Database documentation: obs

B.M. Sanders & K. A. Mackay

NIWA Fisheries Data Management
Database Documentation Series

Revised March 2002

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 to the Scientific Observer Programme (SOP) database **obs**, and is a part of the database documentation series produced by NIWA. It supersedes the previous documentation by Kevin Mackay (1995)[1] 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 the relationships between the tables in **obs**, and the relationships between these tables and other databases.

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

NOTE:

For reasons such as naming conventions; tables not prefixed as t_, lengthy names (more than 12 characters), the obs database does not meet all the Marine Research standards (Ng 1992). This document provides a guide to users of the database in its current form only.

Access to this database and data are 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 Scientific Observer Programme Database

The **obs** database is one of several databases dedicated to information collected by the Scientific Observer Programme (SOP). The **obs** database, contains the catch and effort information for observed commercial trawl vessels. The second is the **obs_lfs,** which contains length frequency and biological data for commercial species as measured by the observers, as well as relevant trip and tow information.

The SOP was created in 1986 to send observers, contracted to the then MAF Fisheries, to monitor the catches of commercial trawlers. Although observer's duties at sea are many, this database deals exclusively with data recorded by observers in their Observer Trawl Catch Effort Logbook.

[1] Mackay, K.A. 1995. Marine Research database documentation. 12. obs. *MAF Fisheries Greta Point Internal Report No. 237*, 26 p.

Observers on each vessel are responsible for completing this logbook. Each logbook documents details for every trawl shot by the vessel such as position, time, total catch; the composition and weight of each catch; and the details of all fish processing carried out on board the vessel. In 1990, the format of the logbooks changed slightly.

Logbooks prior to trip number 1023 (July 1997) were processed by data entry operators at Greta Point. All data were then passed through a validation process before being loaded on to the **obs** database. Since then, the logbooks have been processed by the Ministry of Fisheries and entered into tables in their catch effort system database. Logbook data was then transferred to the **obs** database by MFish, up until 2001. Currently, the logbook data is downloaded from the MFish 'MOBY' server, by the database administrator from NIWA at the Greta Point site.

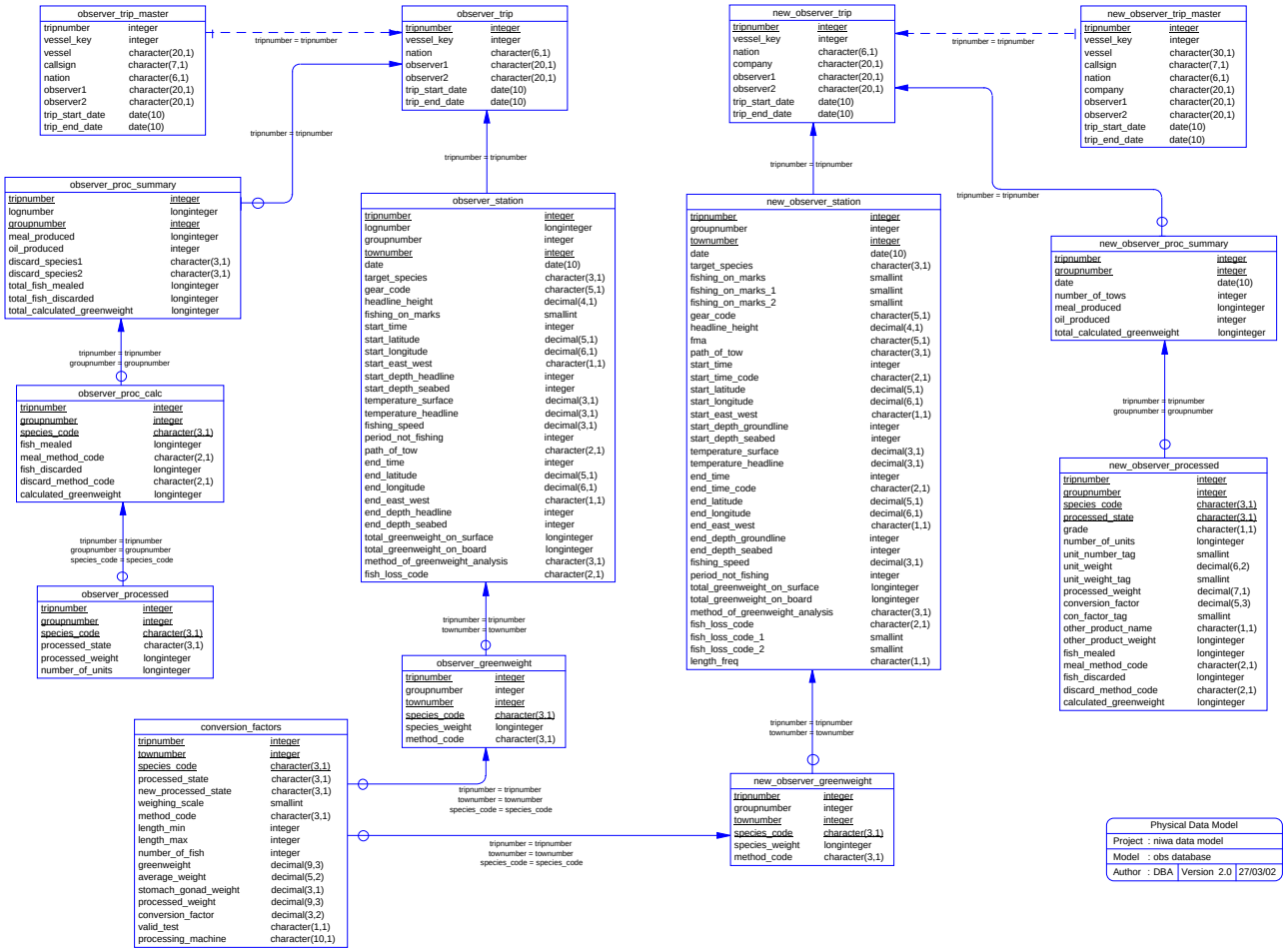

Figure 1: Entity Relationship Diagram (ERD) of the **obs** database.

3 Data Structures

3.1 Table relationships

The **obs** database comprises various related tables. The ERD (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 the tables' 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.

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.

Most of the tables in the **obs** database have some attributes, called foreign keys[3], which contain standard fisheries codes, such as *species*. These attributes provide links to the **rdb** (research database) database, which contains the definitive list of standard codes.

Section 5 shows a listing of all the **obs** tables as implemented by the Empress DBMS. The primary key has a unique index attached to it, that 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 new trip with an existing trip number.

The **obs** database is implemented as a relational database. That is, tables
are linked to one another by relationships. The **obs** database has two fundamental relationships that are repeated throughout the database:

1. The one-to-many relationship[4] This is shown in the ERD by connecting a single line to the parent (indicating 'many') from the child table. For example, consider the relationship between *observer_trip* and *observer_station*. This means that any one record in

[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 in a table (the parent) relates to one or more records in another table (the child).

observer_trip relates to at least one, but can be many records in *observer_station*, and one record in *observer_station* must relate to only one record in *observer_trip*.

2. The optional relationship. This is a special type of relationship, denoted by the symbol ‘o’ by the child table, at one end of the connecting line, which means that this relationship does not have to occur in every case. For example: one station in *observer_station* does not have to have any conversions factors taken from it. But if it does, it can have many. Conversely a conversion factor record, in *conversion_factors*, must relate to a station record in *observer_station*.

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 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.

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

Indices: NORMAL (2, 15) ON (attribute\{, attribute\)

Note that indices may be simple pointing to one attribute or composite pointing to more than one attribute. The numbers, `...(2, 15)...', are Empress default values relating to the amount of space allocated for duplicate entries.

3.2 Database Design

Initially, the tables structures were based on the original Observer logbooks. However, when the logbooks were altered slightly in 1990, completely new tables were created to reflect the new logbooks instead of altering the original data structure. This means that there are two sets of tables for the same data. Those tables prefixed by *observer* are for data from 1986 to April 1990, and those tables prefixed by *new_observer* are for data from May 1990 to present.

The basis for this database is an observer trip. Details for each observer trip are held in the tables *observer_trip* and *new_observer_trip* (Tables 1 and 2). Each trip is uniquely identified by a trip number, stored as the attribute tripnumber. Other details include the vessels name, call sign, nationality, the observer(s) names, and the trip start and finish dates. Note that the table *new_observer_trip* differs from the original table with the additional attribute company.

Observers record all the information pertaining to a trip in their Observer Logbook. Two sets of information are recorded: details about individual stations and catches; and details about fish processing groups. These two sets are reflected in the ERD (Figure 1).

Station and catch details.

Each observer trip has many stations, usually trawl tows. It is the observer's responsibility to record the details for all stations in the Observer Logbook. These details are stored in the tables *observer_station* and *new_observer_station* (Tables 3 and 4). Details for the station such as start and finish locations, time, depth, tow path, time spent not fishing, and estimated greenweight of the whole catch are recorded in this table. Many of these table's attributes are stored as codes.

Code attributes in **obs** are not defined or referenced in this or any database. See the Observer Trawl Catch Effort Logbook instructions for details.

These tables also contain a key attribute ‘groupnumber’ which provides a link to factory processing tables. This link is explained in detail later on in this document.

Note that table *new_observer_station* differs from the original table in that the depth of the groundline at the start and finish of the tows is recorded, instead of the headline depth. Also there is an additional attribute ‘length_freq’.

For each tow landed on the vessel, greenweights for each species are estimated. These estimates are recorded in the identical tables *observer_greenweight* and *new_observer_greenweight* (Tables 5 and 6), which records the trip and station number, the group number, species, estimated greenweight, and codes describing how the greenweight was estimated.

Fish processing group details.

Rather than processing fish constantly, fishing vessels usually process fish in groups (or batches). Most commonly, fish caught from one tow are processed as one group. So one tow can directly relate to one process group. However, if a tow catches a large amount of fish, more than the vessel's processing capacity, then the catch has to be split into smaller groups for processing. So one tow can relate to many process groups. Conversely, a tow can produce a catch of fish so small that it is not worth getting the factory running. The fish is held until more fish arrive from later tows before being processed. So many tows can relate to one process group. Finding exact combinations of station number and group number is not always possible as fish are well mixed up the the holding bins. As can be seen from the ERD, the only link between stations and group is through trip number.

It is at this point that the original and later tables differ significantly.

Summary details for individual fish processing groups are stored as records in the tables *observer_proc_summary* and *new_observer_proc_summary* (Tables 7 and 8). In both tables, each record contains information on the date of processing, the total amount of fish meal and oil

produced, and the estimated greenweight of fish processed. However, the *observer_proc_summary* table also records details about the species and weight of fish discarded.

The original version also contained an additional fish processing table called *observer_proc_calc* (Table 9). This table held summary data of each species recorded in *observer_processed*. In the later version, this data is incorporated into the table *new_observer_processed*.

The processing of fish transforms a whole fish to a "processed state", e.g., fillets, head-and-gutted, frozen whole, mealed, etc. Records for a species' various processed states within a process group are held in *observer_processed* and *new_observer_processed* (Tables 10 and 11).

Each record of *new_observer_processed* also holds additional data on different aspects for each species. For instance, it may contain details for a processed state, such as product grade, number of units and unit weight. Or it may contain details about how much of that species was mealed or discarded. There are codes which describe how the meal or discard weights were calculated.

The last table in the **obs** database is *conversion_factors* (Table 12). The records of this table contain details of conversion factors collected by the observers. Information contained in this table includes: the state the fish started out, e.g., whole; the final state of the fish, e.g., head-and-gutted or filleted; minimum and maximum lengths of the fish; number of fish; greenweight; processed weight; the calculated conversion factor; the name of the processing machine; and a flag to record whether or not the test is valid.

4 Table Summaries

This database has twelve main tables pertaining to trips observed by the SOP. There are also two views of the trip tables are detailed in section 5.

The following is a summary list of the main tables contained in the obs database:

1. *observer_trip* : contains details of trips covered by observers as part of the SOP from 1986 to April 1990.

2. *new_observer_trip* : contains details of trips covered by observers as part of the SOP since May 1990.

3. *observer_station* : contains details of stations (tows) made by an observer on a trip, as taken from the Observer Catch and Effort Logbook from 1986 to April 1990.

4. *new_observer_station* : contains details of stations (tows) made by an observer on a trip, as taken from the Observer Catch and Effort Logbook since May 1990.

5. *observer_greenweight* : contains details of greenweights for a species by station and trip from 1986 to April 1990.

6. *new_observer_greenweight* : contains details of greenweights for a species by station and trip since May 1990.

7. *observer_proc_summary* : contains summary data for all processed fish products for a species by process group, i.e., a summary of the records held in *observer_processed*, from 1986 to April 1990.

8. *new_observer_proc_summary* : contains summary data for all processed fish products for a species by process group, i.e., a summary of the records held in *new_observer_processed*, since May 1990.

9. *observer_proc_calc* : contains summary data for each species in *observer_processed* (only up to April 1990).

10. *observer_processed* : contains details of processed fish products by species, as recorded in the catch and effort logbook from 1986 to April 1990.

11. *new_observer_processed* : contains details of processed fish products by species, as recorded in the catch and effort logbook since May 1990.

12. *conversion_factors* : contains details of conversion factor data collected by the SOP.

5 obs Tables

The following are listings of the tables in the **obs** database.

Comments in this document are NOT all included in the database, but have been added to this document for the purposes of explanation.

The two tables previously storing trip details have been renamed with the suffice "_master" and two views of these tables created, using the original table names. This was implemented during 2001, to replace the attributes 'vessel' and 'callsign' previously available to end users of the trip tables, with an attribute named 'vessel_key'. This key is an identification number assigned by the Ministry of Fisheries to commercial vessels, as in the catch effort system. Thus *observer_trip* is a view of the table *observer_trip_master* and *new_observer_trip* a view of the table *new_observer_trip_master*, as shown in the Entity Relationship Diagram (ERD) in Figure 1. This restricts access of the vessel name and callsign to only the database administrator.

5.1 Table 1: observer_trip

Comment: Header information for observer trips up to April 1990 (i.e., the old format logbook). The 6 tables with the OBSERVER_ prefix contain the data for these trips.

Attributes	Data Type	Null?	Comment
tripnumber	integer	No	Trip identification number. A sequential number for each observed trip.
vessel_key	integer		The MFish id key assigned to this vessel.
nation	Character(6,1)		Nation of origin of the vessel. Can also be nation codes for charter companies.
observer1	Character(20,1)		Name of the first observer.
observer2	Character(20,1)		Name of the second observer.
trip_start_date	date(5)		Start date of the trip.
trip_end_date	date(5)		Finish date of the trip.

```
Creator:     dba
Indices:     UNIQUE BTREE ON (tripnumber)
               NORMAL (2, 15) TIMESERIES ON (trip_end_date)
               NORMAL (2, 15) TIMESERIES ON (trip_start_date)
               NORMAL (2, 15) BTREE ON (vessel)
               NORMAL (2, 15) BTREE ON (callsign)
```

5.2 Table 2: new_observer_trip

Comment: Details for trips since March 1990 (new format logbooks). All tables with the prefix NEW_OBSERVER_ refer to these trips.

Attributes	Data Type	Null?	Comment
tripnumber	integer		unique sequential identifier of trip
vessel_key	integer		Key assigned for vessel identification by MFish.
nation	character(6,1)		Nationality of the vessel
company	character(20,1)		Fishing company the vessel is fishing for
observer1	character(20,1)		Name of the first observer
observer2	character(20,1)		Name of the second observer
trip_start_date	date(5)		Start date of the trip
trip_end_date	date(5)		Finish date of the trip

```
  Creator:     dba
  Indices:     UNIQUE BTREE ON (tripnumber)
               NORMAL (2, 15) TIMESERIES ON (trip_end_date)
               NORMAL (2, 15) TIMESERIES ON (trip_start_date)
               NORMAL (2, 15) BTREE ON (vessel)
               NORMAL (2, 15) BTREE ON (callsign)
```

5.3 Table 3: observer_station

Comment: Station data from catch and effort logbooks. (See OBSERVER_TRIP)

Attributes	Data Type	Null?	Comment
tripnumber	integer		Unique sequential number for each trip
lognumber	longinteger		Unique number printed on the logbook
groupnumber	integer		sequential number for a group (by tow daily) of processed records
townumber	integer		sequential identifier for each tow
date	date(5)		date at start of tow
target_species	Character(3,1)		3 character code for the target species
gear_code	Character(5,1)		net identifier (BT = bottom trawl, MW = midwater)
headline_height	decimal(4,1)		vertical opening distance of net (m)
fishing_on_marks	smallint		see observer logbook instructions
start_time	integer		NZST (24 hour clock)
start_latitude	decimal(5,1)		Latitude for the start of the tow, in decimal minutes (DDMM.m) format
start_longitude	decimal(6,1)		Longitude for the start of the tow, in decimal minutes (DDDMM.m) format
start_east_west	Character(1,1)		Code to denote whether the tow started east (=E) or west (=W) of 180 long.
start_depth_headline	integer		Depth to headline at the start of tow
start_depth_seabed	integer		Depth to seabed at the start of tow (m).

temperature_surface	decimal(3,1)	Sea surface temperature (decimal degrees C)
temperature_headline	decimal(3,1)	Sea temperature at the headline (decimal degrees C)
fishing_speed	decimal(3,1)	knots
period_not_fishing	integer	duration between start and end-time when net not fishing (hr and min)
path_of_tow	Character(2,1)	configuration of tow (see logbook instructions)
end_time	integer	NZST (24 hour clock)
end_latitude	decimal(5,1	Latitude for the end of the tow, decimal minutes (DDMM.m) format
end_longitude	decimal(6,1)	Longitude for the end of the tow, in decimal minutes (DDDMM.m) format
end_east_west	Character(1,1)	Code to denote whether the tow ended east (=E) or west (=W) of 180 long.
end_depth_headline	integer	Depth to headline at the end of tow
end_depth_seabed	integer	Depth to seabed at the end of tow
total_greenweight_on_surface	longinteger	weight of catch when net surfaces (kg)
total_greenweight_on_board	longinteger	weight of catch when net hauled aboard (kg). This will equal total_greenweight_on_surface unless fish are lost from the net.
method_of_greenweight_analysis	Character(3,1)	method used to determine total_greenweight_on_board (see logbook instructions)
fish_loss_code	Character(2,1)	description of type of fish loss (see logbook instructions)

Creator: dba

Indices: NORMAL (2, 15) ON (tripnumber)
NORMAL (2, 15) ON (groupnumber)
NORMAL (2, 15) ON (townumber)
NORMAL (2, 15) ON (date)
NORMAL (2, 15) ON (target_species)

5.4 Table 4: new_observer_station

Comment: Station data from the catch and effort logbook.
(See NEW_OBSERVER_TRIP.)

Attributes	Data Type	Null?	Comment
tripnumber	integer		Unique sequential number for each trip
groupnumber	integer		sequential number for a group (by tow or daily) of processed records
townumber	integer		sequential identifier for each tow
date	date(5)		date at start of tow
target_species	Character(3,1)		3 character code for the target species
fishing_on_marks	smallint		see observer logbook instructions
fishing_on_marks_1	smallint		What was previously the first digit of fishing_on_marks, the code indicates whether the vessel was actively targeting fish sign: 0=no; 1=yes
fishing_on_marks_2	smallint		What was previously the second digit of fishing_on_marks, the code indicates who shot the net (observers make up their own codes
gear_code	Character(5,1)		net identifier (BT = bottom trawl, MW = midwater
headline_height	decimal(4,1)		vertical opening distance of net (m)
fma	Character(5,1)		fisheries management area (see logbook instructions)
path_of_tow	Character(3,1)		configuration of tow (see logbook instructions)
start_time	integer		NZST (24 hour clock)
start_time_code	Character(2,1)		description of what start-time refers to (see logbook instructions)

start_latitude	decimal(5,1)	Latitude for the start of the tow, in decimal minutes (DDMM.m) format
start_longitude	decimal(6,1)	Longitude for the start of the tow, in decimal minutes (DDDMM.m) format
start_east_west	Character(1,1)	Code to denote whether the tow started east (=E) or west (=W) of 180 long.
start_depth_groundline	integer	(m) Depth to groundline at the start of tow
start_depth_seabed	integer	(m)Depth to seabed at the start of tow
temperature_surface	decimal(3,1)	Sea surface temperature (decimal degrees C)
temperature_headline	decimal(3,1)	Sea temperature at the headline (decimal degrees C)
end_time	integer	NZST (24 hour clock)
end_time_code	Character(2,1)	description of what end-time refers to (see logbook instructions)
end_latitude	decimal(5,1)	Latitude for the end of the tow, in decimal minutes (DDMM.m) format
end_longitude	decimal(6,1)	Longitude for the end of the tow, in decimal minutes (DDDMM.m) format
end_east_west	Character(1,1)	Code to denote whether the tow ended east (=E) or west (=W) of 180 long.
end_depth_groundline	integer	(m) Depth to groundline at the end of tow
end_depth_seabed	integer	(m) Depth to seabed at the end of tow
fishing_speed	decimal(3,1)	knots
period_not_fishing	integer	duration between start and end-time when net not fishing (hr and min)
total_greenweight_on_surface	longinteger	weight of catch when net surfaces (kg)

total_greenweight_on_board	longinteger	weight of catch when net hauled aboard(kg). This will equal total_greenweight_on_surface unless fish are lost from the net.
method_of_greenweight_analysis	Character(3,1)	method used to determine total_greenweight_on_board (see logbook instructions)
fish_loss_code	Character(2,1)	description of type of fish loss (see logbook instructions)
fish_loss_code_1	smallint	What was previously the first digit of the fish_loss_code, indicates fish loss below the sea surface
fish_loss_code_2	smallint	What was previously the second digit of the fish_loss_code, indicates fish loss at the sea surface or on the ramp
length_freq	Character(1,1)	Y=biol data collected from this tow.

Creator: dba

Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (groupnumber)
NORMAL (2, 15) BTREE ON (townumber)
NORMAL (2, 15) BTREE ON (date)
NORMAL (2, 15) BTREE ON (target_species)

5.5 Table 5: observer_greenweight

Comment: Catch data from the catch and effort logbook.
(See OBSERVER_TRIP.)

Attributes	Data Type	Null?	Comment
tripnumber	integer		Unique sequential number for each trip
groupnumber	integer		sequential number for a group (by tow or daily) of processed records
townumber	integer		sequential identifier for each tow
species_code	Character(3,1)		3-char code for a species of fish caught
species_weight	longinteger		greenweight of species (kg)
method_code	Character(3,1)		method used to establish greenweight (see logbook instructions)

Comment: Catch data from the catch and effort logbook. (See OBSERVER_TRIP.)

Creator: dba
Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (groupnumber)
NORMAL (2, 15) BTREE ON (townumber)
NORMAL (2, 15) BTREE ON (species_code)

5.6 Table 6: new_observer_greenweight

Comment: Catch data from the Catch and Effort Logbook. (See NEW_OBSERVER_TRIP.)

Attributes	Data Type	Null?	Comment
tripnumber	integer		Unique sequential number for each trip
groupnumber	integer		sequential number for a group (by tow or daily) of processed records
townumber	integer		sequential identifier for each tow
species_code	Character(3,1)		3-char code for a species of fish caught
species_weight	longinteger		greenweight of species (kg)
method_code	Character(3,1)		method used to establish greenweight (see logbook instructions)

Creator: dba
Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (groupnumber)
NORMAL (2, 15) BTREE ON (townumber)
NORMAL (2, 15) BTREE ON (species_code)

5.7 Table 7: observer_proc_summary

Comment: Summary data (all species combined) for product recorded in OBSERVER_PROCESSED and OBSERVER_PROC_CALC.(See OBSERVER_TRIP.)

Attributes	Data Type	Null?	Comment
tripnumber	integer	No	Unique sequential number for each trip
lognumber	longinteger		Unique number printed on the logbook
groupnumber	integer		Sequential number for a group (by tow or daily of processed fish records
meal_produced	longinteger		kgs of meal produced
oil_produced	integer		litres of oil produced
discard_species1	Character(3,1)		species code of discarded species
discard_species2	Character(3,1)		species code of discarded species
total_fish_mealed	longinteger		greenweight of fish mealed (kg)
total_fish_discarded	longinteger		greenweight of fish discarded (kg)
total_calculated_greenweight	longinteger		sum of calculated greenweights (kg)

Creator: dba
Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (groupnumber)

5.8 Table 8: new_observer_proc_summary

Comment: Summary data for records in NEW_OBSERVER_PROCESSED. For each groupnumber, one record in NEW_OBSERVER_PROC_SUMMARY corresponds to several records in NEW_OBSERVER_PROCESSED.(See NEW_OBSERVER_TRIP.)

Attributes	**Data Type**	**Null?**	**Comment**
tripnumber	integer		Unique sequential number for ch trip
groupnumber	integer		Sequential number for a group (by tow or daily of processed fish records
date	date(5)		Date on which processing took place
number_of_tows	integer		number of tows comprising processed catch record
meal_produced	longinteger		kgs of meal produced
oil_produced	integer		litres of oil produced
total_calculated_greenweight	longinteger		sum of calculated_greenweights (kg)

Creator: dba
Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (groupnumber)
NORMAL (2, 15) BTREE ON (date)

5.9 Table 9: observer_proc_calc

Comment: Summary data for each species in OBSERVER_PROCESSED. (See OBSERVER_TRIP.)

Attributes	Data Type	Null?	Comment
tripnumber	integer	No	Unique sequential number for each trip
groupnumber	integer		Sequential number for a group (by tow or daily of processed fish records
species_code	Character(3,1)		3-char code for a species of fish caught
fish_mealed	longinteger		mealed greenweight (kg)
meal_method_code	Character(2,1)		method of analysis of fish mealed (see logbook instructions
fish_discarded	longinteger		discarded greenweight (kg)
discard_method_code	Character(2,1)		method of analysis of fish discarded (see logbook instructions)
calculated_greenweight	longinteger		number_of_units x unit_weight x conversion_factor (kg)

Creator: dba
Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (groupnumber)
NORMAL (2, 15) BTREE ON (species_code)

5.10 Table 10: observer_processed

Comment: Number of trays or weight of product from the catch and effort logbook. The calculated weights for each species are contained in OBSERVER_PROC_CALC.

Attributes	Data Type	Null?	Comment
tripnumber	integer	No	Unique sequential number for each trip
groupnumber	integer		Sequential number for a group (by tow or daily of processed fish records
species_code	Character(3,1)		3-char code for a species of fish caught
processed_state	Character(3,1)		3-character code for the state to which the fish has been processed to
processed_weight	longinteger		only used for a few trips (kg)
number_of_units	longinteger		number of cartons/trays/bags produced for that species, state and grade.

Creator: dba
Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (species_code)
NORMAL (2, 15) BTREE ON (groupnumber)

5.11 Table 11: new_observer_processed

Comment: Processed fish recorded in the catch and effort logbook. The attribute GROUPNUMBER links this table to NEW_OBSERVER_STATION or NEW_OBSERVER_PROC_SUMMARY. (See NEW_OBSERVER_TRIP.)

Attributes	Data Type	Null?	Comment
tripnumber	integer	No	Unique sequential number for each trip
groupnumber	integer		Sequential number for a group (by tow or daily of processed fish records
species_code	Character(3,1)		3-char code for a species of fish caught
processed_state	Character(3,1)		3-character code for the state to which the fish has been processed to
grade	Character(1,1)		grade code of product
number_of_units	longinteger		number of cartons/trays/bags produced for that species, state and grade
unit_number_tag	smallint		
unit_weight	decimal(6,2)		(kg)
unit_weight_tag	smallint		
processed_weight	decimal(7,1)		number_of_units x unit_weight (kg)
conversion_factor	decimal(5,3)		Conversion factor applied to processed product to get weight of fish processed
con_factor_tag	smallint		code for which conversion factor used (see logbook instructions)
other_product_name	Character(1,1)		code for other products (see logbook instructions)
other_product_weight	longinteger		kg
fish_mealed	longinteger		mealed greenweight (kg)

meal_method_code	Character(2,1)	method of analysis of fish mealed (see logbook instructions)
fish_discarded	longinteger	discarded greenweight (kg)
discard_method_code	Character(2,1)	method of analysis of fish discarded (see logbook instructions)
calculated_greenweight	longinteger	number_of_units x unit_weight x conversion_factor (kg)

Creator: dba
Indices: NORMAL (2, 15) BTREE ON (tripnumber)
NORMAL (2, 15) BTREE ON (groupnumber)
NORMAL (2, 15) BTREE ON (species_code)

5.12 Table 12: conversion_factors

Comment: Scientific Observer Programme conversion factor data. All lengths are in cm and weights in kg.

Attributes	Data Type	Null?	Comment
tripnumber	integer		Unique sequential number for each trip
townumber	integer		Sequential number for each tow
species_code	Character(3,1)		3-char code for a species of fish caught
processed_state	Character(3,1)		valid states = HGU, DRE, TRU, FIL.
new_processed_state	Character(3,1)		valid states = HGU, DRE, FIL, SKF.
weighing_scale	smallint		1 = electronic scales, 2 = flatbed, 3 = salter.
method_code	Character(3,1)		3-character code to define the method(s) used to determine weight (see logbook) instructions
length_min	integer		minimum length of fish in sample (cm)
length_max	integer		maximum length of fish in sample (cm)
number_of_fish	integer		number of fish in sample
greenweight	decimal(9,3)		Greenweight of the fish used to calculate the conversion factor
average_weight	decimal(5,2)		average weight of fish in sample (kg)
stomach_gonad_weight	decimal(3,1)		weight of stomach and gonads if significant (kg)
processed_weight	decimal(9,3)		Weight (kg) of the fish after processing
conversion_factor	decimal(3,2)		greenweight/processed_weight
valid_test	Character(1,1)		Y = yes, N = no

```
processing_machine            Character(10,1)  name of heading \& gutting or
                                                filleting machine used

  Creator:     dba
  Indices:     NORMAL (2, 15) BTREE ON (tripnumber)
               NORMAL (2, 15) BTREE ON (townumber)
               NORMAL (2, 15) BTREE ON (species_code)
               NORMAL (2, 15) BTREE ON (processed_state)
  Lock Level:  NONE
```

6 References

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

7 obs business rules

7.1 Introduction to business rules

The following are a list of business rules applying to the **obs** database. A business rule is a written statement specifying what the information system must do or how it must be structured. In this instance the information system is any system that is designed to handle observer length frequency sampling data.

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.

The rules out lined in this document have been mainly derived from rules on relevant tables in the **obs_lfs** database. No account is made of any validation and error checking made on logbook data during the process of it being entered into the catch effort system by the Ministry of Fisheries. This part of the data management is not visible to NIWA. (Also see data entry Appendix 2).

7.2 Summary of rules

Observer trip record (observer_trip and new_observer_trip)

tripnumber	Must be a unique integer.
vessel_key	Must be a valid vessel key of the vessel observed, as assigned by MFish.
trip_start_date	The start date of the trip must be a legitimate date within the specified period the data set covers.
trip_end_date	The finish date of the trip must be a legitimate date within the specified period the data set covers.
	Multiple column checks on date: The start date must not be later than the finish date. The dates should be within a period of six weeks of each other.

Observer station record (observer_station and new_observer_station)

tripnumber	Must be equal to a trip number held in the *trip_master* table.
townumber	Must be a unique integer within all station records, for a given trip number.
date	The date of the station must be a legitimate date. **Multiple column checks on station date, trip start date and trip finish date**: The station 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_species	Must be a valid species code as listed in the *curr_spp* table in the **rdb** database.
start_time	Station start time must be a valid 24-hour time of between 0000 - 2359.
headline_height	The headline height should fall within the reasonable range of 10 – 120 m.
fma	Must be one of the valid area codes for the New Zealand Exclusive Economic Zone (EEZ) as listed in Appendix 1 or "ET' area code for outside of the zone).
path_of_tow	Consists of three parts: tow type, tow configuration and number of turns. The tow type code and configuration must be a valid codes as listed in Appendix 1.
start_latitude	Must be a valid latitude and degrees should fall within the range of 33 - 48 South.
start_longitude	Must be a valid longitude and degrees should fall within the range of 164 East to 170 West.
start_east_west	Longitude East or West at start, must be either "E" or "W".
start_depth_groundline	Net depth at start, should fall within the reasonable range of 10 – 2000 m.
start_depth_seabed	Depth of seabed at start, should fall within the range of 10 – 2000 m.

temperature_surface	Sea surface temperature should be in the range 8.0 to 24.0 degrees Celsius.
temperature_headline	Sea bottom temperature should be in the range 4.0 to 15.5 degrees Celsius.
end_time	Station finish time must be a valid 24-hour time of between 0000 - 2359.
	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. The finish date/start must be before the start date/time of any subsequent stations.
end_latitude	Latitude degree at finish, must be a valid latitude and degrees should fall within the range of 33 - 48 South.
end_longitude	Longitude degree finish, must be a valid longitude and degrees should fall within the reasonable range of 164 East to 170 West.
end_east_west	Longitude East or West at finish, must be either "E" or "W".
	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 time elapsed between the start and the finish of the station is taken into account on validation. The distance between stations must be within a distance that could be covered by the vessel in the elapsed time period between stations. The validation parameter is set at 15 knots for this check. Note, for drop lines, the end of the line set is not required as it is equal to the start position.
end_depth_groundline	Net depth at finish, should fall within the reasonable range of 10 – 2000 meters.
end_depth_seabed	Bottom depth at finish, should fall within the range of 10 – 2000 m.
fishing_speed	Speed should fall within the reasonable range of 1.0 – 6.0 knots.

	Multiple column checks on trip and station number: The combination of tripnumber and townumber must exist in the *observer_ station* table.
species_code	Must be a valid species code as listed in the *curr_spp* table in the **rdb** database.
species_weight	Must be a number greater than zero.
method_code	Weight method code, must be a valid code combining two parts. Part 1: the location of the catch at the time of analysis. Part 2: an Alpha to indicate method used to analysis the total catch. eg., means 7K analysis in processing area (7) and weighted in full (K). This code must compile the codes listed in Appendix 1.

Appendix 1 - Reference Code Tables

The information listed in this Appendix is current at the time of writing, and as implemented at November 30, 2001.

Area codes

The valid area codes as in this list are a sub-set of the area codes in the **rdb** database.

AKE	East North Is. from North Cape to Bay of Plenty (FMA 1)
AKW	West North Is. from North Cp. to North Taranaki Bight (FMA 9)
CEE	East North Is. from south of Bay of Plenty to Wgtn (FMA 2)
CEW	West North Is. from South Taranaki Bight to Wgtn (FMA 8)
CHA	West Coast South Island to Fiordland incl. Kaikoura (FMA 7)
KER	Kermadec (FMA 10)
SEC	East Coast South Island from Pegasus Bay to Catlins (FMA 3)
SOE	Chatham Rise (FMA 4)
SOI	Southern Offshore Islands - Auckland & Campbell Is. (FMA 6A)
SOU	South Island from Foveaux Strait to Fiordland (FMA 5)
SUB	Subantarctic incl. Bounty Is and Pukaki Rise (FMA 6)

Tow type codes

1	Bottom throughout tow.
2	Midwater at relatively constant depth.
3	Midwater in a broad range of depths.
4	Mixed bottom & midwater.

Tow configuration codes

A	Straight line
B	”U”
C	Zigzag
D	Closed pattern (circle, loop etc)
E	Constant depth contour
F	Pinnacle fishing

Trawl catch weight method codes (for catch weight for trawl methods.)
Part1: The location of the catch at the time of analysis.

1 In or spilling from codend.
3 Loose on deck.
5 In holding bins.
7 On sorting conveyor or in processing area
9 Packing area.

Part 2: Method used to analysis the total catch.

A Extrapolated from other catches (retrospectively).
B Visual estimate
C Inexact count *x* estimated average weight
D Calculated by deduction (total minus other species)
E Measured dimensions of catch *x* density
F Calculated from percentage composition in a volume of fish
G Calculated from percentage composition in a sample over several tows
H Measuring fish and correlating length with weight
I Accurate count *x* average weight previous tows
J Accurate count *x* average weight in random sample this tows
K Weighed in full.
If pan/block counts used, the following codes are applicable.
L accurate full count by vessel x official conversion factor x nominal weight
M accurate full count by vessel x official conversion factor x observer weight
N accurate full count by vessel x observer list conversion factor x nominal weight
O accurate full count by vessel x observer list conversion factor x observer weight
P accurate full count by vessel x observer trip conversion factor x nominal weight
Q accurate full count by vessel x observer trip conversion factor x observer weight
R accurate full count by observer x official conversion factor x nominal weight
S accurate full count by observer x official conversion factor x observer weight
T accurate full count by observer x observer list conversion factor x nominal weight
U accurate full count by observer x observer list conversion factor x observer weight
V accurate full count by observer x observer trip conversion factor x nominal weight
W accurate full count by observer x observer trip conversion factor x observer weight

X Any other technique (should be defined in comments).

Appendix 2 - Data entry, error checking, and loading

Prior to July 1997 the observer log-books were processed in a similar way to other databases now administrated by NIWA, under the Data Management contract with the Ministry of Fisheries.

Trained data entry operators keyed in data from the log-books to an electronic fixed format ASCII file format. (NIWA uses the KEYS Data Emulator for data entry). Data were then verified, with each page of data keyed in twice and the two results crosschecked for mismatches. Any data entry operator errors were corrected at this point.

Data were checked and corrected in HOLD_NEW_ tables in a different database, using the files CHECK_NEW_STATION, _PROC, _SPECIES and the checkq program CHECKNEW. When the data had been checked and any corrections made, the file LOADUP dumped the data into files which could then be loaded into the main tables.

Post July 1997, the processing of log-books is now carried out by the Ministry of Fisheries. The first trip number this applies to is 1023, (some overlap occurs), being trips that started during July 1997. The log-book data since this time have been incorporated into the catch effort system, therefore validation and error checking as above is no longer applicable by NIWA personnel. The log-books are retained by the Ministry of Fisheries. Note, as loading of the ***obs_lfs*** database is dependent upon the loading of correct data into the **obs** database, this process includes checking of critical station data.

Initially Ministry of Fisheries staff carried out the transfer of data across to the **obs** database. Currently, data is downloaded by NIWA Nominated Personnel, from the views available on the Ministry of Fisheries catch effort system.