, unique, and only 1 character in length..

life_signs Must exist.

Fish processing codes table (t_processed_code)

proc_code Must exist, unique, and only 2 characters in length..

proc_type Must exist.

Sample codes table (t_sample_code)

samp_code Must exist and be an integer between 1 and 127.

sample Must exist.

Specimen status codes table (t_sp_status_code)

sp_status_code Must exist and be an integer between 1 and 127.

status Must exist.

Weather codes table (t_weather_code)

weath_code Must exist and be an integer between 1 and 127.

weather Must exist.

Appendix 1 - Map of Observer southern bluefin tuna areas

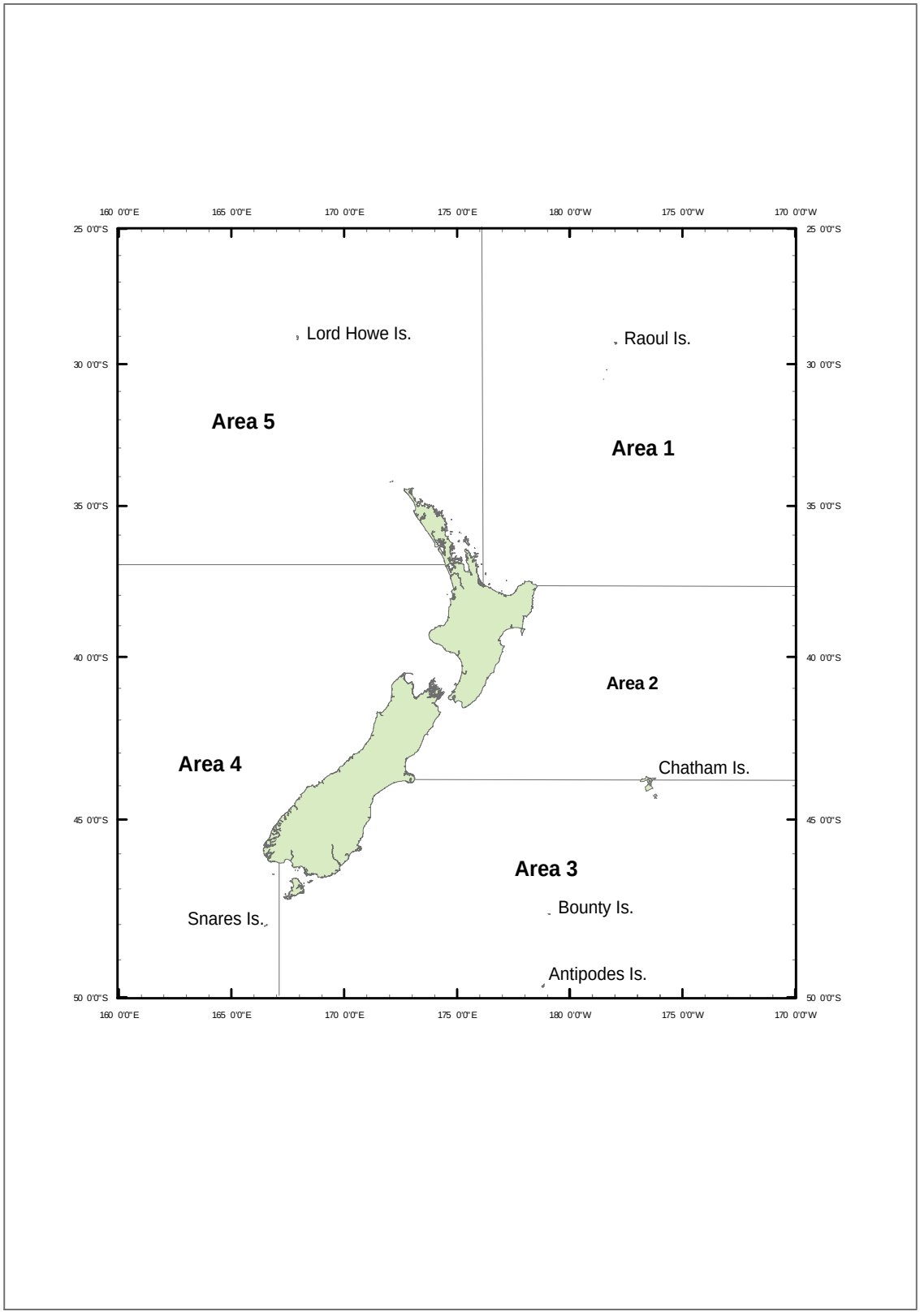

Figure 2: Map showing the longline tuna Observer areas

Appendix 2 – Reference Code Tables

Green weight ranges by species code

Species Code	MIN(GW)	MAX(GW)
ABR	1	5
AGR	5	10
ALB	1	30
BAR	1	8
BAS	1	6
BBA	1	5
BDA	1	5
BET	68	200
BIG	12	150
BNS	2	9
BRA	4	5
BSH	1	3
BSP	1	21
BTU	4	117
BWH	85	195
BWS	1	237
CAR	2	8
CYL	1	5
CYO	1	118
CYP	1	66
DAS	1	10
DEA	1	20
DOF	1	7
DPO	1	15
DWD	1	14
EMA	1	8
FAN	1	1
FTU	2	8
FUR	30	61
HAK	2	34
HAP	1	3
HHS	8	8
HOK	1	5
HPB	3	9
KIN	1	19
LAT	1	102
LCA	5	6
LEP	3	50
MAK	2	248
MEZ	5	15
MOO	4	66
NEX	1	1
NTU	7	242
OFH	1	45
PAH	10	33
PLS	3	4
POS	1	164
RAG	2	2
RAY	7	113
RBM	1	18
RUD	1	98
SAW	0	0
SCH	3	142

Species Code	MIN(GW)	MAX(GW)
SEV	54	54
SHA	2	75
SKA	7	7
SKI	1	1
SKJ	1	9
SLB	10	10
SPD	1	13
SSF	17	20
STM	11	142
STN	10	215
STO	12	12
STR	3	3
STU	5	144
SUN	10	250
SWO	5	341
TAS	1	3
THR	53	410
TJA	10	10
WIN	0	2
WWA	6	6
XBM	3	6
XGP	1	15
XKM	6	6
XPE	1	1
XRA	5	15
XWA	5	16
XWC	1	2
XWM	2	6
YFN	4	68
ZEL	1	3