Database documentation for
MPI's Shellfish diseases database:

shellfish

D. O. Fisher
K. Michael
A. Pande
J.Q. Maggs

NIWA Fisheries Data Management
Database Documentation Series

Version 1.1, December 2020

Version Control

Version	Change	Date	Responsible
1.0	First official release	September 2020	D Fisher
1.1	Updated background text, field comments	December 2020	JQ Maggs

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 introduces the database **shellfish** 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 the database.

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

The **shellfish** database has been implemented as a schema within the Postgres database called **fish**. This facilitates linkages to reference tables in other schema in fish, particularly the rdb schema, and tables such as t_vessels.

2. Shellfish diseases sampling programme

2.1 *Bonamia ostreae* sampling programme

Bonamia ostreae is a Haplosporidia protozoan parasite that infects flat oysters, causing high levels of mortality. *Bonamia ostreae* is a notifiable parasite to the World Organisation for Animal Health (OIE) due to the severity of disease they can cause.

In 2015, *Bonamia ostreae* was discovered for the first time in New Zealand in a flat oyster (Ostrea chilensis) farm in the Marlborough Sounds by the Ministry for Primary Industries (MPI) Animal Health Laboratory (AHL). MPI undertook a nationwide delimiting survey to determine the extent of the distribution of this parasite. During the first survey the parasite was not found to exist outside of the Marlborough Sounds. MPI then designed a biannual detection survey that was operationalised by NIWA. Oyster samples were taken from both farmed and wild populations, around New Zealand sampling enough oysters to detect infection, if it were present, at a 2% level. Sample numbers taken from the populations in the Marlborough Sounds were reduced since it had already been determined that infection was present.

In 2017, *Bonamia ostreae* was found in a farm in Big Glory Bay, Stewart Island. The MPI biosecurity response ordered all flat oyster farms to be removed as the presence of the parasite close to the wild flat oyster population was considered a big risk to potentially infecting the oyster fishery in Foveaux Strait. Only one farm and hatchery in Horseshoe Bay (Stewart Island) continued to operate as it was deemed to be in a separate epidemiological unit. Once the oyster farms were removed, the wild populations of oysters around the country and the Horseshoe Bay farm continued to be surveyed in order to determine whether there was any subsequent spread of the parasite.

2.2 Future use

The database has been designed to accommodate potential future results from disease surveillance from other shellfish.

3 Data structures

3.1 Table relationships

This database schema contains several tables. The ERD for **shellfish** (Figure 1) shows the logical structure[1] of the database and its 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.

Each of the table's attributes are shown in the ERD. The underlined attributes represent the table's primary key[2]. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed. The ERD's in this document show attributes within the tables with generic datatypes.

Section 5 shows a listing of all the tables as implemented by the Postgres DBMS. As can be seen in the listing of the tables, each table includes a primary key attribute. In shellfish the primary keys are all integers, and except for table t_survey, are all system generated values. 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. This prevents records with duplicate key values from being inserted into the table, e.g., a new survey being inserted with an existing survey number. 'btree' is a technical reference to the type of index, namely a balanced tree index.

This database is implemented as a relational database. That is, tables are linked to one another by their relationships. All relationships in **shellfish** are of the type *one-to-many*[3]. This is shown in the

---

[1] Also known as a database *schema*.

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

[3] 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 stations in *t_station* but any one station can only come from one survey.

ERD by connecting a single line (indicating 'many') from the child table (e.g., *t_station*) to the parent table (e.g., *t_survey*) with an arrowhead (indicating 'one') pointing to the parent. For example, consider the relationship between the tables *t_survey* (the parent table) and *t_station* (the child table). Any one survey in *t_survey* can have one or more stations in *t_station*, but any one station can only be a part of one survey.

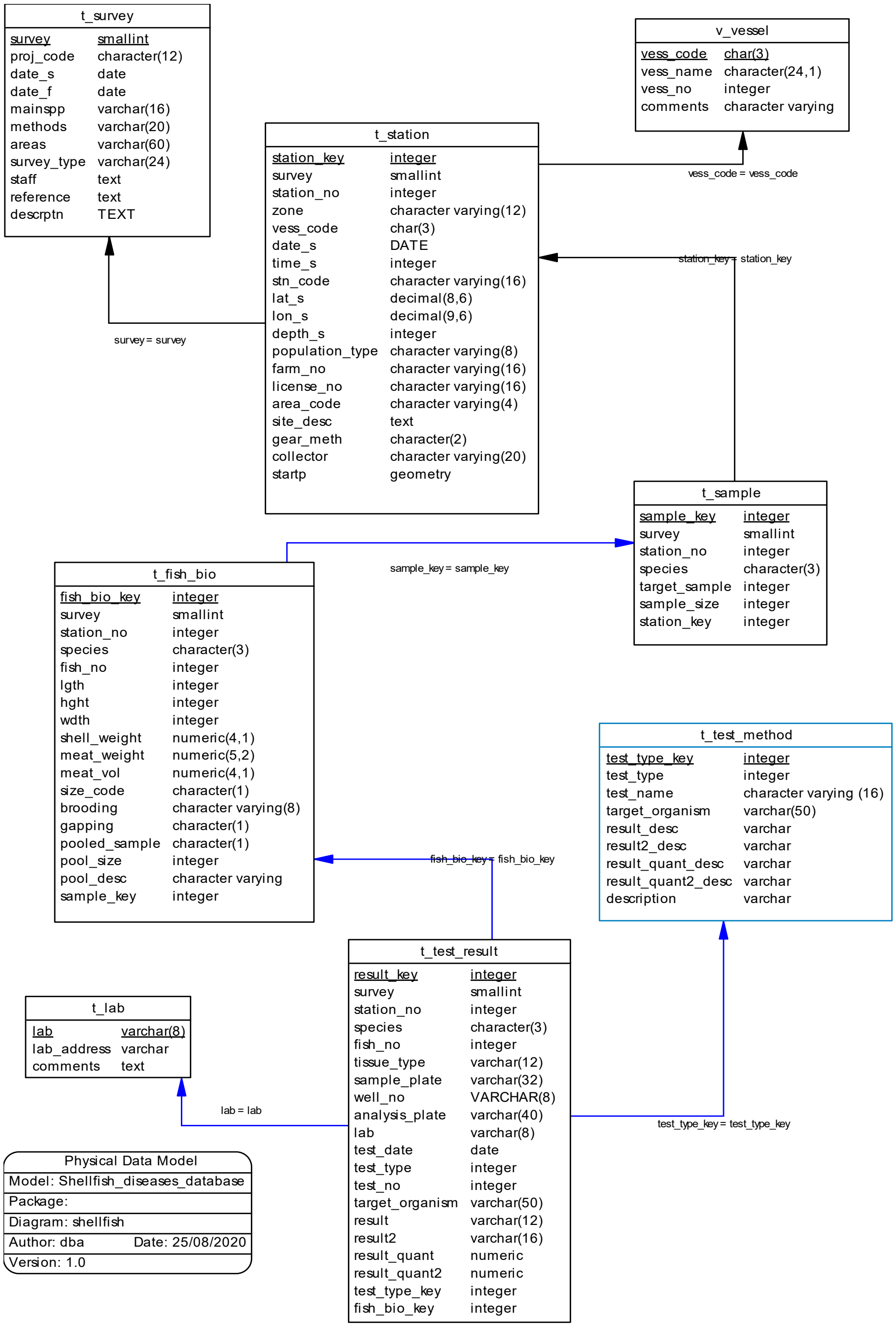

Figure 1: Entity Relationship Diagram (ERD) of the database shellfish

3.2 Database design

The genesis of **shellfish** lies in the need for a central data repository for data collected by the oyster disease surveillance programme, particularly for *Bonamia ostreae*. The data collected from the surveillance surveys were initially held in Microsoft Excel (CSV) files.

3.2.1 Oyster surveillance surveys

Oysters were collected either by snorkelling, diving, fishing (dredge) or by harvesting off the lines depending on the collection site.

Oysters off farmed sites were collected by NIWA, MPI or the farmers. Wild populations were sampled using a combination of dredging from commercial fishing vessels if they were a deep water, commercially harvested population or diving and snorkelling if it was a shallow water non-commercial population. Multiple sites existed within epidemiological units (locations) and were sampled repeatedly at specified intervals.

For these surveillance surveys whole oysters were sampled. Larger oysters were targeted, as were any oysters that were gaping, in order to increase the chance of detection. DNA extracts were taken from the sample tissue. Half the DNA extract and half the tissue from each oyster was put aside and any positive or inconclusive results were confirmed by the MPI. AHL using the remainder DNA extract and tissue. This confirmatory testing ensured that the specificity of the sampling system was 100%. PCR was used as a screening test, and a repeat PCR and DNA sequencing was used as a confirmatory test if necessary.

4 Table summaries

This database is arranged as a set of five main tables containing survey data, and three other tables describing the various codes that are used.

The following is a listing and brief outline of the tables contained within this database:

1. **t_survey**: Descriptions of the various surveys undertaken and a numeric code to identify each survey.
2. **t_station**: Station details including date, location and depth.
3. **t_sample:** Sample details for a surveyed station.
4. **t_fish_bio:** Details of individual shellfish sampled, including size.
5. **t_test_result:** Results of analysis for disease of individual shellfish specimens.
6. **t_test_method**: Testing methods, including the host species or target pathogen. Generic fields in t_test_result are defined.
7. **t_lab**: Test laboratories used for disease testing of shellfish.
8. **v_vessel**: View containing vessel details for vessels in shellfish, from rdb.t_vessels, including names of vessels with vessel codes.

5 Database Tables

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

5.1 Table 1: t_survey

Comment: Table of surveillance surveys and descriptions.

Column	Type	Null?	Description
survey	smallint	No	Unique sequential number for each survey or surveillance round.
proj_code	character varying(12)		Project code for this survey.
date_s	date		Survey start date.
date_f	date		Survey finish date.
mainspp	character varying(16)		Species codes of host species being surveyed separated by commas.
methods	character varying(20)		List of method codes separated by commas, refer rdb.meth_codes. E.g.: 21 Dredge (bit), 25 Dredge (bag), 70 Handgathering, 71 Divers.
areas	character varying(50)		Codes of area(s) surveyed separated by commas. Refer rdb.area_codes for 4 char area codes or https://marlin.niwa.co.nz/area_codes/
survey_type	character varying(24)		Type of survey, eg: Target surveillance (specific biosecurity sampling programme);

		Passive surveillance (Opportunistic e.g. reports and students); Research (disease research programmes).
staff	text	Names of staff involved in the survey.
reference	text	Bibliographic reference or full citation.
descrptn	text	Descriptive comments about the survey.

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

Check constraints:
    "date_check" CHECK (date_f >= date_s)

Referenced by:
    TABLE "t_station" CONSTRAINT "fk_t_station_ref_t_survey"
     FOREIGN KEY (survey) REFERENCES t_survey(survey)
     ON UPDATE RESTRICT ON DELETE RESTRICT
```

5.2 Table 2: t_station

Comment: Station details for a survey station, including date and location.

Column	Type	Null?	Description
station_key	integer	No.	System generated unique integer to identify the station.
survey	smallint	No	Survey number, refer t_survey.
station_no	integer	No	Numeric station number, unique within each survey.
zone	character varying(12)		MPI biosecurity designation, e.g., Contained, Protected Stewart Island (SIZ), or Unaffected. See Appendix 1, *Bonamia ostreae* controlled area notice https://www.biosecurity.govt.nz/dmsdocument/18259-bonamia-ostreae-controlled-area-notice.
vess_code	integer		3 character vessel code, refer v_vessel.
date_s	date		Date at start of station or sampling event.
time_s	integer		Time (24hr, NZST) at start of station.
stn_code	character varying(16)		Station code, alpha-numeric. Additional identifier for the station.
lat_s	numeric(8,6)		Latitude at start of station in decimal degrees, (6 decimal places).

long_s	numeric(9,6)	Longitude at start of station in decimal degrees East of Greenwich, 0-360 (6 decimal places).
depth_s	integer	Depth (m) at start of the station.
population_type	character varying(8)	Column to document if “Wild” or farmed for population being sampled. Land-based farming such as hatchery recorded as “Farm_L” and grown in the sea “Farm_S”.
farm_no	character varying(16)	Farm consent number.
license_no	character varying(16)	Farm license number.
area_code	character varying(4)	Area code, refer rdb.area_codes, eg FOVE = Foveaux Strait.
site_desc	text	Site location description for this station, as text string.
gear_meth	character(2)	Method code, for method used to gather shellfish, refer rdb.meth_codes or Appendix 1. eg 71 diving.
collector	character varying(4)	The organisation responsible for the sampling at the station, e.g., NIWA, MPI, contractor or farmer.
startp	geometry	Start position of station as GIS point type, derived from lat_s and long_s. Pertains to start position of vessel or dive.

Indexes:

"pk_t_station" PRIMARY KEY, btree (station_key)

```
“ui_t_station” UNIQUE INDEX (survey, station_no)
```

Foreign-key constraints:

"fk_shellfish_t_station_ref_t_vessels" FOREIGN KEY (vess_code) REFERENCES rdb.t_vessels(code) ON UPDATE RESTRICT ON DELETE RESTRICT

"fk_t_station_ref_t_survey" FOREIGN KEY (survey) REFERENCES t_survey(survey) ON UPDATE RESTRICT ON DELETE RESTRICT

Referenced by:

TABLE "t_sample" CONSTRAINT "fk_t_sample_ref_t_station" FOREIGN KEY (station_key) REFERENCES t_station(station_key) ON UPDATE RESTRICT ON DELETE RESTRICT

5.3 Table 3: t_sample

Comment: Sample details from a surveyed station.

Column	Type	Null?	Description
sample_key	integer	No.	System generated unique integer to identify the sample.
survey	smallint	No	Survey number, refer t_survey.
station_no	integer	No	Station number.
species	character(3)	No	3 character species code for the host species being sampled.
target_sample	integer		Target sample size for number of oysters or other shellfish to be collected.
sample_size	integer		Number of shellfish collected or number of pooled samples.
station_key	integer	No	Parent station_key, refer t_station.

Indexes:

"pk_t_sample" PRIMARY KEY, btree (sample_key)

ui_t_sample UNIQUE INDEX (survey, station_no, species)

"fk_t_sample_ref_t_station" FOREIGN KEY (station_key) references t_station(station_key)

Referenced by:

TABLE "t_fish_bio" CONSTRAINT "fk_t_fish_bio_reference_t_sample" FOREIGN KEY (sample_key) REFERENCES t_sample(sample_key)

5.4 Table 4: t_fish_bio

Comment: Details of shellfish specimens, including morphological measurements, typically for individual specimens or may be for pooled samples eg of larvae or spat.

Column	Type	Null?	Description
fish_bio_key	integer	No	System generated unique integer to identify the individual shellfish, or rarely a pooled sample.
survey	smallint	No	Survey number, refer t_survey.
station_no	character varying(8)	No	Station number.
species	character(3)	No	3 character code for the shellfish species.
fish_no	integer	No	Specimen number, unique within each station. Use -1 for (first) pooled sample if applicable.
lgth	integer		Shell length (mm), along the anterior posterior axis.
hght	integer		Shell height (mm), along the dorsal ventral axis.
wdth	integer		Shell width (mm), greatest thickness across both valves.
shell_weight	numeric(4,1)		Weight of shell only (grams).

meat_weight	numeric(5,2)		Drained wet weight of shellfish meat only (grams).
meat_vol	numeric(4,1)		Volume of shellfish meat (ml) as measured by water displacement.
size_code	character varying(1)		1-character code for the size of the shellfish. E.g., for oysters: R=Recruit; P=Pre-recruit; S=Small recruit. Or A = Larvae, B = Spat. See Appendix 1.
brooding	character varying(8)		Is shellfish eg oyster brooding, E.g., No, or colour to determine larval development stage. (White, Grey, Black, Golden for OYS)
gaping	character(1)		If shellfish is gaping, Y = Yes or N = No.
pooled_sample	character(1)		Is the sample pooled, Y = Yes, N = No. I.e., to record if a disease test is from multiple shellfish spat or individuals analysed in 1 test.
pool_size	integer		Number of individuals pooled for this sample and subsequent test, for pooled samples only.
pool_desc	character varying		Description of pooled sample, particularly if eg 20 ml of larval suspension.
sample_key	integer	No	Parent sample key, refer t_sample.

Indexes:

```
   pk_t_fish_bio  PRIMRY KEY, (fish_bio_key)

   ui_t_fish  UNIQUE INDEX  (survey, station_no, species, fish_no)

Check constraints:

    "ckc_size_code_t_fish_bio" CHECK (size_code IS NULL OR (size_code = ANY
     (ARRAY['R'::bpchar, 'P'::bpchar, 'S'::bpchar, 'A'::bpchar,
     'B'::bpchar])))

Foreign-key constraints:

    "fk_t_fish_bio_reference_t_sample" FOREIGN KEY (sample_key) REFERENCES
     t_sample(sample_key)

Referenced by:

    TABLE "t_test_result" CONSTRAINT "fk_t_test_result_ref_t_fish_bio"
     FOREIGN KEY (fish_bio_key) REFERENCES t_fish_bio(fish_bio_key)
```

5.5 Table 5: t_test_result

Comment: Results from a test for disease from individual shellfish specimens. Each test for 1 pathogen will typically provide results for 2 target organisms in this table, a control result and a test result.

Column	Type	Null?	Description
result_key	integer	No	System generated unique integer to identify the test result.
survey	smallint	No	Survey number, refer t_survey.
station_no	integer	No	Station number.
species	character(3)	No	3 character species code for the host species being tested, eg OYS. refer rdb.species_master or https://marlin.niwa.co.nz/species_codes/
fish_no	integer	No	Specimen number, unique within each station, for each individual shellfish from this station.
tissue_type	character varying(12)		Tissue type being tested, eg. Heart, Gill or heart/gill for both combined.
sample_plate	character varying(32)		Character string to identify the sample plate.
well_no	character varying(8)		Well number for PCR plate well location for each individual tested.

analysis_plate	character varying(40)		Character string to identify the analysis plate.
lab	character varying(8)	No	Code for the testing laboratory, e.g., NIWA or AHL. Refer table t_test_lab.
test_date	date		Date the test was made.
test_type	integer	No	Number for the test type (e.g., histology, qPCR, ddPCR, ISH), refer table t_test_method.
test_no	integer	No	Test number, as sequential number for each test for an individual shellfish. E.g., NIWA tests the shellfish as test number 1 to n and then AHL tests again as test number n+1 to last test.
target_organism	character varying(50)	No	Scientific name of the target organism for the test, e.g. Bonamia ostreae or Bonamia exitosa as organism for the test, or Ostrea chilensis as control organism.
result	character varying(12)		Disease test result, e.g. NEG for negative, POS for positive and INC for inconclusive (neither positive nor negative but suspicious results sent to MPI for further testing). PEN for pending, no result provided.
result2	character varying(16)		Column for additional result of the test if required. Non-quantitative. Meaning of this value is documented in t_test_method.result2_desc.

result_quant	numeric		Quantitative (numeric) result of the test. Meaning of this value is documented in t_test_method.result_quant_desc.
result_quant2	numeric		Second quantitative (numeric), result of the test if applicable. Meaning of this value is documented in t_test_method.result_quant2_desc.
test_type_key	integer	No	Unique integer to identify the test type. Refer table t_test_method.
fish_bio_key	integer	No	System generated unique integer to identify the shellfish tested.

Indexes:

"pk_t_test_result" PRIMARY KEY, btree (result_key)

ui_t_test_result UNIQUE INDEX (survey, station_no, species, fish_no, test_no, target_organism)

Foreign-key constraints:

"fk_t_test_result_ref_t_fish_bio" FOREIGN KEY (fish_bio_key) REFERENCES t_fish_bio(fish_bio_key)

"fk_t_test_result_ref_t_lab" FOREIGN KEY (lab) REFERENCES t_lab(lab)

"fk_t_test_result_ref_t_test_meth" FOREIGN KEY (test_type_key) REFERENCES t_test_method(test_type_key)

5.6 Table 6: t_test_method

Comment: Testing methods, including the host species or target pathogen, and the meaning of generic fields in t_test_result.

Column	Type	Null?	Description
test_type_key	integer	No	Unique system generated unique integer to identify the test method record.
test_type	integer	No	Unique number for the type of disease test. Where a method includes a control as well as testing for the target organism this number links these 2 components of the test.
test_name	character varying(12)	No	Short name for the test, eg qPCR_S, qPCR_P, ddPCR.
target_organism	character varying(50)	No	Scientific name of the target organism for the test, e.g. Bonamia ostreae or Bonamia exitosa as organism for the test, or Ostrea chilensis as control organism.
result_desc	character varying		Description of the meaning of column result in t_test_result, typically POS, NEG or Inconclusive.
result2_desc	character varying		Description of the meaning of column result2 in t_test_result, where applicable. E.g., PEAK.

result_quant_desc	character varying	Description of meaning of numeric result_quant value in t_test_result where applicable. E.g., cq values from qPCR test.
result_quant2_desc	character varying	Description of meaning of numeric result_quant2 value in t_test_result where applicable. E.g., Melt temperature.
description	text	Description of disease testing method. Should include detail of test type such as qPCR probe, qPCR cyber, ddPCR etc and document aspects including Fluor and control used such as B.actin.

Indexes:

"pk_t_test_method " PRIMARY KEY, btree (test_type_key)

“ui_test_method” UNIQUE index on (test_type, target_organism)

5.7 Table 7: t_lab

Comment: Test laboratories used for disease testing of shellfish.

Column	Type	Null?	Description
lab	character varying (8)	No	Code or acronym for the testing laboratory, eg NIWA, AHL.
lab_address	character varying		Text field for physical address of the test laboratory.
comments	character varying		Comments with additional info about the disease testing laboratory, if useful.

Indexes:

"pk_t_lab" PRIMARY KEY, btree (lab)

Referenced by:

TABLE "t_test_result" CONSTRAINT "fk_t_test_result_ref_t_lab" FOREIGN KEY (lab) REFERENCES t_lab(lab)

5.8 View: v_vessel

Comment: View containing vessel details for vessels in shellfish, from rdb.t_vessels, including names of vessels with vessel codes.

Column	Type	Null?	Description
vess_code	character(3)	No	3-character lowercase vessel name code. E.g. ika for Ikatere
vess_name	character varying(50)	No	Vessel name.
vess_no	numeric(6,0)		Vessel registration number for commercial fishing vessels.
comments	character varying		

View definition:

```
SELECT t_vessels.code AS vess_code, t_vessels.name AS vess_name,
       t_vessels.vess_no, t_vessels.comments

   FROM rdb.t_vessels

   WHERE (t_vessels.code IN

   ( SELECT DISTINCT t_station.vess_code FROM t_station));
```

6 shellfish business rules

6.1 Introduction to business rules

The following are a list of business rules applying to the **shellfish** 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 shellfish diseases 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

Survey details (t_survey)

survey	Survey number must be present and be a unique integer.
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.
mainspp	Each species code within the main species list should be a valid species code as listed in the *curr_spp* table of the rdb schema of the fish database. A comma should separate each species code.
methods	Each method code within the methods list should be a valid method code as listed in the *meth_codes* table of the rdb schema. A comma should separate each method code
areas	Each area code within the areas list should be a valid area code as listed in the *area_codes* table of the rdb schema in the fish database. A comma must separate each area code.

Vessel details (t_vessel)

vess_code	Vessel code must have a value and should be a 3 character lower case code.

vess_no	Vessel number if applicable must be an integer and should be a valid MAF/MPI FNZ (Fishserve) vessel registration number.

Station details (t_station)

station_key	Station key must have a value and be a unique integer.
survey	Survey number must be present and be a unique integer and must exist in the t_survey table.
station_no	Must be a unique number within a single survey.
vess_code	Must be a 3 letter lower case code, must exist in the t_vessel table and should exist in the rdb.t_vessel table.
date_s	The date at the start of a station must be a legitimate date. Multiple column checks on start date: The date should fall within the range of the survey start and finish dates.
time_s	Start time of the station must be a valid 24-hour time and fall within the range of 0 – 2359 hours.
lat_s	Should be a valid latitude in decimal degrees format, including negative for latitudes South of the equator
long_s	Should be a valid longitude in decimal degrees format, 0 to 360 degrees east of Greenwich.

population_type	Population type should be Farmed or Wild or an alternative suitable documented value.
area code	Area code must be a valid research area code as listed in the area_codes table in the rdb database.
gear_meth	Gear method code must be a valid code as listed in the *meth_codes* table in the **rdb** database.
startp	The start point for the station must be a valid geometry, in the current GIS.

Sample details (t_sample)

sample_key	Sample key must have a value and be a unique integer.
survey	Survey number must be present and be a unique integer, and must exist in the t_survey table.
station_no	Must be a unique number within a single survey.
species	The species code must be a 3 character uppercase code, and be a valid species code as listed in the *curr_spp* table of the **rdb** schema of the fish database.
target sample	The target sample size if recorded must be numeric, i.e., an integer.
station_key	Station key must have a value and be a unique integer, and must be present in table t_station.

Individual shellfish details (t_fish_bio)

fish_bio_key	The fish biological key must have a value and be a unique integer.
survey	Survey number must be present and be a unique integer and must exist in the t_survey table.

station_no	Must be a unique number within a single survey.
species	The species code must be a 3 character uppercase code, and be a valid species code as listed in the *curr_spp* table of the **rdb** schema of the fish database.
fish_no	The specimen or fish number must be an integer and be unique within each station.
lgth	The length of the shellfish must be an integer and should be a reasonable value for the species measured.
hgth	The height of the shellfish must be an integer and should be a reasonable value for the species measured.
size_code	The size code for the shellfish should be as documented in the database documentation: R=Recruit; P=Pre-recruit; S=Small recruit. Or A = larvae, B = spat. Or documented within the data supplied, eg t_survey.
pooled_sample	The value to indicate if the sample is a pooled sample should be ‘Y’ or ‘N’.
sample_key	Sample key must have a value and be a unique integer and must be present in table t_sample.

Test results (t_test_result)

result_key	The test result key must have a value and be a unique integer.
survey	Survey number must be present and be a unique integer and must exist in the t_survey table.

station_no	Must have a value and be a unique number within a single survey.
species	The species code must have a value and be a 3 character uppercase code, and be a valid species code as listed in the *curr_spp* table of the **rdb** schema of the fish database.
fish_no	The specimen or fish number must be present and be an integer and be unique within each station.

lab	The laboratory code must have a value and must be a valid code as listed in table t_test_lab.
test_date	The date of the test must be a valid date and should be greater than the station (collection) date.

test_type	Test type must be an integer value and should be a valid test type in table t_test_method.
test_no	Test number must have a value and should be a positive integer.
target_organism	The scientific name of the target organism should be a valid scientific name.
result	The value of the result must have a value and should have a value of either 'NEG' or 'POS'.
test_type_key	The test type key must have a value and this must be present in table t_test_method.
fish_bio_key	The fish biological key must have a value and be a unique integer, and be present in table t_fish_bio.

Test methods (t_test_method)

test_type_key	The test type key must have a value, and must be unique.
test_type	Test type must have a value and must be an integer and should be a positive number.
target_organism	The scientific name of the target organism should be a valid scientific name.

Laboratory (t_lab)

lab	Laboratory code or acronym must be no longer than 8 characters and should be in uppercase.

7 Acknowledgements

The authors would like to thank Owen Anderson for his contribution to the design of the database and review of this document.

Appendix 1 Reference code tables

Method codes from rdb.meth_codes.

Code	Description
21	Dredge (bit)
25	Dredge (bag)
70	Handgathering
71	Divers

Note these are method codes initially used in **shellfish***,*

Refer to rdb.meth_codes.for the full set of research method codes.

Types of survey as used in t_survey survey_type

Targeted surveillance: A structured biosecurity sampling programme to collect information on a specific disease or pathogen.

Passive surveillance: Collection, collation, analysis of information that has been collected opportunistically, by reporting, from a centralised system or from an information system designed for another purpose.

Research: A systematic investigation looking to elucidate and generate new information and conclusions

Zone as used in t_station is defined as: MPI biosecurity designation, e.g., Contained, Protected Stewart Island (SIZ), or Unaffected.

See *Bonamia ostreae* controlled area notice

https://www.biosecurity.govt.nz/dmsdocument/18259-bonamia-ostreae-controlled-area-notice size_code in table t_fish_bio is defined as

1-character code for the size of the shellfish.

E.g., for oysters: R=recruit; P=pre-recruit; S=small recruit, or A = Larvae, B = Spat

Individual oysters were allocated to size groups based on their ability to pass through three standard rings with internal diameters of 65 mm, 58 mm and 50 mm:

C = Commercial sized oysters, ≥ 65 mm

R = Recruit-sized (minimum legal size) oysters, ≥ 58 mm to 64 mm

P = Pre-recruit oysters, ≥ 50 mm to 57 mm

S =Small oysters, 49 mm down to 10 mm in diameter.

Additionally,

A = larvae, oyster larvae in the water column and before settlement

B = Spat, sizes range from recently settler larvae ~1 mm to 10 mm (0+ age cohort)

Appendix 2 Example sql

Example SQL to join up test results with control result:

```
select a.survey, a.station_no, a.species, a.fish_no, a.tissue_type as tissue, a.lab, a.test_date,
a.test_type, a.result, a.result2, a.result_quant as result_no, a.result_quant2 as result_no2,
      b.result as control_result, b.result2 as control_result2
from t_test_result a join t_test_result b
on a.target_organism = 'Bonamia ostreae' and b.target_organism = 'Ostrea chilensis'
  and a.test_no = b.test_no
  and a.fish_no = b.fish_no
  and a.station_no = b.station_no
  and a.survey = b.survey
order by  a.survey, a.station_no, a.species, a.fish_no,  a.test_type ;
```