Database documentation for the Ministry for Primary Industries

beach survey database

beach

K. A. Mackay
and
D.O. Fisher

NIWA Fisheries Data Management
Database documentation series

Revised January 2015

Revision History

Version	Change	Date	Person Responsible
1.0	First documented release as NIWA Internal Report No. 46	Mar 1999	Kevin Mackay
1.1	?. Includes business rules	Jan 2001	Kevin Mackay
1.2	Revised	23 Jun 2004	Kevin Mackay
1.3	Added proj_code to t_survey table	May 2005	Fred Wei
1.4	Added rdb reference data to Appendix 1, and refined some column comments.	Jul-Oct 2012	David Fisher
2.0	Postgres version	Jan 2015	D Fisher, F Wei

Table of Contents

List of Figures

1 Database Document Series

The National Institute of Water and Atmospheric Research (NIWA) currently carries out the role of Data Manager and Custodian for the fisheries research data owned by the Ministry 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 is a brief introduction to the beach sampling database **beach**, and is a part of the database documentation series produced by NIWA.

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

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

2 Beach Sampling Programmes

Beach surveys are used to monitor the population size and length structure of selected shellfish species at various beaches within the inter-tidal zone. Traditionally, these surveys have been done for toheroa (*Paphies ventricosa*), but lately have included cockles (*Austrovenus stutchburyi*), large wedge shells (*Macomona liliana*), pipi (*Paphies australis*), tuatua (*Paphies subtriangulata*) and sea lettuce (*Ulva*). Predominately, beaches are sampled with transect lines, with samples dug from quadrats along the transect line, and the beach material sieved for shellfish.

Surveys of beaches are either non-stratified, from transects sampled at fixed distance intervals[1], or as randomly selected transects within strata defined on the beach. Usually, a two-phase stratified random design[2] is adopted. Individual beaches are broken up into rectangular strata based on previous survey results or previous phases of the same survey. Stratification incorporates both upshore and alongshore variation, and is usually optimised across all target shellfish species to produce the best overall *c.v.*s

In some surveys, the starting location of each transect within a stratum is pre-determined using a random number generator, for both the x co-ordinate (distance along the stratum boundary) and y co-ordinate (0 – 10 m from the stratum boundary). Transects are run from these points down or up

[1] E.g., Carbines, G.D, 1997. Survey of toheroa at Oreti Beach, June 1996. *NIWA Technical Report 1. 1997*. 10p.

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

the beach. Quadrats are sampled at regular 10 m intervals along each transect, and run until either the boundary of another stratum is encountered, or until no more habitat is available; i.e. low-tide mark, or top of beach. The basic sampling unit for analysis is the mean quadrat density for each transect, rather than that for each individual quadrat.

At each sampling point, one quadrat is sampled, where the usually quadrat is a box of 0.1 m^2 and 0.15 m depth and is pushed into the sediment until flush with the surface. The contents of this are dug out and passed through a aperture sieve. All target species are counted and measured down to the nearest millimetre total shell length.

During data analysis, stratum boundaries (and hence areas) are modifed where sampling is found to extend greater or lesser distances down / up the beach than predicted from previous surveys. Areas of unsuitable habitat; e.g. rock platforms are also removed from strata area estimates if present (total available habitat area appeared to flucuate slightly from year to year at some beaches, depending on sand movement). Total population sizes are calculated using scaled estimates of population sizes from each stratum.

Data from previous surveys are used to refine sampling strategies in subsequent programmes. For example, in the 1997/98 Auckland regional beach survey (see Appendix 1 for survey locations), four hundred sampling quadrats were simply assigned to each surveyed beach. In 1998/99, a more adaptive strategy was possible. Four hundred quadrats were initially assigned per beach. Three hundred of these were assigned to phase 1 sampling, and were allocated across strata to acheive the smallest coefficient of variance (*c.v.*) possible. The number of transects per beach varied depending on beach size and the stratification adopted. The remaining one hundred quadrats were deployed as phase 2 stations, and assigned to achieve a maximal reduction in the species with the highest *c.v.* However, phase 2 allocation was done across all beaches; i.e., where a target c.v. had already been reached for a beach during phase 1 sampling, the phase 2 quadrats were utilised on other, more variable beaches.

3 Data Structures

3.1 Table Relationships

This database encompasses several tables. The ERD for **beach** (Figure 1) shows the physical data model structure[3] of the database and it's entities (each entity is implemented as a database *table*) and relationships between these tables. Each table represents an object, event, or concept in the real world that has been represented in the database. Each *attribute* of a table is a defining property or quality of the table.

All of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary key[4]. This schema is valid regardless of the database system chosen, and it can remain

[3] Also known as database *schema*

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

correct even if the Database Management System (DBMS) is changed. Most of the tables in the **beach** database also contain special attributes, called foreign keys[5].

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

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

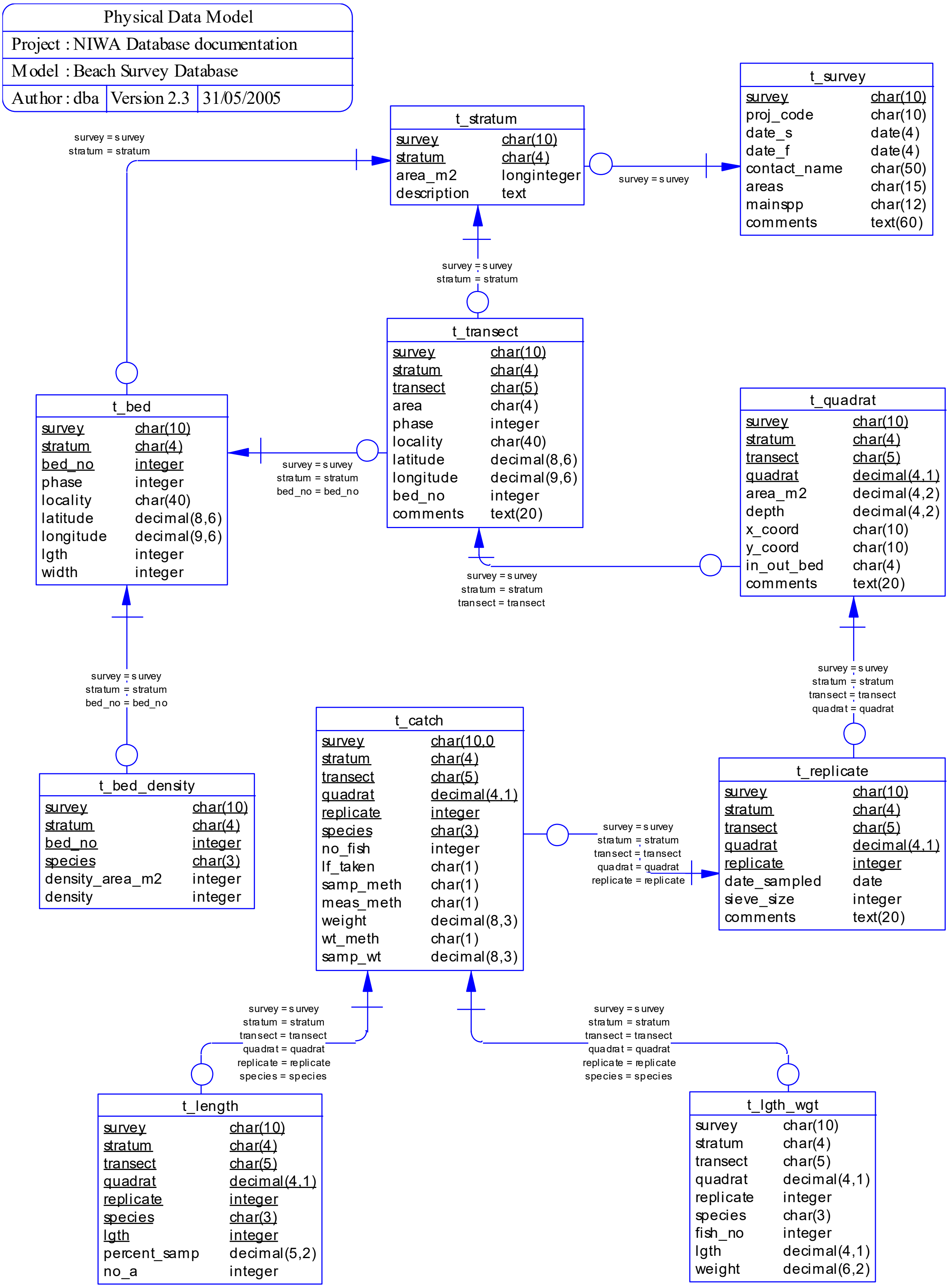
where the attribute(s) make up the primary key (the key attributes) and the index name is the primary key name. Note that the typographical convention for the above format is that square brackets [] may contain more than one item or none at all. A unique index prevents records with duplicate key values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

The **beach** 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 **beach** are of the type *one-to-many*[6]. This is shown in the ERD by connecting a single line (indicating 'many') from the child table; e.g., *t_quadrat*, to the parent table; e.g., *t_transect*, with an arrowhead (indicating 'one') pointing to the parent. For example, consider the relationship between the tables' *t_transect* (the parent table) and *t_quadrat* (the child table). Any one transect in *t_transect* can have one or more quadrats in *t_quadrat*, but any one quadrat can only be a part of one transect. 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_transect* and *t_quadrat*, which has a mandatory 'one' and an optional 'many'. This means that one transect record can have zero or more (many) quadrats within it, but one quadrat must have one, and only one, associated record in the transect table.
The ERD's in this document show attributes within the tables with generic data-types.

[5] 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.

[6] A one-to-many relationship is where one record (the *parent*) in a table relates to one or many records (the *child*) in another table; e.g., one survey in *t_survey* can have many strata in *t_stratum* but any one stratum can only come from one survey.



Physical Data Model		
Project : NIWA Database documentation		
Model : Beach Survey Database		
Author : dba	Version 2.3	31/05/2005

t_survey	
survey	char(10)
proj_code	char(10)
date_s	date(4)
date_f	date(4)
contact_name	char(50)
areas	char(15)
mainspp	char(12)
comments	text(60)

t_stratum	
survey	char(10)
stratum	char(4)
area_m2	longinteger
description	text

t_transect	
survey	char(10)
stratum	char(4)
transect	char(5)
area	char(4)
phase	integer
locality	char(40)
latitude	decimal(8,6)
longitude	decimal(9,6)
bed_no	integer
comments	text(20)

t_bed	
survey	char(10)
stratum	char(4)
bed_no	integer
phase	integer
locality	char(40)
latitude	decimal(8,6)
longitude	decimal(9,6)
lgth	integer
width	integer

t_quadrat	
survey	char(10)
stratum	char(4)
transect	char(5)
quadrat	decimal(4,1)
area_m2	decimal(4,2)
depth	decimal(4,2)
x_coord	char(10)
y_coord	char(10)
in_out_bed	char(4)
comments	text(20)

t_bed_density	
survey	char(10)
stratum	char(4)
bed_no	integer
species	char(3)
density_area_m2	integer
density	integer

t_catch	
survey	char(10,0
stratum	char(4)
transect	char(5)
quadrat	decimal(4,1)
replicate	integer
species	char(3)
no_fish	integer
lf_taken	char(1)
samp_meth	char(1)
meas_meth	char(1)
weight	decimal(8,3)
wt_meth	char(1)
samp_wt	decimal(8,3)

t_replicate	
survey	char(10)
stratum	char(4)
transect	char(5)
quadrat	decimal(4,1)
replicate	integer
date_sampled	date
sieve_size	integer
comments	text(20)

t_length	
survey	char(10)
stratum	char(4)
transect	char(5)
quadrat	decimal(4,1)
replicate	integer
species	char(3)
lgth	integer
percent_samp	decimal(5,2)
no_a	integer

t_lgth_wgt	
survey	char(10)
stratum	char(4)
transect	char(5)
quadrat	decimal(4,1)
replicate	integer
species	char(3)
fish_no	integer
lgth	decimal(4,1)
weight	decimal(6,2)

Figure 1: Entity Relationship Diagram (ERD) for the beach database.

These relationships are enforced in the database by the use of foreign key constraints. 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 **beach** prevent these from occurring. Constraints are shown in the table listings by the following format:

Foreign-key constraints:
  "foreign key name" FOREIGN KEY (attribute[,attribute]) REFERENCES
   parent table (attribute [, attribute])

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

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

Foreign-key constraints:
  "fk_t_transect_t_stratum" FOREIGN KEY (survey, stratum) REFERENCES t_stratum(survey, stratum)

This means that the value of the combination of attributes *survey* and *stratum*, (that is, one stratum) in the current transect record must already exist in the parent table *t_stratum* or the record will be rejected and the following message will be displayed:

ERROR: insert or update on table "t_transect" violates foreign key constraint "fk_t_transect_t_stratum"

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

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

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

3.2 Database Design

At the top level of **beach** is the table *t_survey* (Table 1), which contains all details of individual beach surveys including, dates, areas, and species involved. Individual surveys are identified by a unique attribute survey, which typically comprises of an abbreviated form of the beach name followed by a 2-digit year and a 2-digit sequential number; e.g., how9501 is the first survey at Howick Beach in 1995. Both the areas covered by a survey, and the species involved are recorded in attributes *areas* and *mainspp* respectively. These attributes are a comma-separated list of area and species codes. Descriptions of the codes used can be found in the table *area_codes* for areas, and *curr_spp* for species in the **rdb** database. These codes can also be found via http://marlin.niwa.co.nz/

Each survey has at least one stratum record in the table *t_stratum* (Table 2). Strata are each given a stratum code, recorded as the attribute *stratum*, which is unique within a survey. Non-stratified surveys effectively become single stratum surveys and are given a default stratum record.

Within each stratum, transect lines are established for sampling and these transect details are recorded in the table *t_transect* (Table 3). Transect are identified by the attribute *transect* that is unique within a stratum. Positions of the start of transects are recorded in the attributes *latitude*, *longitude*, and *E_W*. The attribute *phase* records the phase number for transects in two-phase surveys.

Along these transect lines quadrats are sampled. Quadrats are usually a box-like frame that is pressed into the sand until the top is level with the beach surface. This defines a fixed area and depth that is then sampled. Quadrat records are recorded in the table t_quadrat (Table 4). Individual quadrats are identified either as a sequential number along a transect line, or as a distance along the transect line. In some surveys, the position of each quadrat is recorded as x and y coordinates from some fixed point in the stratum. These coordinates are recorded in the attributes *x_coord* and *y_coord* respectively.

Quadrats may require several replicates, for example when sand is sieved in two different grades of sieve mesh. Replicate details, including date sampled and mesh size, are recorded in the table *t_replicate* (Table 5).

The catch for each species, as a result of sieving, is recorded in the table *t_catch* (Table 6). The catch of a species may be represented by: a number of specimens caught (as represented by the attribute *no_fish*); and/or the total species weight (as represented by the attribute *weight*). The attribute *lf_taken* is a Y/N flag to indicate whether length frequencies were taken for that particular species. There are three attributes within *t_catch*, which are codes to define various methodologies, and as a result, have foreign key constraints to tables in the **rdb** database (Figure 2):

- *samp_meth* – a code to indicate the method used to select which shellfish were selected for further sampling.
- *meas_meth* – a code to indicate the method used to measure the lengths of individual shellfish for length frequencies.
- *wt_meth* – a code to indicate the method by which the species weight was calculated.

Definitions of these three codes can be found in the tables *t_samp_sel_codes*, *t_fish_meas_codes*, and *t_wgt_meth_codes* respectively in the **rdb** database. Similarly, the attribute species is also constrained by a foreign key to the table *curr_spp* in the **rdb** database.

Length frequencies are recorded in the table *t_length* (Table 7) with the attribute *no_a* recording the number of shellfish at *lgth*, the millimetre class. The attribute *percent_samp* represents the percentage of the total catch, as recorded in *t_catch*, which was measured.

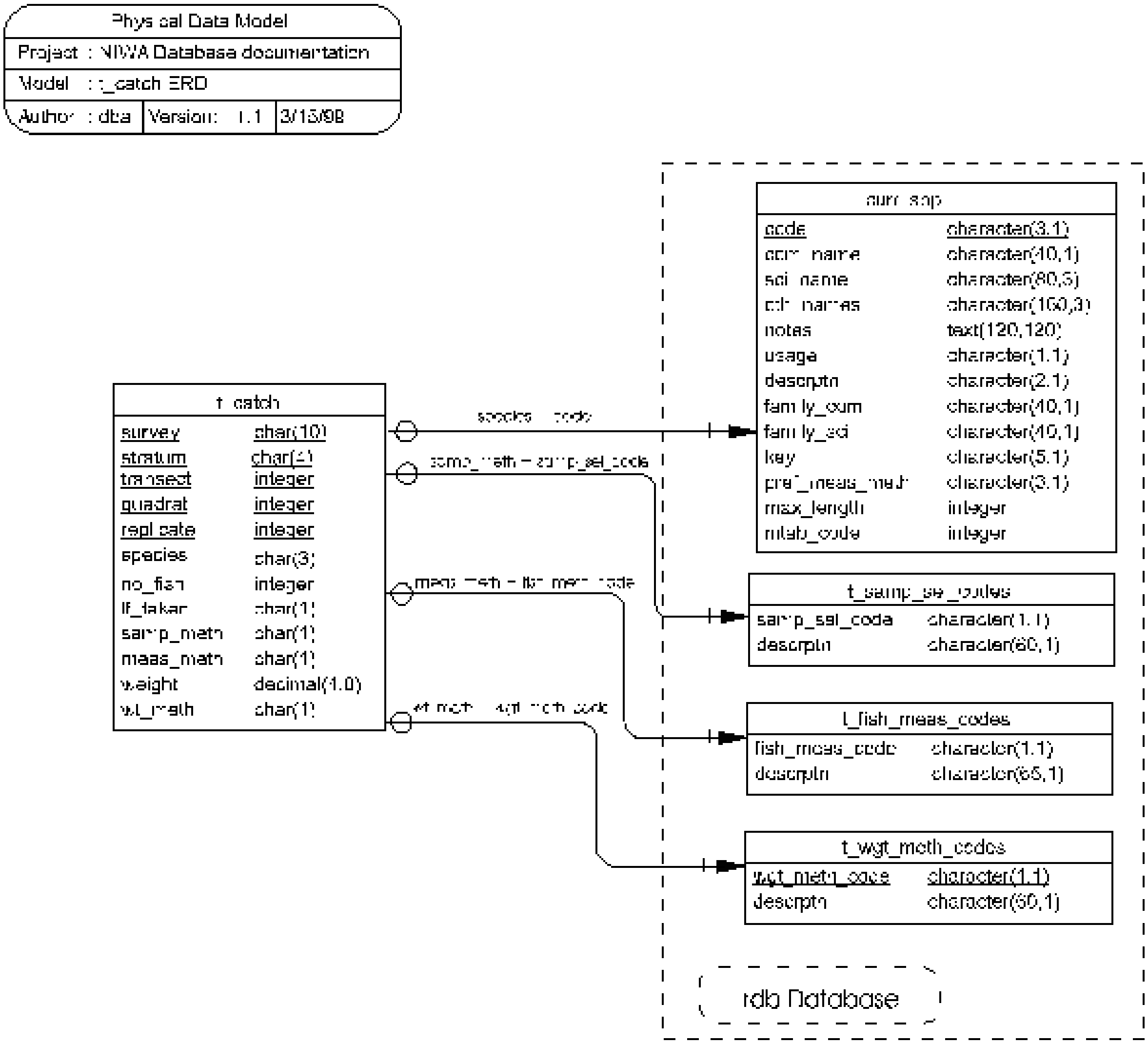


Figure 2: ERD showing the relationship between the table t_catch and the rdb database.

Lengths and weights of individual shellfish are recorded in the table *t_lgth_wgt* (Table 8). These lengths are included in the length frequencies as recorded in *t_length*.

Often, beach surveys target discrete beds of a particular flora or fauna species; e.g. ulva (sea lettuce) or toheroa beds. These beds are usually identified by previous surveys or by earlier phases of the same survey. The dimension and location details for such beds are recorded in the table *t_bed* (Table 9). The densities of these species within these beds are recorded in the table t_bed_density (Table 10).

4 Table Summaries

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

1 **t_survey :** contains details of individual beach surveys. Usually a survey is restricted to just one beach.
2 **t_stratum :** contains details of strata used for individual beach surveys.
3 **t_transect :** contains details of individual transects used during a survey.
4 **t_quadrat :** contains details of each quadrat used within a transect in a beach survey.
5 **t_replicate :** contains details of each replicate within a quadrat.
6 **t_catch :** contains catch details for each species caught in a replicate
7 **t_length :** contains length frequency measurements for animals caught.
8 **t_lgth_wgt :** contains length and weight data for individual animals caught.
9 **t_bed :** contains dimensions and location details for flora or fauna beds along a beach; e.g., ulva (sea lettuce) or toheroa beds.
10 **t_bed_density :** contains data on densities or a flora or fauna species within a bed.

5 beach Tables

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

5.1 Table 1: t_survey

Comment: Details of individual beach surveys. Usually, a survey is restricted to just one beach.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, typically comprised of a 3 to 4 character abbreviated form of the beach name, followed by 2-digits for the year of the survey, and then a 2-digit sequential number.
proj_code	character varying(16)		Project code for the project that funded collection of the data.
date_s	date		Start date of the survey.
date_f	date		Finish date of the survey.
contact_name	character varying(50)		Name of contact person regarding the survey data.
areas	character varying(15)		List of 4-character area codes, separated by commas, for areas where the survey takes place. Refer rdb.area_codes or http://marlin.niwa.co.nz for code descriptions.
mainspp	character varying(12)	No	List of 3-character species codes, separated by commas, for species involved in the survey. Refer rdb.curr_spp or Appendix 1 of the database documentation for codes.
comments	text		General comments about the survey, including name of the beach surveyed.

```
Indexes:
   "pk_t_survey" PRIMARY KEY, btree (survey)
```

```
Check constraints:
    "t_survey_proj_code_check" CHECK (proj_code::text ~ '[A-Z]+[0-9]+'::text)
```

5.2 Table 2: t_stratum

Comment: Details of strata used for individual beach surveys.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4-character stratum code. Unique for each stratum within an individual survey. Default="1".
area_m2	integer		Area (sq. metres) of the stratum.
descrptn	text		General description of the stratum, preferably with latitude and longitude coordinates.

```
Indexes:
   "pk_t_stratum" PRIMARY KEY, btree (survey, stratum)

Foreign-key constraints:
   "fk_t_stratum_t_survey" FOREIGN KEY (survey)
    REFERENCES beach.t_survey(survey)
```

5.3 Table 3: t_transect

Comment: Details of individual transects used during a survey.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4-character stratum code (Default="1").
transect	character varying(5)	No	Transect number. Unique within each stratum.
area	character varying(4)	No	4 character area code. Refer rdb.area_codes or http://marlin.niwa.co.nz.
phase	integer		Phase number. Used for two-phase survey design.
locality	character varying(40)		General descriptive locality of the transect.
latitude	numeric(8,6)		Latitude (decimal degrees, -90 to +90) of the transect.
longitude	numeric(9,6)		Longitude (decimal degrees, 0 to 360) of the transect.
bed_no	integer		Identifier number of the bed that the transect runs through. Refer t_bed.
comments	text		General comments about the transect.
position	geometry		Position of the transect as gis point type.

```
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:
   "fk_t_transect_t_bed" FOREIGN KEY (survey, stratum, bed_no)
    REFERENCES beach.t_bed(survey, stratum, bed_no)
   "fk_t_transect_t_stratum" FOREIGN KEY (survey, stratum)
    REFERENCES beach.t_stratum(survey, stratum)
```

5.4 Table 4: t_quadrat

Comment: Details of each quadrat used within a transect in a beach survey.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4-character stratum code (Default="1").
transect	character varying(5)	No	Transect number. Unique within each stratum.
quadrat	numeric(4,1)	No	Quadrat number. May be distance (m) along the transect from a fixed point (such as the mean high-tide mark).
area_m2	numeric(5,3)		Area (sq. metres) for each quadrat.
depth	numeric(4,2)		Depth (m) quadrat dug to.
x_coord	character varying(10)		X coordinate for the quadrat.
y_coord	character varying(10)		Y coordinate for the quadrat.
in_out_bed	character varying(4)		Was quadrat in or out of the bed, values IN, OUT or NONB. Generally Not Applicable so Null.
comments	text		General comments about the quadrat.

```
Indexes:
   "pk_t_quadrat" PRIMARY KEY, btree
    (survey, stratum, transect, quadrat)
```

5.5 Table 5: t_replicate

Comment: Details of replicates within a quadrat.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4-character stratum code (Default="1").
transect	character varying(5)	No	Transect number.
quadrat	numeric(4,1)	No	Quadrat number. May be distance (m) along the transect from a fixed point (such as the mean high-tide mark).
replicate	integer	No	Unique replicate number (within a quadrat).
date_sampled	date		Date quadrat replicate sampled.
sieve_size	integer		Sieve size (mm) used for replicate.
comments	text		General comments about the replicate.

```
Indexes:
   "pk_t_replicate" PRIMARY KEY, btree
    (survey, stratum, transect, quadrat, replicate)

Foreign-key constraints:
   "fk_t_replicate_t_quadrat" FOREIGN KEY (survey, stratum, transect,
                                          quadrat)
    REFERENCES beach.t_quadrat(survey, stratum, transect, quadrat)
```

5.6 Table 6: t_catch

Comment: Catch details for each replicate by species.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	Up to 4 character stratum code (Default="1").
transect	character varying(5)	No	Transect number.
quadrat	numeric(4,1)	No	Quadrat number.
replicate	integer	No	Replicate number.
species	character(3)	No	3 character species code. Refer rdb.curr_spp or Appendix 1 of the database documentation.
no_fish	integer		Number of the species caught.
lf_taken	character varying(1)		1-character code to indicate if length frequencies were taken. Y=Yes; N=No.
samp_meth	character varying(1)		1-character code for the method used to select a sample for length measurements. Refer rdb.t_samp_sel_codes or Appendix 1 of the database documentation.
meas_meth	character varying(1)		1-character code for method used to measure the species. Refer rdb.t_fish_meas_codes. Typically = 8 as Shell Length - maximum dimension of the shellfish (except oysters).
weight	numeric(8,3)		Weight (kg) of the species caught.
wt_meth	character varying(1)		1-character code for method used to weigh catch. Refer rdb.t_wgt_meth_codes or Appendix 1 of the database documentation.
samp_wt	numeric(8,3)		Weight (grams) of the sample taken from the whole catch for further measurement.

```
Indexes:
   "pk_catch" PRIMARY KEY, btree
    (survey, stratum, transect, quadrat, replicate, species)
Check constraints:
   "t_catch_lf_taken_check" CHECK (lf_taken::text ~ '[YN]'::text)

Foreign-key constraints:
   "fk_t_catch_curr_spp" FOREIGN KEY (species)
    REFERENCES rdb.curr_spp(code)
   "fk_t_catch_t_fish_meas_codes" FOREIGN KEY (meas_meth)
    REFERENCES rdb.t_fish_meas_codes(fish_meas_code)
   "fk_t_catch_t_samp_sel_codes" FOREIGN KEY (samp_meth)
    REFERENCES rdb.t_samp_sel_codes(samp_sel_code)
   "fk_t_catch_t_wgt_meth_codes" FOREIGN KEY (wt_meth)
    REFERENCES rdb.t_wgt_meth_codes(wgt_meth_code)
```

5.7 Table 7: t_length

Comment: Length frequency measurements for specimens caught.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4 character stratum code (Default = "1").
transect	character varying(5)	No	Transect number.
quadrat	numeric(4,1)	No	Quadrat number.
replicate	integer	No	Replicate number.
species	character(3)	No	3-character species code. Refer rdb.curr_spp or Appendix 1 of the database documentation.
lgth	integer	No	Length class (mm).
percent_samp	numeric(5,2)		Percent (%) of total catch measured.
no_a	integer		Number of animals at this length.

```
Check constraints:
   "ckc_percent_samp_t_length" CHECK (percent_samp IS NULL OR percent_samp
     >= 0::numeric AND percent_samp <= 100::numeric)
   "t_length_percent_samp_check" CHECK (percent_samp >= 0.00 AND
     percent_samp <= 100.00)

Foreign-key constraints:
   "fk_t_length_curr_spp" FOREIGN KEY (species)
    REFERENCES rdb.curr_spp(code)
```

5.8 Table 8: t_lgth_wgt

Comment: Length and weight data for individual animals caught.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4 character stratum code (Default="1").
transect	character varying(5)	No	Transect number.
quadrat	numeric(4,1)	No	Quadrat number.
replicate	integer	No	Replicate number.
species	character(3)	No	3 character species code. Refer rdb.curr_spp.
fish_no	integer		Unique sequential number for each specimen in a sample.
lgth	numeric(4,1)		Length (mm).
weight	numeric(6,2)		Weight (grams).

```
Foreign-key constraints:
   "fk_t_lgth_wgt_curr_spp" FOREIGN KEY (species)
    REFERENCES rdb.curr_spp(code)
```

5.9 Table 9: t_bed

Comment: Contains dimensions and location details for flora or fauna beds along a beach; e.g., ulva (sea lettuce) or toheroa beds.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4-character stratum code (Default="1").
bed_no	integer	No	Unique number to identify each bed.
phase	integer		Phase number. Used for two-phase survey design.
locality	character varying(40)		General descriptive locality of the bed.
latitude	numeric(8,6)		Latitude (decimal degrees, -90 to +90) of the bed.
longitude	numeric(9,6)		Longitude (decimal degrees, 0 to 360) of the bed.
bed_length	integer		Length (metres) of the bed.
bed_width	integer		Width (metres) of the bed.

```
Indexes:
   "pk_t_bed" PRIMARY KEY, btree (survey, stratum, bed_no)

Foreign-key constraints:
   "fk_t_bed_t_stratum" FOREIGN KEY (survey, stratum)
    REFERENCES beach.t_stratum(survey, stratum)
```

5.10 Table 10: t_bed_density

Comment: Contains data on densities of a species within a bed.

Column	Type	Null?	Description
survey	character varying(10)	No	Unique identifier for each survey, as defined for the t_survey table.
stratum	character varying(4)	No	4-character stratum code (Default="1").
bed_no	integer	No	Unique number to identify each bed.
species	character(3)	No	3-character species code. Refer rdb.curr_spp or Appendix 1 of the database documentation.
density_area_m2	integer		Surface area (sq. metres) by which the species density is calculated for.
density	integer		Density (number of the species per density_area_m2).

```
Indexes:
   "pk_bed_density" PRIMARY KEY, btree
    (survey, stratum, bed_no, species)
```

6 beach business rules

6.1 Introduction to business rules

The following are a list of business rules applying to the **beach** 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 beach 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.

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

6.2 Summary of rules

Beach survey details (t_survey)

survey Survey code must be present and be a unique value. Survey codes are of the format: up to a 6-character alphanumeric to denote the location; followed by a two-digit year ranging from 70-99 and 00 on for years since 1970; and a 2-digit sequential number to denote the number of surveys carried out at that location for that year.

date_s Start date must be a valid date.

date_f Finish date must be a valid date.

Multiple column check on start date and finish date:
Finish date must be equal to, or after, the survey start date.

areas Each area code within the areas list must be a valid area code as listed in the *area_codes* table of the **rdb** database. A comma must separate each area code.

mainspp Each species code within the main species list must be a valid species code as listed in the *curr_spp* table of the **rdb** database. Must have a value entered and a comma must separate each species code.

Stratum details (t_stratum)

survey Must be valid survey code as listed in the *t_survey* table.

stratum Must have a value entered and be a unique value within a survey. Stratum code must equal “1” or any combination of up to 4 alphanumeric characters.

area_m2 Must be an integer greater than zero.

Transect line details (t_transect)

Multiple column checks on survey and stratum:
The combination of survey code and stratum code must exist in the *t_stratum* table.

transect Transect number must be present and be a unique value within a stratum. Can be any combination of up to 5 alphanumeric characters.

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

phase Must be an integer greater than zero and should be within the reasonable range of 1 to 2.

latitude Must be a valid latitude within the range of 90 to –90.

longitude Must be a valid longitude within the range of 0 to 360.

bed_no Must be a valid bed number as listed in the *t_bed* table.

Quadrat details (t_quadrat)

Multiple column checks on survey, stratum and transect number:
The combination of survey code stratum code and transect number must exist in the *t_transect* table.

quadrat Quadrat number must be entered, greater than zero, and be a unique value within a transect line.

area_m2 Must be a number greater than zero and should be within the reasonable range of 0.06 to 1.00.

depth Must be a number greater than zero and should be within the reasonable range of 0.10 to 0.30

x_coord Should be either: (i) an integer between the range of 0 to 1600 for metres along an x-axis; (ii) an integer between the range of 1660000 to 1790000 representing a valid longitude in degrees and decimal minutes; (iii) a short text describing an azimuth and a distance.

y_coord Should be either: (i) an integer between the range of -45 to 500 for metres up and down an y-axis; (ii) an integer between the range of 340000 to 470000 representing a valid latitude in degrees and decimal minutes;

in_out_bed Must be a valid value as listed in Appendix 1.

Replicate details (t_replicate)

Multiple column checks on survey, stratum, transect and quadrat number:
The combination of survey code stratum code, transect and quadrat number must exist in the *t_quadrat* table.

replicate Replicate number must be entered and be a unique integer, greater than zero, for each quadrat. Should be within the reasonable range of 1 to 2.

date_sampled Must be a valid date.

Multiple column check on date sampled, survey start date and survey finish date:
Date sampled must be on or after the survey start date and on or before the survey finish date.

sieve_size Must be a integer greater than zero and should be within the reasonable range of 2 to 5.

Catch details (t_catch)

Multiple column checks on survey, stratum, transect, quadrat and replicate number:
The combination of survey code stratum code, transect, quadrat and replicate number must exist in the *t_replicate* table.

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

no_fish Must be an integer greater than zero and should be within the reasonable range of 1 to 1000.

Multiple column checks on species and number of fish:
The number of fish per species should be less than the reasonable maximum as listed in Appendix 1.

lf_taken Length frequency taken flag must be equal to either an “Y” or an “N”.

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

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

weight Must be a number greater zero and should be within the reasonable range of 0.02 to 25.00

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

samp_wt Must be a number greater or equal to zero and should be within the reasonable range of 0.02 to 0.80

Multiple column checks on sample weight and catch weight:
Sample weight must be less than or equal to catch weight.

Multiple column checks on sample weight and sample selection method:
Sample weight must be greater than zero when there is a sample selection method.

Length frequency details table (t_length)

Multiple column checks on survey, stratum, transect, quadrat, replicate number and species code:
The combination of survey code stratum code, transect, quadrat, replicate number and species code must exist in the *t_catch* table.

lgth The fish length must be an integer greater than zero and should be within the reasonable range of 2 to 155.

Multiple columns check on species and length:
The fish length should be within a reasonable range for the species. Minimum and maximum lengths by species are listed in Appendix 1.

percent_samp Must be a number greater than 0 and less than or equal to 100.

no_a Must be a number greater than or equal to zero and within the reasonable range of 0 to 140.

Length and weight details for individual shellfish table (t_lgth_wgt)

Multiple column checks on survey, stratum, transect, quadrat, replicate number and species code:
The combination of survey code stratum code, transect, quadrat, replicate number and species code must exist in the *t_catch* table.

fish_no Must be number greater than zero and be unique within a catch.

lgth The fish length must be an number greater than zero and should be within the reasonable range of 2 to 155.

Multiple columns check on species and length:
The fish length should be within a reasonable range for the species. Minimum and maximum lengths by species are listed in Appendix 1.

weight The weight of the fish must be a number greater than zero and should be within the reasonable range of 0.01 to 125.00

Multiple columns check on species, length and weight:
The fish weight should be less than a reasonable maximum fish weight for the species and be reasonable for the fish length. Minimum and maximum fish weights by species as listed in Appendix 1.

Shellfish bed table (t_bed)

Multiple column checks on survey and stratum:
The combination of survey code and stratum code must exist in the *t_stratum* table.

bed_no Must be an integer greater than one and should be within the reasonable range of 1 to 200.

phase Must be an integer greater than zero and should be within the reasonable range of 1 to 2.

locality Can be any combination of up to 40 ASCII characters.

latitude Must be a valid latitude within the range of 90 to –90.

longitude Must be a valid longitude within the range of 0 to 360.

bed_length Must be an integer greater than zero and should be within the reasonable range of 20 to 300.

bed_width Must be an integer greater than zero and should be within the reasonable range of 5 to 40

Shellfish bed density table (t_bed_density)

Multiple column checks on survey, stratum and bed number:
The combination of survey code, stratum code and bed number must exist in the *t_bed* table.

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

density_area_m2 Must be an integer greater than to zero and should be within the range of 10 to 25.

density Must be an integer greater than or equal to zero and should be within the range of 80 to 3800.

7 Acknowledgments

The author would like to thank Dave Banks for his review and editorial comment for this document, and Mark Morrison for his technical input to the beach survey methodology.

Appendix 1 – Reference code tables

Species code	**Common name** (from rdb:curr_spp)
COC	Cockle
LWS	Large wedge shell
PPI	Pipi
TOH	Toheroa
TUA	Tuatua

These are the main species surveyed in beach surveys, refer rdb:curr_spp or http://marlin.niwa.co.nz for a full list of species codes.

samp_meth	**description** (from rdb:t_samp_sel_codes)
1	Rigorous simple random sample (SRS)
4	Stratified sample
5	Approx. simple random sample
8	Non-random, not representative of whole catch
9	Whole catch

Codes 9 or 5 are those typically used.

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

In or out of bed code

IN	Quadrat is within the targeted bed
OUT	Quadrat is outside the targeted bed
NONB	Quadrat is not targeting a bed.

Number of catch by species

Species	*Maximum catch*
COC	1000
LWS	50
PPI	250
TOH	150

Length ranges by species (millimeters)

Species	*Minimum*	*Maximum*
COC	2	65
LWS	5	60
PPI	2	55
TOH	5	155

Weight ranges by species (grams)

Species	*Minimum*	*Maximum*
COC	0.01	122.05

Appendix 2 – Survey location maps

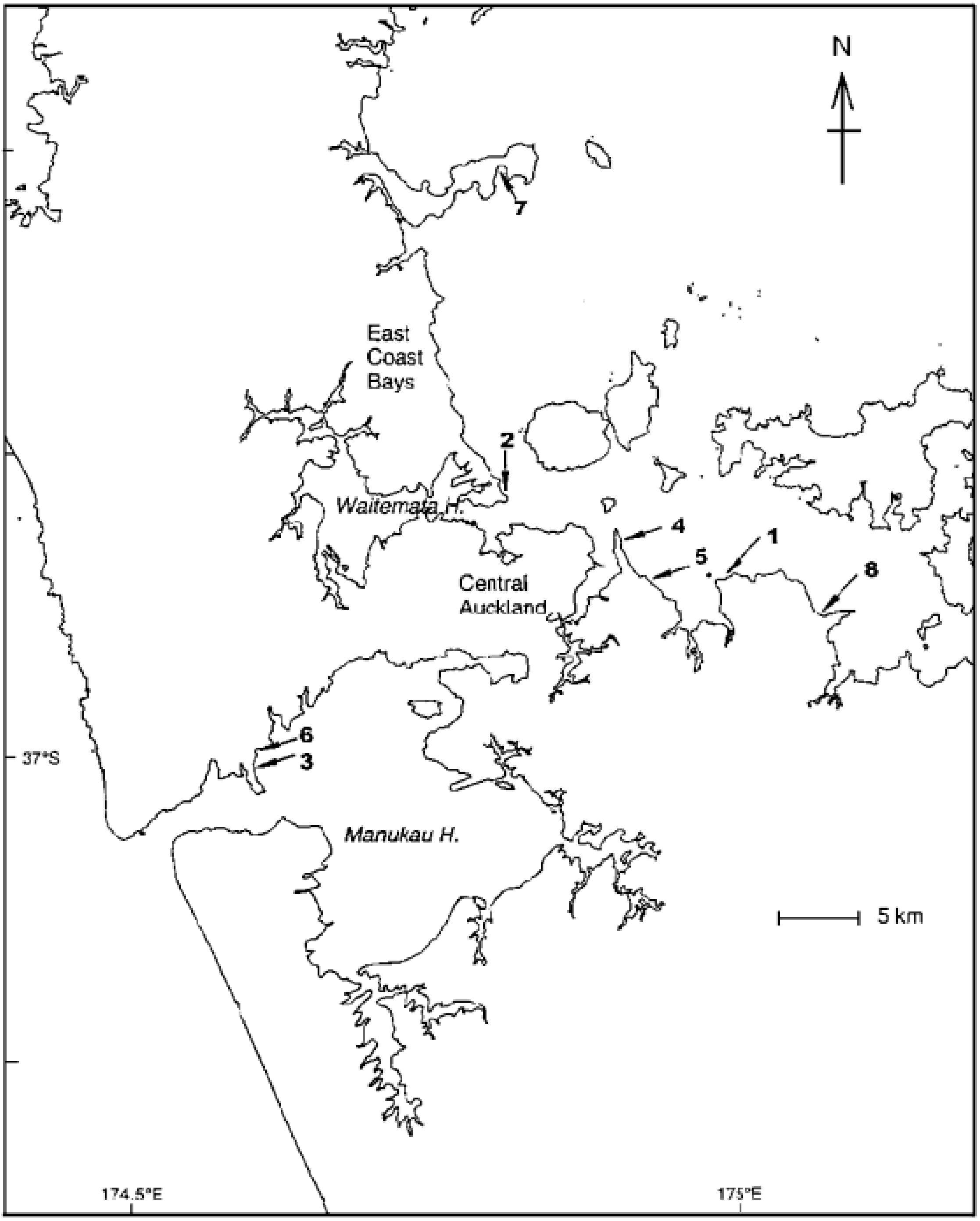


Figure 3: Beach survey locations in the greater Auckland metropolitan area.

1. Beachlands
2. Cheltenham Beach
3. Cornwallis Beach
4. Eastern Beach
5. Howick Beach
6. Mill Bay
7. Okomorai Bay
8. Umupuia Beach