Database documentation for the Ministry for Primary Industries, Regional Catch Effort database regional_ce

B. M. Sanders

NIWA Fisheries Data Management
Database Document Series

January 2019

Revision History

Version	Status	Changed by	Reason	Date
1.0	Release	B. Sanders	New documentation.	October 2005
2.0	Update	D Fisher, F Wei	Postgres version	December 2015
2.1	Update	D Fisher	Added ui to t_trip	Feb 2019

Contents

List of Figures

1 Introduction to the Database Document series

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

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

This document provides an introduction to the Regional Catch Effort database **regional_ce**, and is part of the database documentation series produced by NIWA.

All documents in this series include a summary of the database design, a description of the main data structures accompanied by an Entity Relationship Diagram (ERD), and a listing of all the main tables. The ERD graphically shows how all the tables link together.

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

Access to this database is restricted to specific nominated personnel as specified in the current Data Management contract between the Ministry and NIWA. Any requests for data should in the first instance be directed to the Ministry.

2 Regional Catch Effort data

The **regional_ce** database holds commercial catch effort data provided from New Zealand vessels fishing on the high seas and furnishing non-NZ Catch Effort forms. The **regional_ce** database is required to store catch, effort, landings, processing and environment information provided to the Ministry by commercial fishers or their agents. Current data sets stored are provided from the South Pacific Regional Purse-seine Fishery and information collected under the Convention for the Conservation of the Antarctic Marine Living Resources (CCAMLR) to aid in the management of the Antarctic marine resources, provided from Bottom Longline vessels fishing for toothfish.

Of these data:

- Catch data are the catch (kg of each species) recorded by the vessel as they fish.
- Effort data summarise the amount of effort that a vessel put into catching fish, including the method that was used, position, date, times and the species targeted where applicable. The South Pacific Regional Purse-seine does not record target species.
- Landings data summarise either the actual quantity of fish landed at a wharf (or transferred to another vessel at sea). South Pacific Regional Purse-seine data only.
- Environment data are provided only by CCAMLR and summarise water depth and bottom type where fishing taking place.

Information is received from CCAMLR trips electronically, currently as Microsoft Excel spreadsheets. South Pacific Regional Purse-seine data has also been received electronically in Excel spreadsheets since about December 2011.

Information from the South Pacific Regional Purse-seine fishers is supplied on several forms types:

1. **South Pacific Regional Purse-seine Logsheet** This is a Catch Effort Landing Return for the fishery. The form consists of a header section at the top for vessel and trip details, a number of rows to record catch and effort information, and a landings section in the lower part of the form. Generally only one form type is used, except for a variation of the form when fishing in Kiribati waters. A separate form is required for each change in Fishing permit or licence number used.
2. **Purse-seine Activity and catch report form – Kiribati Waters** This form is very similar to the first one in appearance, with some slight variations in the data fields collected, e.g. Effort includes Duration (hrs) and Aerial Search (Y/N).

The **regional_ce** database is designed to store these different data sets in a central, managed Relational Data Base Management System (RDBMS), as well as to provide the ability to pre-emptively hold data from other regional fisheries that the Research Provider may be required to maintain in future times.

3 Data Structures

3.1 Table relationships

This database contains a number of tables. The ERD for **regional_ce** (Figure 1) shows the logical structure[1] of the database and it's entities (each entity is implemented as a database *table)* and the relationships between these tables. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed. Each table represents an object, event, or concept in the real world that is selected to be represented in the database. Each *attribute* of a table is a defining property or quality of the table. All of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary key[2].

Note that Figure 1 shows the main tables only. Note that most tables contain foreign keys[3]. These foreign keys define the relationships between the tables in **regional_ce**.

[1] Also known as a database *schema.*
[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 an attribute or a combination of attributes that is a primary key in another table.

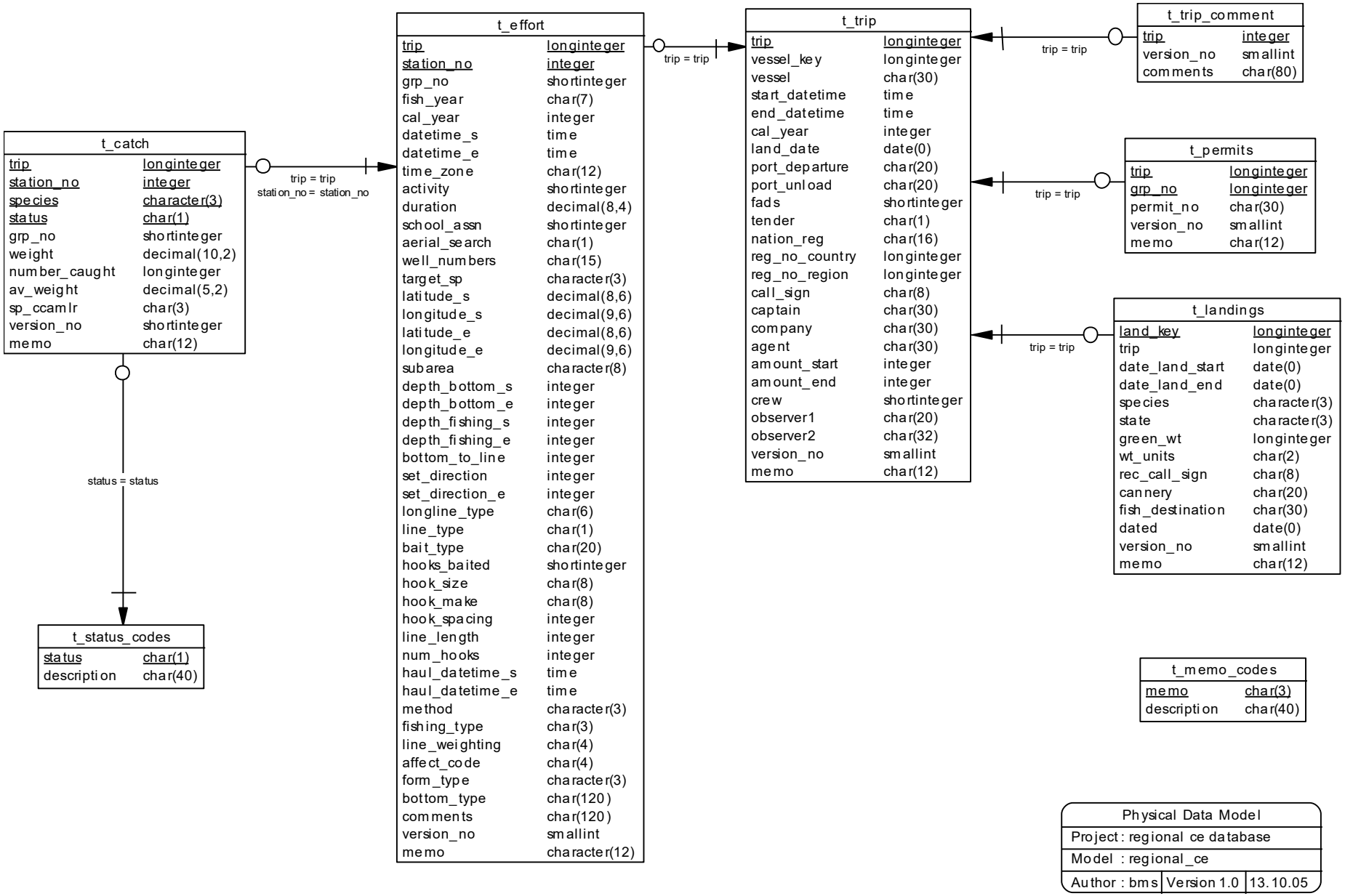

Figure 1 Entity Relationship Diagram (ERD) for the regional_ce database

The **regional_ce** database is implemented as a relational database; i.e., each table is a special case of the mathematical construct known as a *relation* and hence elementary relation theory is used to deal with the data within tables and the relationships between them. There are three types of relationships possible between tables, but only one exists in **regional_ce**: one-to-many[4]. These relationships can be seen in ERDs by connecting a single line (indicating "many") from the child table; e.g., *t_catch*, to the parent table; e.g., *t_effort,* with an arrowhead (indicating "one") pointing to the parent.

Every relationship has a mandatory or optional aspect to it. If a relationship is mandatory, then it has to occur at least once, while an optional relationship might not occur at all. For example, in Figure 1, consider that relationship between the table *t_effort* and it's child table *t_catch.* The symbol "O" by the child *t_catch* means that a fishing event can have zero or many catch records, while the bar by the parent *t_effort* means that for every catch there must be a matching fishing event record.

These links are enforced by foreign key constraints[5]. Foreign key constraints do not allow *orphans* to exist in any table; i.e., where a child record exists without a related parent record. This may happen when: a parent record is deleted; the parent record is altered so the relationship is lost; or a child record is entered without a parent record.

Foreign keys are shown in the table listings by the following format:

```
Referential:      constraint name (attribute[, attribute])  |INSERT|
                                                            |DELETE|
                  parent table (attribute[, attribute])
```

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

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

```
Referential:      Invalid trip stn ver (trip, station_no, version_no) INSERT
                  t_effort (trip, station_no, version_no)
```

This means that the value of the combination of attributes *trip, station_no* and *version_no* in the current record must already exist in the parent table *t_effort* or the record will be rejected and the following message will be displayed:

```
*** User Error: insert constraint "Invalid trip stn ver" violation
```

Section 5 lists all the **regional_ce** tables as implemented by the Postgres RDBMS. As can be seen in the listing of the tables, each table has a primary key on it. Primary keys are generally listed using the following format:

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

[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 fishing event in *t_effort* can have many catches in *t_catch* but one catch can only come from one fishing event.

[5] Also known as referential or integrity checks.

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

The database listing (Tables 1-8) show that the tables also have indices on some attributes. That is, attributes that are most likely to be used as a searching key have like values linked together to speed up searches. These indices are listed using the following format:

`Indices:` index_name btree (*attribute*)

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

3.2 Database design

The design of the **regional_ce** database incorporates table structures and field names to allow data extracts to be compatible with data extracted from the Ministry of Fisheries Catch Effort **warehou** database. Each vessel is identified by the same unique *vessel_key* attribute, as would be used in the **warehou** database. The same concept of a trip is used, with the start and end dates of trips being provided by fishers, to define a trip. The *trip* attribute is assigned a 'trip key' to link all records in the various tables for a single trip. All the main tables have a *version_no* field, to allow for meeting any requirements to update existing data supplied by fishers, although at the time of writing, only one version exists for all forms entered into the database.

The data from both the South Pacific Regional Purse-seine fishery and the CCAMLR electronic logbooks are submitted on a trip basis. The table *t_trip* stores all the constant information for a trip in one record, these details can be linked with various other tables in the regional_ce database for a given trip, using the *trip* (trip number) attribute. For the CCAMLR data set the trip number assigned by the observer programme has been used. The South Pacific Regional Purse-seine data do not have distinctive trip numbers associated with the individual trips received, therefore a system assigned number (started from 10 001) is issued in sequence of the data being loaded.

Each Purse-seine form contains header information that is constant on a trip basis, (repeated on each page used), except the "fishing permit or licence number" which will change if more than one permit fished within a trip, therefore all the header data is stored in the *t_trip* table, except for the 'fishing permit' that is stored in a separate *t_permits* table. The permit number is linked by the *trip* attribute and to the catch, effort records by a system assigned 'group number', stored in the *grp_no* attribute. The *grp_no* represents a group of stations recorded on a form, covered by the particular 'permit number', as held in the *permit_no* attribute of the *t_permits* table. In this context, a group can consist of one or more pages, as labelled page __ of __ on the paper forms.

The table *t_trip_comment* is used to store any remarks or comments relating to the overall trip.

The second main table is *t_effort*. Details stored include: dates, times, and location of fishing events, the fishing method used and various physical parameters about the gear used (e.g., no of hooks used, length of line, etc.). The Purse-seine logsheet includes an activity code that indicates either a fishing set was made, or defines the main activity for the day, if no fishing sets have been made. When the activity (stored in the *activity* attribute in *t_effort*), shows a fishing set has not been made by the vessel during that day, the position recorded should be the latitude and longitude at 01:00 UTC on

the South Pacific Regional Purse-seine logsheet (form type = PUR) or noon for Kiribati waters (form type = KIR). Because *t_effort* has to handle several fishing types, the meaning of various *t_effort* attributes may change depending on the form type and fishing method used. In a number of cases in the *t_effort* table, attributes will only be populated for one method, e.g. *num_hooks* is only populated for the Bottom Longline data set. Further to this, some Bottom Longline effort data were collected for restricted periods only, e.g. set direction for both the start and end of set was collected for a limited numbers of trips only (one season), most trips do not have an end set direction recorded. A method can be reported on several form types, hence the *form_type* is stored in the database, as is the case for the Purse-seine method.

Details of the catch by species, recorded by the vessel, are stored in the *t_catch* table. Details include species code, a catch weight and the number caught where this is provided. The *status* attribute is used to indicate the destination type of the fish recorded on this record for the set. For Purse-seine catch the discard code (stored in the attribute *status),* incorporates the reason for the discard, this is coded 1, 2, 3, 4 or 5 as defined in the appendix.

The species codes used in the CCAMLR catch data as supplied differ from the standard NZ species codes used by the Ministry of Fisheries and NIWA. For instance the CCAMLR code for Patagonian toothfish (*Dissostichus eleginoides*) is TOP, while the NZ code is PTO and TOP is the NZ code for Pale Toadfish (*Ambophthalmos angustus*). The original CCAMLR species code is retained in the *t_catch* table in the *sp_ccamlr* attribute.

Landing details are stored in the table *t_landings* (Table 4). These details come from the South Pacific Regional Purse-seine PUR and KIR form types. Details include species, and landed weight. Landings for a single species may be split, for example between several canneries or destinations, therefore a system generated number is assigned to uniquely identify each landing record, this key is stored in the attribute *land_key*. The CCAMLR electronic logbook data for NZ bottom longline vessels does not include any landings information.

4 Table Summaries

The following is a listing of the tables contained in the **regional_ce** database:

1. **t_trip**: trip information.
2. **t_trip_comment**: trip comments
3. **t_effort**: contains details of fishing effort
4. **t_catch**: contains details of each species caught. including green weight
5. **t_permits**: records permit or licence numbers being fished for a trip.
6. **t_landings**: contains details about a landing or transhipment by species and processed state.
7. **t_memo_codes**: contains descriptions for memo codes used in main tables.
8. **t_status_codes**: contains descriptions for status used for the catch.

5 regional_ce Tables

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

5.1 Table 1: t_trip

Comment: Trip information for regional purse-seine and CCAMLR BLL trips.

Column	Type	Null?	Description
trip	integer	No	Trip identification number for regional purse-seine and CCAMLR BLL trips.
vessel_key	integer		MFish id key for vessel.
vessel	character varying(30)		Vessel name.
start_datetime	timestamp without time zone		Trip start date and departure time.
end_datetime	timestamp without time zone		Trip end date and arrival time in port.
cal_year	integer		Calendar year.
land_date	date		Trip landing date.
port_departure	character varying(30)		Port of departure.
port_unload	character varying(30)		Port of unloading.
fads	smallint		For Purse-seine, number of fads used .
tender	character varying(1)		Purse-seine tender used. Y or N.
nation_reg	character varying(16)		Country of registration.
reg_no_country	integer		Registration number in country of registration.
reg_no_region	integer		FFA regional registration number.
call_sign	character varying(8)		International radio call sign.
captain	character varying(30)		Name of vessels captain.
company	character varying(30)		Company name.
agent	character varying(30)		Name of agent in port.
amount_start	integer		Amount of fish onboard at start of the trip.

amount_end	integer		Amount of fish onboard after unloading.
crew	smallint		Number of crew - KIR forms only.
observer1	character varying(20)		Name of observer 1.
observer2	character varying(32)		Name of observer 2.
version_no	smallint	No	Version number of record.
memo	character varying(12)		Memo code that tracks any changes to the record, refer t_memo_codes table.

Indexes:
"pk_t_trip" PRIMARY KEY, btree (trip, version_no)
"ui_trip" UNIQUE, btree (call_sign, start_datetime)

5.2 Table 2: t_trip_comment

Comment: Overall comments for a trip.

Column	Type	Null?	Description
trip	integer		Trip identification number.
version_no	smallint	No	Version number of record.
comments	character varying(80)		Any comments for the trip.

Indexes:
"nx_t_trip_comment_trip" btree (trip)

Foreign-key constraints:
"fk_t_trip_comment_t_trip" FOREIGN KEY (trip, version_no)
REFERENCES regional_ce.t_trip(trip, version_no)

5.3 Table 3: t_effort

Comment: This table stores effort details.

Column	Type	Null?	Description
trip	integer	No	Trip identification number for regional purse-seine and CCAMLR BLL trips.
station_no	integer	No	Identification number for a set - distinct within a trip.
grp_no	smallint		Associates each South Pacific Regional Purse-seine station with a fishing permit or licence number stored in t_permits.
fish_year	character varying(7)		Fishing year for the period 01 October to 30 September displayed as year/yr e.g. 2004/05.
cal_year	integer		Calendar year.
datetime_s	timestamp without time zone		Set start date and time, when first part of longline cast from vessel.
datetime_e	timestamp without time zone		Set end date and time when last part of longline cast from vessel.
time_zone	character varying(12)		The time difference ahead or behind Greenwich Mean Time (GMT) (hrs).
activity	smallint		PS effort activity code.
duration	numeric(8,4)		Fishing duration (hours).
school_assn	smallint		PS school association code.
aerial_search	character varying(1)		PS aerial search: Y=Yes, N=No.
well_numbers	character varying(40)		PS well numbers catch put into.
target_sp	character varying(3)		Target species.
latitude_s	numeric(8,6)		Latitude at start of set in decimal degrees.
longitude_s	numeric(9,6)		Longitude at start of set in decimal degrees (east of Greenwich).
latitude_e	numeric(8,6)		Latitude at end of set in decimal degrees.

longitude_e	numeric(9,6)	Longitude at end of set in decimal degrees (east of Greenwich).
subarea	character varying(8)	CCAMLR area codes.
depth_bottom_s	integer	Bottom depth at start of set.
depth_bottom_e	integer	Bottom depth at end of set.
depth_fishing_s	integer	Fishing depth at start.
depth_fishing_e	integer	Fishing depth at end.
bottom_to_line	integer	Distance from the bottom to fishing line (m).
set_direction	integer	Direction in which the fishing gear is set (degrees).
set_direction_e	integer	Direction in which the fishing gear is set at end of set (degrees), where provided separately from start direction.
longline_type	character varying(6)	Type of longline configuration - CCAMLR code, see Appendix 2 of the database documentation.
line_type	character varying(1)	Type of material used in main line - CCAMLR code, see Appendix 2 of the database documentation.
bait_type	character varying(20)	Type of bait used - CCAMLR code.
hooks_baited	smallint	Percentage of hooks set with bait.
hook_size	character varying(8)	Mean width of the gap of hooks used (mm).
hook_make	character varying(8)	Brand name and type of hooks used - CCAMLR code.
hook_spacing	integer	Spacing between hooks on line (cm).
line_length	integer	Length of the main line set (m).
num_hooks	integer	Total number of hooks used for the set.
haul_datetime_s	timestamp without time zone	Date and time when the first part of the longline is retrieved.

haul_datetime_e	timestamp without time zone		Date and time when the last part of the longline is retrieved.
method	character varying(3)		Fishing method code.
fishing_type	character varying(3)		Main activity during each set: Commercial, Research.
line_weighting	character varying(4)		Description of the line weighting used - CCAMLR code.
affect_code	character varying(4)		Factors influencing fishing strategy - CCAMLR code.
bottom_type	character varying(120)		Description of the bottom type.
comments	character varying(120)		Comments as required.
form_type	character varying(3)		Form type: Purse-seine = PUR or KIR, CCAMLR = xls.
version_no	smallint	No	Version number of an event.
memo	character varying(12)		Memo codes that document any changes to the record, refer t_memo_codes.
startp	geometry		Position of vessel at start of the station as gis point type.
endp	geometry		Position of vessel at end of the station as gis point type.
track	geometry		Track line of vessel from start position to end position of station as gis line type.

```
Indexes:
   "pk_t_effort" PRIMARY KEY, btree (trip, station_no)
   "nx_t_effort_endp" gist (endp)
   "nx_t_effort_startp" gist (startp)
   "nx_t_effort_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_effort_latitude_e_check" CHECK (latitude_e >= (-80)::numeric
    AND latitude_e <= (-30)::numeric)
```

```
    "t_effort_longitude_e_check" CHECK
     (longitude_e >= 30.000000 AND longitude_e <= 330.000000)
    "t_effort_longitude_s_check" CHECK
     (longitude_s >= 30.000000 AND longitude_s <= 330.000000)

Foreign-key constraints:
    "fk_t_effort_t_trip" FOREIGN KEY (trip, version_no)
     REFERENCES regional_ce.t_trip(trip, version_no)
```

5.4 Table 4: t_catch

Comment: This table stores the catch recorded for each set of a South Pacific Regional Purse-seine set or Bottom Longline set.

Column	Type	Null?	Description
trip	integer	No	Trip identification number.
station_no	integer	No	Identification number for a set - distinct within a trip.
species	character(3)	No	Species code identifying the species caught.
status	character varying(1)	No	Code to identify the destination or status of the catch. E.g. R=Retained, D=Discarded. Refer t_status_codes. For method PS records some tuna discard codes but these are inconsistent between vessels.
grp_no	smallint		Identifies a fishing permit or licence number stored in t_permits for the catch for South Pacific Regional Purse-seine data set.
weight	numeric(10,2)		Weight of the fish caught, kg.
number_caught	integer		The number of fish caught.
av_weight	numeric(5,2)		The average weight of fish in the catch.
sp_ccamlr	character varying(3)		Species code as recorded in the CCAMLR data set.
version_no	smallint	No	Version number of an event.
memo	character varying(12)	No	Memo codes that document any changes to the record, refer t_memo_codes.
comments	character varying(100)		

Indexes:
"pk_t_catch" PRIMARY KEY, btree
(trip, station_no, species, status, memo)

Foreign-key constraints:
"fk_t_catch_reference_t_effort" FOREIGN KEY (trip, station_no)
REFERENCES regional_ce.t_effort(trip, station_no)
"fk_t_catch_reference_t_status" FOREIGN KEY (status)
REFERENCES regional_ce.t_status_codes(status)

5.5 Table 5: t_permits

Comment: This table stores each fishing permit or licence number issued for a South Pacific Regional Purse-seine trip.

Column	Type	Null?	Description
trip	integer	No	Trip identification number for regional purse-seine trips.
grp_no	integer	No	Associates each record in the t_effort table for a South Pacific Regional Purse-seine trip with a fishing permit or licence number stored in this table.
permit_no	character varying(100)		The fishing permit or licence number issued to South Pacific Regional Purse-seine vessels.
version_no	smallint	No	Version number of an event - currently all ver. 1.
memo	character varying(12)		Memo codes that document any changes to the record, refer t_memo_codes.

Indexes:
"pk_t_permits" PRIMARY KEY, btree (trip, grp_no)

5.6 Table 6: t_landings

Comment: Table containing landing information for the trip.

Column	Type	Null?	Description
land_key	integer	No	Primary key for the identification of landings.
trip	integer	No	Trip identification number for regional purse-seine trips and CCAMLR BLL trips.
date_land_start	date		First day of unloading / transhipping.
date_land_end	date		Last day of unloading / transhipping.
species	character(3)	No	Species code of the species landed.
state	character varying(3)		Code for the processed state of fish.
green_wt	integer		Weight of landed fish as whole weight (kgs).
wt_units	character varying(2)		Landing weight units, only recorded on Purse-seine Kiribati waters form: MT or ST (all weights stored as kgs).
rec_call_sign	character varying(8)		Carrier Radio Call sign.
cannery	character varying(30)		Name of Cannery or Carrier fish unloaded to.
fish_destination	character varying(30)		Destination of unloaded fish.
dated	date		Date recorded against landing signature.
version_no	smallint	No	Version number of an event.
memo	character varying(12)		Memo codes that document any changes to the record, refer t_memo_codes.

Indexes:
"pk_t_landings" PRIMARY KEY, btree (land_key)

Foreign-key constraints:
"fk_t_landings_t_trip" FOREIGN KEY (trip, version_no)
REFERENCES regional_ce.t_trip(trip, version_no)

5.7 Table 7: t_memo_codes

Comment: Table containing description for memo codes that document any changes to record.

Column	Type	Null?	Description
memo	character (3)	No	Memo code used in memo fields.
description	character varying(80)		Description for a change made or to tag a record.

Indexes:
"pk_t_memo_codes" PRIMARY KEY, btree (memo)

5.8 Table 8: t_status_codes

Comment: Table containing codes for the catch status (or destination) and descriptions.

Column	Type	Null?	Description
status	character varying(1)	No	Code for catch status; CCAMLR BLL 1 char and Purse-seine are numeric tuna discard codes.
description	character varying(40)		A description for each status code.

Indexes:
"pk_t_status_codes" PRIMARY KEY, btree (status)

6 Regional_ce business rules

6.1 Introduction to business rules

The following are a list of business rules applying to the **regional_ce** database. A business rule is a written statement specifying what the information system (i.e., any system that is designed to handle regional catch effort 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. Hence in a small number of cases values may legitimately be outside the range of business rules containing the word 'should'.

6.2 Summary of rules

Observer trip record (t_trip)

trip_no Must be a unique integer.

vessel_key Must be a valid vessel key as allocated for the vessel by the Ministry of Fisheries.

version_no Must be an integer greater than zero.

start_datetime The start date of the trip must be a valid date.

end_datetime The end date of the trip must be a valid date.

Multiple column checks on date:
The start date must not be later than the finish date. The dates should be within a period of 180 days of each other.

land_date The land date of the trip must be a valid date.

fads Must be an integer greater than zero.

tender Must be Y or N.

Trip comment record (t_trip_comment)

trip_no Must be equal to a trip number as listed in the *t_trip* table.

version_no Must be an integer greater than zero.

Permit record (t_permits)

trip_no Must be equal to a trip number as listed in the *t_trip* table.

version_no Must be an integer greater than zero.

grp_no Must be an integer greater than zero.

Station record (t_effort)

trip_no Must be equal to a trip number held in the *t_trip* table.

station_no Each station number must be a unique integer within all station records, for a given trip number.

version_no Must be an integer greater than zero.

datetime_s The start date of the station must be a legitimate date and the start time must be a valid 24-hour time of between 0000 - 2359.

Multiple column checks on station start date, trip start date and trip finish date:
The station start date must fall within the range of the trip start and finish dates. The station start date should be sequential between stations, for a given trip.

target_sp Must be a valid species code as listed in the *curr_spp* table in the **rdb** database, or a CCAMLR Target Species Codes as listed in Appendix 2

latitude_s Must be a valid latitude and degrees should fall within the range of 0 - 80.

longitude_s Must be a valid longitude.

depth_bottom_ s Bottom depth at start, should fall within the range of 10 – 2000 meters.

datetime_e The end date of the set must be a legitimate date and the end time must be a valid 24-hour time between 0000 - 2359.

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

Multiple column checks on station start date/time and station finish date/time:
The station finish date/time must not be before the station start date/time.

latitude_e Latitude degree at end, must be a valid latitude and degrees should fall within the range of 0 – 80.

longitude_e Longitude end must be a valid longitude in the range 0 –359 degrees.

Multiple column checks on station start and finish positions:
The start and finish positions should be within a defined maximum distance. The validation parameter for the distance between positions is set at 25 nautical miles. The distance between stations should be within a distance that could be covered by the vessel in the elapsed time period between stations.

depth_bottom_ e Bottom depth at finish, should fall within the range of 10 – 2000 meters.

method Fishing method must be a valid code, as listed in Appendix 2.

depth_fishing_s Depth at start, should fall within the reasonable range of 10 – 2000 meters.

depth_fishing_e Depth at end, should fall within the reasonable range of 10 – 2000 meters.

hooks_baited Percent baited must be a value within the range 0 - 100.

hook_spacing Must be an integer greater than zero.

line_length Must be an integer greater than zero.

num_hooks The number of hooks should fall within the range of 10 – 15000.

form_type Must be a valid form type code as listed in Appendix 2.

Regional_ce catch record (t_catch)

trip_no Must be equal to a trip number as listed in the *t_trip* table.

Multiple column checks on trip and station number:
The combination of trip and station_no must exist in the t_*effort* table.

version_no Must be an integer greater than zero.

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

discard Must be a valid code, indicating discard status, as listed in Appendix 2.

weight Must be a number greater than zero.

number_caught Must be a number greater than zero.

average_weight Must be a number greater than zero.

Regional_ce landings record (t_landings)

trip_no Must be equal to a trip number as listed in the *t_trip* table.

version_no Must be an integer greater than zero.

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

green_wt Must be a number greater than zero.

dated Must be a valid date.

7 Acknowledements

The author wishes to thank David Fisher for his technical assistance and review of this database documentation. Funding for developing this database and loading existing data was provided by the Ministry of Fisheries project DAT2004-01U, Upgrade ID – L9.

Appendix 1 - Data entry, error checking, and loading

The data in ***regional_ce*** have come from companies with New Zealand vessels fishing on the high seas and furnishing non-NZ Catch Effort forms. Data from CCAMLR trips are supplied in electronic form. These data from CCAMLR trips are not subject to the same level of checking by NIWA, as would be expected if NIWA was supplied with the raw data and was responsible for the data entry of these data. That is the data from CCAMLR trips do not pass through the data entry stages described here.

This section outlines the flow of paper-recorded data, namely for the South Pacific Regional Purse-seine fishery, and defines the separate tasks that are required to do this.

The South Pacific Regional Purse-seine data are supplied on forms as either hand written paper or typed format. Each trip is assigned a unique trip number and each set a sequential station number. The date and time will also be recorded as part of the station data.

1. **Pre-key entry, visual checking and batching:**
The data are then forwarded via the Ministry of Fisheries, to a project team member, who checks the above, and forwards the data to key entry.

2. **Key entry of data:**
At this point, trained data entry operators key in the data from the collated forms to a electronic fixed format ASCII file format on computer by keyboard entry. NIWA uses the KEYS Data Emulator for data entry.

All data entry is verified, that is, each page of data are keyed in twice and the two results are crosschecked for mismatches. Any data entry operator errors are corrected at this point.

The electronic data files are transferred for error checking along with the original raw data file. At this point the data are now ready for error checking and formatting routines.

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

The data are inserted into "working tables". This allows further checks of the integrity of the data, by taking advantage of relational databases ability to manipulate, match and compare related sets of data.

4. **"Groomed", validated data loaded to database. Available for analysis:**
The clean, groomed, and validated data are inserted into the appropriate database (in this case **regional_ce**) and now become available for extraction and analysis.

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

Appendix 2 – Reference codes

Fishing Method Codes

BLL	Bottom Longline
PS	Purse Seining

Activity codes
Activity codes recorded on the South Pacific Regional Purse-seine Logsheet.

1	Fishing Set
2	Searching
3	Transit
4	No fishing – breakdown
5	No fishing – bad weather
6	In port
7	Net cleaning set
10	Deploying or retrieving rafts, fads or payaos.
11	Catch transfer between wells at sea.

School association codes
School codes recorded on the South Pacific Regional Purse-seine Logsheet.

1	unassociated
2	Feeding on baitfish
3	Drifting log, debris or dead animal
4	Drifting rafts, fads or payaos.
5	Anchored rafts, fads or payaos.
6	Live whale
7	Live whale shark
8	Other

Status codes
Catch codes, 1 to 5 and 9 are all Purse-seine (PS) discard codes.

R	Retained
D	Discarded
L	Lost
T	Tagged and released
A	Alive Incidental catch
X	Dead Incidental catch
S	Skates released but not tagged
1	PS: undesirable species
2	PS: fish too small
3	PS: fish damaged
4	PS: vessel fully loaded
5	PS: other reason
9	PS: discard code not entered

Type of fishing
Main activity undertaken during each set– CCAMLR BLL

C	Commercial fishing
R	Research fishing

Type of longline
Type of longline configuration used during the period for which data are reported:– CCAMLR BLL

SP	Spanish,
AU	Autoliner,
O	Other

Type of line
Type of material used to construct the main line of the longline – CCAMLR BLL.

M	Monofilament
H	Hollow filament
T	multi filament
U	Unknown
O	Other

Description of line weighting used
A code to describe the line weighting used **–** CCAMLR BLL

IWL	Internally weighted longline;
UWW	Longline to which weights are manually attached;
STD	Longline with no weights manually attached;
COM	A combination of the any of the three other methods

Factors influencing fishing strategy
A code for describing any factors **strongly** affecting the fishing strategy**: –** CCAMLR BLL

ICE	Severe ice conditions;
CPT	Competition with other vessels;
CLO	An area closure due to a catch limit being reached;
NEW	Fishing activities in a new ground;
WEA	Adverse weather conditions; and
NON	No variables strongly affecting fishing activities.

Form_type codes

PUR	South Pacific Regional Purse-seine Logsheet
KIR	Purse-seine – Kiribati waters
xls	Data supplied electronically in a Microsoft excel spreadsheet.

CCAMLR Target Species Codes

TOA:	Antarctic toothfish (*Dissostichus mawsoni*)
TOP:	Patagonian toothfish (*Dissostichus eleginoides*)
TOT:	Both above (*Dissostichus spp)*