Database documentation: nonfish_bycatch

C. Sutton and F. Wei

NIWA Fisheries Data Management
Database Documentation Series

December 2015

Version Control

Version	Status	Changed By	Reason	Date
1.0	Release	C.Sutton, F.Wei	initial version	Oct 2003
1.1	Update	D Fisher	Noted finish of time series in Oct 2008.	Oct 2012
1.2	Update	D Fisher, F Wei	Postgres version	Dec 2015

Contents

List of Figures

1 Database documentation series

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

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

This document describes the **nonfish_bycatch** database, and is a 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 and business rules. The ERD graphically shows how all the tables fit in together.

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

Access to this database is restricted to 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 Nonfish bycatch data

Protected species such as seabirds and marine mammals are caught incidentally in New Zealand fisheries. In some cases information on populations of protected species, or the actual amount of bycatch occurring is insufficient to determine precise effects from fishing. For some protected species, caught in particular fisheries, a Maximum Allowable Level of Fishing-Related Mortality (MALFIRM) may be required, e.g., NZ Sea Lions in the Squid fishery. However, in other fisheries, e.g., NZ Fur Seals in the hoki fishery, fewer controls exist over the impact of fishing.

In an attempt to mitigate this problem of insufficient information, all vessel masters are obligated to complete a Non-Fish Incidental Catch Reporting Form following the capture of a protected species. The data from these forms are used to populate the **nonfish_bycatch** database. The completion of this form is a statutory requirement and is based on the Wildlife Act (1953) and the Marine Mammals Protection Act (1978).

As at the date of the first publication of this report in October 2003, these data cover a continuous time-series since December 1996. A new NPC form was introduced on 1 October 2008 and data ceased to be loaded to this database, with the last record captured from 20 October 2008. Subsequent data are held in the MFish / MPI Warehou database.

3 Data structures

3.1 Introduction

The Ministry of Fisheries designed the Non-Fish Incidental Catch Reporting Form in 1996 in consultation with the then New Zealand Fishing Industry Board. Since this time these data have been stored in a *Microsoft Access* database held by the Ministry of Fisheries. In 2003 the responsibility of this database was transferred to NIWA and ported into a Relational Database Management System (RDBMS). It is this current implementation of the design and structure of **nonfish_bycatch** that is described in this document.

3.2 Database description

The ERD for **nonfish_bycatch** (Figure 1) shows the logical structure of the database and its entities (each entity is implemented as a database *table*) and relationships between these. All of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary keys[1]. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed.

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

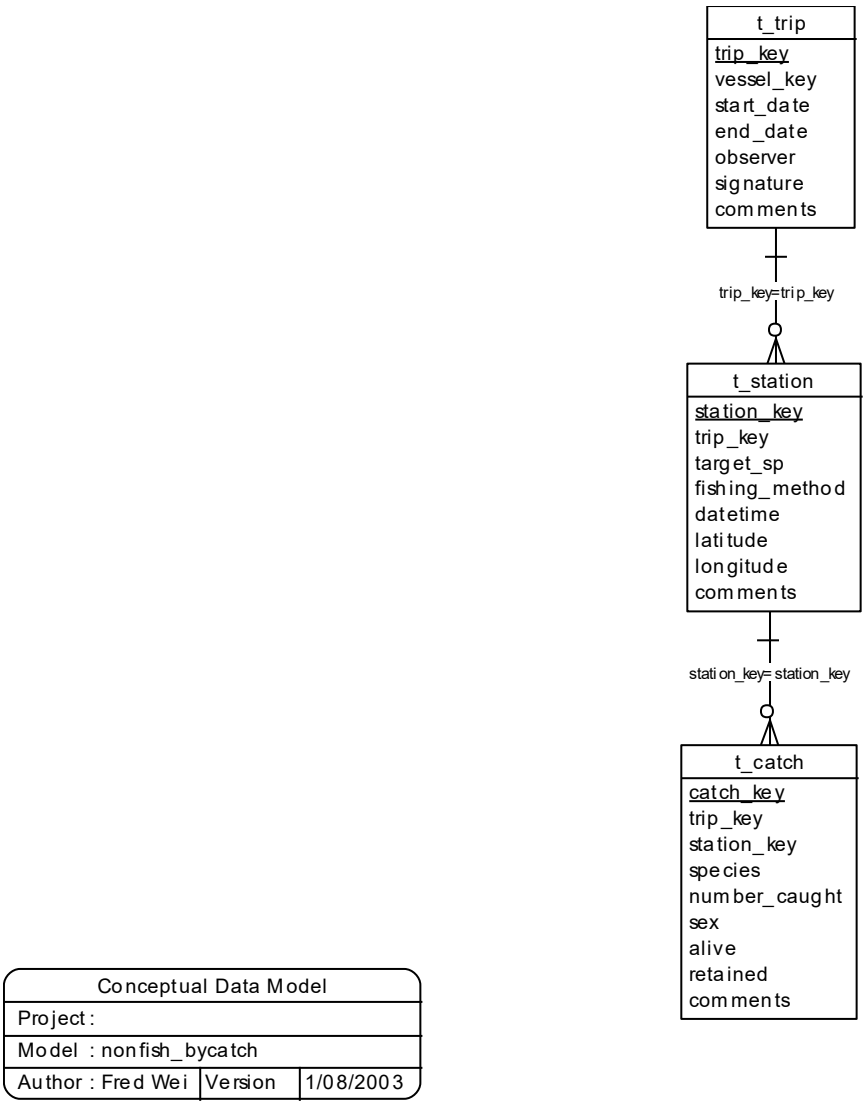

Figure 1: Entity Relationship Diagram (ERD) of the nonfish_bycatch database.

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. The tables in the **nonfish_bycatch** database also contain special attributes, called foreign keys[2]. Section 5 shows a listing of all the **nonfish_bycatch** 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*])

[2] 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.

This prevents records with duplicate key values from being inserted into the table; e.g., a trip with an existing trip key.

The **nonfish_bycatch** 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 **nonfish_bycatch** are of the type *one-to-many*. This is shown in the ERD by connecting a single line from the "parent" table; e.g., *t_trip*, to the "child" table; e.g., *t_station*, with crows foot (indicating 'many') pointing to the "child". For example, consider the relationship between the tables' *t_trip* (the "parent" table) and *t_station* (the "child" table). Any one trip in *t_trip* can have one or more associated stations in *t_station*, but any one station 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 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_station*, which has a mandatory 'one' and an optional 'many'. This means that one trip record can have zero or more (many) stations within it, but one station must have one, and only one, associated record in the trip table.

These relationships are enforced in the database by the use of foreign key constraints[3]. Foreign keys 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 **nonfish_bycatch** prevent these from occurring. Constraints are shown in the table listings by the following example:

Foreign-key constraints:
"t_station_trip_key_fkey" FOREIGN KEY (trip_key) REFERENCES t_trip(trip_key)

This means that the value of the attribute *trip_key* in the current record must already exist in the parent table *t_trip* or the record will be rejected and an error message will be displayed:

All tables in the 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:` index_name btree (*attribute*)

Details for each trip are held in the table *t_trip*. Each trip is uniquely identified by a trip code, stored as the attribute *trip_key*.

[3] Also known as integrity checks.

Similarly, details for each station are held in the table *t_station*. Each station is uniquely identified by a station code, stored as the attribute *station_key*.

Details for each catch are held in the table *t_catch*. Each catch is uniquely identified by a catch code, stored as the attribute *catch_key*.

4 Table summaries

The **nonfish_bycatch** database has three main tables.

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

- **t_trip :** contains profile information on all trips.
- **t_station :** contains details common to both trawl and longline sets, including date, position, and target fish species of the tow/set.
- **t_catch :** contains catch data, including identification of species caught, number of animals caught, sex and status (alive or dead).

5 Nonfish_bycatch tables

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

5.1 Table 1: t_trip

Comment: Contains header information for each trip event.

Column	Type	Null?	Description
trip_key	integer	No	Primary key using sequential trip key value.
vessel_key	integer		MFish identification number of the fishing vessel.
start_date	date		Start date for a trip.
end_date	date		End date for a trip.
observer	character varying(32)		Indicates if observer was present and type.
signature	character varying(1)		Indicates if form was signed: Y=Yes, N=No.
comments	character varying(80)		Comments added in loading process.

Indexes:
"t_trip_pkey" PRIMARY KEY, btree (trip_key)

5.2 Table 2: t_station

Comment: Contains details for stations with nonfish bycatch, including target_sp, fishing method, date and position.

Column	Type	Null?	Description
station_key	integer	No	Primary key used to identify a sequential number for each tow made during a trip.
trip_key	integer		Foreign key to reference t_trip
target_sp	character varying(3)		Target fish species code.
fishing_method	character varying(3)		Code for fishing method used.
datetime	timestamp without time zone		Date and time when incident occurred (24-hour format, NZDT).
latitude	numeric(8,6)		Decimalised latitude of vessel when animal was bought on board.
longitude	numeric(9,6)		Decimalised longitude of vessel when animal was bought on board (in degrees east of Greenwich).
comments	character varying(80)		Comments added in loading process.
position	geometry		Position of vessel as gis point type.

```
Indexes:
   "t_station_pkey" PRIMARY KEY, btree (station_key)
   "nx_t_station_position" gist ("position")
Check constraints:
   "enforce_dims_position" CHECK (ndims("position") = 2)
   "enforce_geotype_position" CHECK (geometrytype("position") = 'POINT'::text
                                     OR "position" IS NULL)
   "enforce_srid_position" CHECK (srid("position") = 4326)

Foreign-key constraints:
   "t_station_trip_key_fkey" FOREIGN KEY (trip_key)
    REFERENCES nonfish.t_trip(trip_key)
```

5.3 Table 3: t_catch

Comment: Contains each unit of catch recorded in Non-Fish Incidental Catch Reporting Form.

Column	Type	Null?	Description
catch_key	integer	No	Primary key generated from a counter.
trip_key	integer		Redundant key to reference t_trip, no foreign key constraint is applied.
station_key	integer		Foreign key to reference t_station.
species	character(3)		3 character species code. Refer rdb.curr_spp.
number_caught	integer		The number caught for this fishing event.
sex	character varying(1)		Sex code: 0=Unsexed, 1=Male, 2=Female, 3=Unknown (unable to determine).
alive	character varying(1)		Status of animal: A=Alive, D=Dead.
retained	character varying(1)		Records if species was retained on board: Y=Yes, N=No.
comments	character varying(40)		Comments added in loading process.

Indexes:
"t_catch_pkey" PRIMARY KEY, btree (catch_key)

Foreign-key constraints:
"t_catch_station_key_fkey" FOREIGN KEY (station_key)
REFERENCES nonfish.t_station(station_key)

6 Data Loading and Validation

A loading and validating shell script will be used to load and validate nonfish_bycatch data.

7 Nonfish_bycatch business rules

7.1 Introduction to business rules

The following are a list of business rules applying to the **nonfish_bycatch** database. A business rule is a written statement specifying what the information system 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.

Data model integrity is ensured by database built-in constraints. If the term 'must' is used then a database constraint is enforced.

7.2 Summary of rules

Nonfish_bycatch trip record (t_trip)

trip_key Must be a unique integer greater than zero.

vessel_key Must be a valid vessel key.

start_date Must be a legitimate date after (and including) December 1996.

end_date Must be a legitimate date after (and including) December 1996.

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

observer Must be a legitimate value, (i.e., None, Company, MFish, Both).

signature Must be a valid 1 character code ("Y" = yes, "N" = no).

Nonfish_bycatch station record (t_station)

trip_key Must be equal to a trip_key as listed in the t_trip table.

station_key Must be a unique integer greater than zero for a given trip_key.

target_sp Must be a valid 3 character fish species code as listed in the *curr_spp* table in the **rdb** database.

fishing_method Must be a valid 3 character fishing code, as listed in Appendix 1.

datetime The start date of the station must be a legitimate date. Time caught must be a valid 24-hour time between 0000-2359.

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

latitude Must be a real number that represents a valid latitude and the value in degrees should fall within the range of –33 to –56.

longitude Must be a real number that represents a valid longitude and the value in degrees should fall within the range of 164 to 190.
(Note: values greater than 180 degrees represent values traditionally recorded as west of Greenwich).

Nonfish_bycatch catch record (t_catch)

	Multiple columns check on trip_key and station_key: The combination of trip_key and station_key must exist in the *t_station* table.
catch_key	Must be a unique integer greater than zero for a given catch_key.
species	Must be a valid 3 character species code as listed in the *curr_spp* table in the **rdb** database and should be one of the valid code as listed in Appendix 1.
number_caught	Must be an integer greater than zero.
sex	Must be a valid sex code ("0" = unsexed, "1" = male, "2" = female, "3" = unknown-unable to determine).
alive	Must be a valid 1 character code ("D" = Dead, or "A" = Alive).
retained	Must be a valid 1 character code ("Y" = yes, "N" = no).

8 Acknowledgements

The authors thank Brian Sanders for providing technical assistance and David Fisher and Kevin Mackay for technical advice and review of the draft manuscript.

Appendix 1 - Reference code tables

The information listed in this Appendix is current at the time of writing, and as implemented at July 2003.

Fishing method codes
The valid fishing method codes are provided in the **Non-Fish Incidental Catch Reporting Form** instructions.

Code	Method
BLL	Bottom Longline
BPT	Bottom Pair Trawl
BT	Bottom Trawl
DL	Dahn Line
DPS	Danish Purse Seine
DS	Danish Seine
FN	Fyke Net
HL	Handline
MPT	Mid Water Pair Trawl
MW	Mid Water Trawl
PS	Purse Seine
SJ	Squid Jigging
SLL	Surface Longline
SN	Gill Net (includes Set Net)
T	Troll
TL	Trot Line
TR	Trawl (generic code that is not provided in the **Non-Fish Incidental Catch Reporting Form** instructions. This code has been used where it is obvious that the fishing method is either BT , BPT, MW, or MPT but it is unclear which of these methods was used).

Species codes
The valid species codes are in provided in the **Non-Fish Incidental Catch Reporting Form** instructions.

Code	Name
<u>Mammals</u>	
BDO	Bottlenose Dolphin
CDD	Common Dolphin
DDO	Dusky Dolphin
HDO	Hectors Dolphin
DOT	Dolphin (other)
HSL	Hooker's Sealion
FUR	NZ Furseal
ORC	Killer Whale
SEA	Seal (other)
PIW	Pilot Whale (longfinned)
SMW	Small Whale (unidentified)

Seabirds	
XAL	Albatross
XPE	Petrel
XSL	Seabird (large)
XSS	Seabird (small)

Appendix 2 - Data entry, error checking, and loading

The data in **nonfish_bycatch** have come from the fishing industry.

This section outlines the flow of paper-recorded data, from collection through to its availability to researchers for analysis, and defines the separate tasks that are required to do this.

In summary, the nonfish_bycatch data are recorded on hand written paper forms. Each trip is identified by a unique trip_key, each tow or set by a unique station_key, and each capture by a unique catch_key.

1. **Pre-key entry, visual checking and batching:**
The data are forwarded via MFish, to a project team member, who checks the forms, 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 an 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 is keyed in twice and the two results are cross-checked 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 **nonfish_bycatch**) and now become available for extraction and analysis.

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

Page____of____

NON-FISH INCIDENTAL CATCH REPORTING FORM

Vessel Name: _____

Vessel Registration No.: _____

Trip date from:_____ **to:** _____

Observer Presence:	None	Company	MFISH	Both

Registered Owner: _____

Registered Owner's QRN: _____

Vessel Master:_____

Pursuant to the Marine Mammals Protection Act 1978 and the Wildlife Act 1953, I give notice that the following were accidentally caught:

TARGET FISH SPECIES	FISHING METHOD	DATE	TIME	LATITUDE deg.	LATITUDE min.	LATITUDE S	LONGITUDE deg.	LONGITUDE min.	LONGITUDE E/W	SPECIES CODE	NUMBER CAUGHT	SEX (M/F/U)	DEAD ALIVE	RETAINED

All details set out above are true and correct to the best of my knowledge.

_____

Master of Vessel (signature)

____/____/____ (date)

Appendix 3 - Data form