Database documentation: rocklob

K. A. Mackay

NIWA Fisheries Data Management
Database Document Series

Updated Dec-2015

Contents

List of Figures

Revision History

Version	Change	Date	Person responsible
1.0	Initial version as NIWA Internal Report No. 26	1998	Kevin Mackay
1.1	Added business rules ? Released as NIWA Internal Report No. 70	2000	Kevin Mackay
1.2	Explained station_no = 0 for t_evpsumm	15Feb13	D Fisher with K Mackay
2.0	Postgres version	Dec 2015	D Fisher, F Wei

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 is a brief introduction to the rock lobster phyllosoma and puerulus database **rocklob**, 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 **rocklob**.

This database has been implemented as a schema within the Postgres database called **fish**.

This document is intended as a guide for users and administrators of the **rocklob** database, and supersedes Mackay (1998) [1] .

2 Abundance of early life history stages of the rock lobster

Red rock lobsters (*Jasus edwardsii*) support one of New Zealand's most valuable fisheries. Understanding larval recruitment processes greatly assist management of this fishery.

Rock lobsters spend several months as phyllosoma larvae tens to hundreds of kilometres offshore. Phyllosomas can disperse large distances: advanced phyllosomas were taken to the seaward extent of east coast trawl transects during trawl surveys and are widespread in the south Tasman Sea. At least recently, advanced (mid- and late-stage) phyllosomas have been much more abundant off the east coast of the North Island south of East Cape than off the east coast of the South Island (Booth & Forman 1995). This pattern appears to be determined by factors that include levels of local rock lobster larval production and the oceanography. Surveys have been carried out since the 1970s using mainly fine-meshed mid-water trawls, but also bottom trawls and bongo nets. Survey designs have include both transects and strata area.

Rock lobsters return to the shore as pueruli. The puerulus stage is the settling stage: it resembles the juvenile in shape and is 9-13mm in carapace length, but is transparent. Puerulus settlement happens when pelagic pueruli cease extensive forward swimming and take up residence on the substrate or in a crevice collector. Some older pueruli and young juveniles, however, move into collectors after first settling elsewhere.

Key sites to follow levels of settlement on crevice collectors (see Booth *et al*. 1991 for collector design) have been set up in the main rock lobster fishing coasts of New Zealand. Collectors are set in groups of 3-6, with a minimum spacing of 2-3m between individual collectors. At each key site there is a core group; additional groups of collectors are set in both directions along the coast, as conditions allow, 0.1-25km from the core collectors. At most sites, collectors are checked monthly and all lobsters removed. These collectors provide a combined index of:

a) the number of pelagic pueruli in the water column which are settling;
b) the result of post-settlement migration, the net number of older animals (older pueruli, and less often, young juveniles) moving onto the collector after having lived on the surrounding sea floor, and animals of similar age moving from the collector to the surrounding sea floor.

The index of annual settlement is the average catch per collector of pueruli, plus post-pueruli up to and including 14.5mm carapace length combined, of the core collectors over the main settlement season. The main settlement season varies between 6 and 10 months according to site, so values of annual index are not always directly comparable between sites.

Juvenile abundance has been followed at three locations (Gisborne, Wellington, and Stewart Island, the last two having several sites within the location) to check whether peaks and troughs in settlement are mirrored in the abundance of 1-3 year olds.

Knowing the abundance of early life history stages (phyllosomas, pueruli, and young juveniles) may lead researchers to the factors that drive fishery recruitment. It may be possible to relate changes in levels of settlement to changes in breeding stock abundance, abundance of advanced larvae, and to changes in the ocean climate. Information on year-to-year settlement levels may be used to predict trends in recruitment, provide early warning of over fishing, and indicate to what extent recruitment varies from year to year. A discussion of the abundance of early life history stages of lobsters and the implications to fishery management are detailed in Booth *et al*. 1998.

3 Data structures

3.1 Table relationships

This database contains several tables. The ERD for **rocklob** (Figure 1) shows the physical data model 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.

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

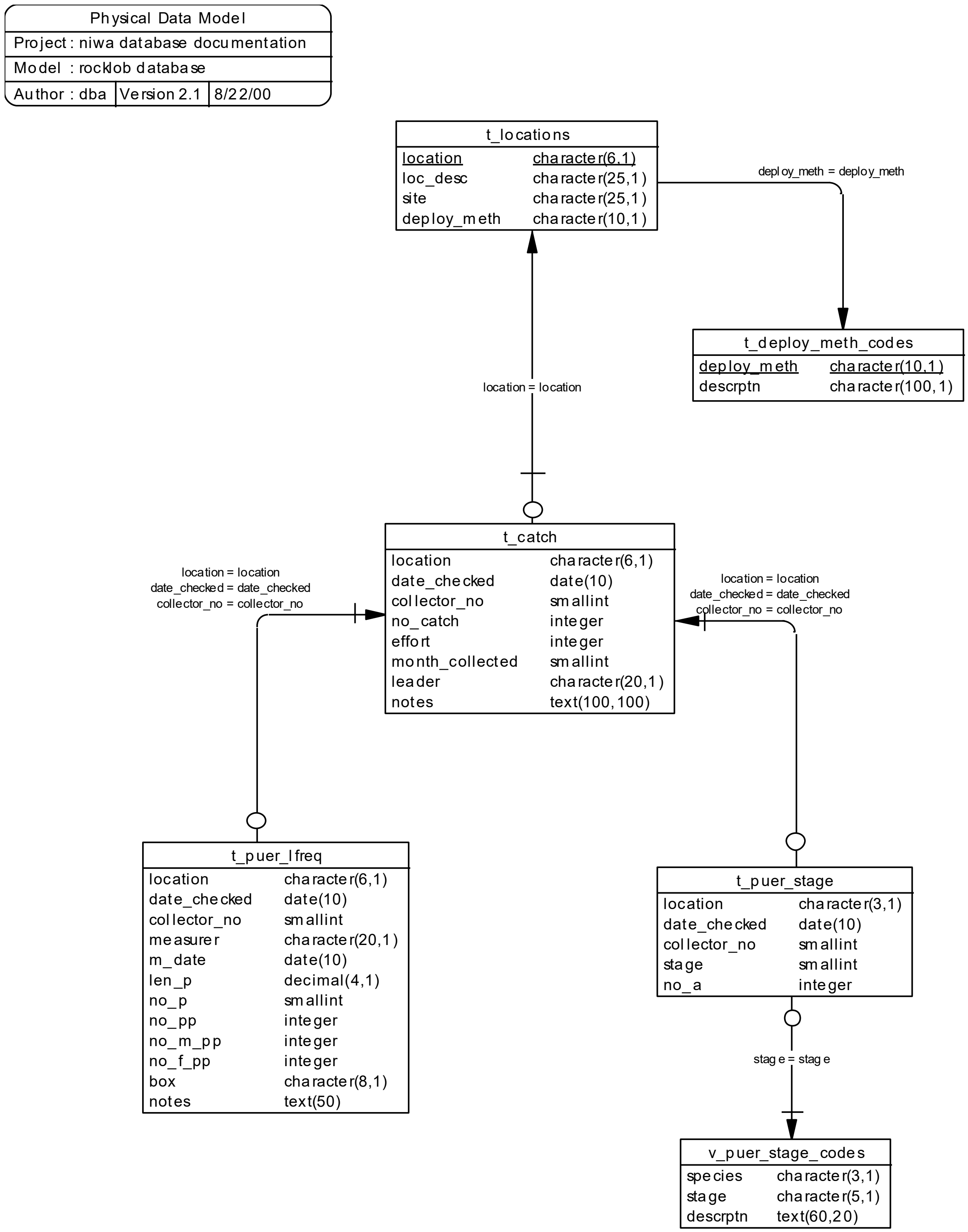

Figure 1: Entity Relationship Diagram (ERD) showing the relationships between the tables that hold data from puerulus settlement collectors and juvenile abundance surveys.

All 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. Most of the tables in the **rocklob** database also contain special attributes, called foreign keys[3].

Section 5 shows a listing of all the **rocklob** 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 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.

The primary key prevents records with duplicate key values from being inserted into the table; e.g., a new survey being inserted with an existing survey number, and hence ensures that every record can be uniquely identified.

The **rocklob** 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 **rocklob** are of the type *one-to-many*[4]. This is shown in the ERD by connecting a single line (indicating 'many') from the child table to the parent table with an arrowhead (indicating 'one') pointing to the parent (e.g., *t_station* to *t_trip* in Figure 2). 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 stations in *t_station*, but any one station can only be a part of one trip.

Every relationship has a mandatory or optional aspect to it. That is, if a relationship is mandatory, then it has to occur and least once, while an optional relationship might not occur at all. For example, in Figure 2, consider that relationship between the table *t_station* and it's child table *t_phy_stage*. The symbol "O" across the relationship line by the child *t_phy_stage* means that a station record can have zero or many phyllosoma catch records, while the bar across the relationship line by the parent *t_station* means that for every phyllosoma catch record there must be a matching station record.

---

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

[3] A foreign key is any attribute, or a combination of attributes, in a table that is a primary key of another table. Tables are linked together through foreign keys.

[4] A one-to-many relationship is where one record (the *parent*) in a table relates to one or many records (the *child*) in another table; e.g., one survey in *t_trip* can have many stations in *t_station* but any one station can only come from one survey.

These relationships are enforced in the database by the use of foreign key constraints[5]. Foreign key constraints do not allow *orphans* to exist in any table; i.e., where a child record exists without a related parent record. This may happen when: a parent record is deleted; the parent record is altered so the relationship is lost; or a child record is entered without a parent record. All constraints in **rocklob** prevent the latter from occurring. Foreign key constraints are shown in the table listings by the following format:

```
Referential:      error message (attribute[, attribute]) INSERT
                  parent table (attribute[, attribute])
```

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

```
Foreign-key constraints:
    "fk_t_station_t_trip" FOREIGN KEY (trip_code) REFERENCES
     t_trip(trip_code)
```

This means that the value of the attribute *trip_code* 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. Most tables are indexed. That is, attributes that are most likely to be used for searching, such as *trip_code*, have like values linked together to optimise search times.

Such indices are shown in the table listings (Section 5) by the following syntax:

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

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

3.2 Database design

3.2.1 Puerulus collector and juvenile abundance monitoring

Pueruli and young juveniles of *J. edwardsii* occur most abundantly in shallow waters. Around New Zealand, pueruli occur intertidally in crevices, holes, and indentations under boulders. Artificial crevice collectors, deployed by various methods (shore, sea floor, and surface/midwater), are used in locations around the country as a means of measuring abundance of settlement. The abundance of 1-3 year old juveniles are surveyed in several locations around New Zealand as a check on the usefulness of collectors. Locations of crevice collectors and juvenile abundance surveys are stored in the table *t_locations* (Table 1). Each location is identified by a unique location code stored in the attribute *location*. Juvenile abundance monitoring is distinguished from puerulus collectors by the suffix “J” in *location*. The attribute *deploy_meth* is a code used to identify the method by which monitoring was done; e.g., as shore- or boat-deployed crevice collectors; or scuba diver abundance counts. Full descriptions of the *deploy_meth* code are listed in the table *t_deploy_meth_codes* (Table 11).

[5] Also known as referential constraints or integrity checks.

Periodically (usually monthly) these collectors and 'natural' settlement (such as intertidal beaches, natural crevices and indentations in boulders) are checked for pueruli settlement and juvenile abundance is monitored. The details of such checks, including numbers of pueruli caught and juveniles seen, are stored the table *t_catch* (Table 2). Each check is uniquely identified by the attributes: *location*, *date_checked*; and *collector_no*, a unique sequential number given to each collector at a location. The total number of all *J. edwardsii* caught for each collector during the check or seen during a dive are in stored in *no_catch.* At the time of the check, lobsters are not staged so there is no distinction made been pueruli and young juveniles until each lobster is measured and staged at a later date. Effort, as measured by total dive times in minutes, for juvenile abundance surveys is stored in the attribute *effort*. The month for which the collector was checked for is stored as an integer from 1 to 12 in the attribute *month_collected*. Often collectors are checked early the next month for the previous months pueruli settlement. Similarly, the year for which the collector was checked for is stored in the attribute *year_collected*. There are specific occasions where the count of pueruli settlement for December is actually made in January of the next year.

Historically, all lobsters caught by the collectors at certain locations were measured by carapace length. These length frequency data are stored in the table *t_puer_lfreq* (Table 3). Each record in this table stores the number of lobsters per millimetre carapace length class. Lobsters were also recorded as being either pueruli or post-pueruli. So, for each millimetre length class, *t_puer_lfreq* records the numbers of pueruli (*no_p*) and the numbers of juveniles (post-pueruli) (*no_pp*). The smaller length classes having only pueruli, the larger lengths having only post-pueruli, and in some uncommon cases, a length class may have counts for both.

Counts of juveniles seen during abundance monitoring are also stored in *t_puer_lfreq*.

Currently, all pueruli and post-pueruli lobsters are staged based on their life history. These staging data are stored in the table *t_puer_stage* (Table 4). Staging only takes place on lobsters up to 14.5mm carapace length as it is generally accepted that this is the maximum size that lobsters can grow in one month (the time between collector checks). Lobsters below this size are staged to a four-point scale. The number of all lobsters over this size is also recorded as a measure of post-settlement migration.

Descriptions of the pueruli stages are listed in *v_puer_stage_codes*, which is a view on the table *t_stage_codes* (Table 12).

3.2.2 Phyllosoma trawl surveys

The phyllosoma trawl survey data model is an adaptation of the trawl survey data model (Mackay 1998).

Several trawl surveys have been carried out since the 1970s to define the extent of occurrence and to index abundance of advanced stage phyllosoma. Details of such trips are stored in the table *t_trip* (Table 5), including start/finish dates, areas surveyed, and parameters of trawl gear used.

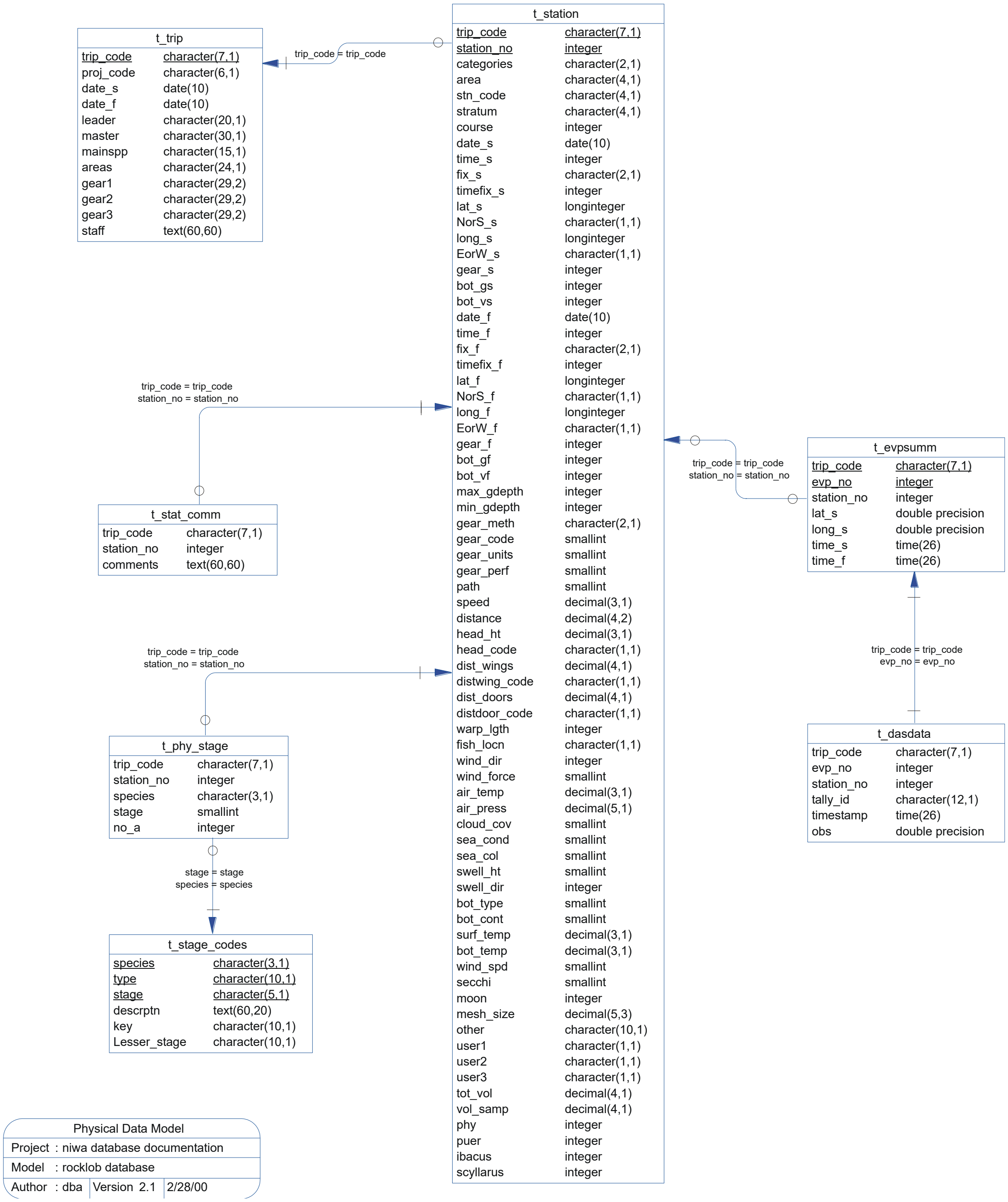

Figure 2: ERD showing the relationships between the tables for the phyllosoma trawl surveys.

Surveys may or may not have included strata as part of the methodology, but where they were involved no details of strata are recorded in **rocklob**. Unlike other stratified trawl surveys, phyllosoma surveys do not involve random stations nor are catches scaled up to any area.

These surveys all involve transects, with a number of stations (where trawl gear was deployed) occurring along the transect path. Station details are stored in the table *t_station* (Table 6). Catch details from each station are included within t_station; e.g., total volume caught, and volume sampled. Only 4 items within each catch are recorded so each has it's own attribute within t_station for catch numbers. These attributes are: *phy* for *Jasus* sp. phyllosomas; *puer* for pueruli; *ibacus* for *Ibacus* sp. phyllosoma; and *scyllarus* for *Scyllarus* sp. phyllosoma.

Comments on stations are stored in a separate table *t_stat_comm* (Table 7). Having the comments in their own tables means that each station can have none or as many comments as a user wishes.

Measurements taken from individual phyllosoma are stored in *t_phy_stage* (Table 8). Currently, only life history stages are recorded for individual specimens, although, this table can be expanded to include other measurements if required.

Two views of this table (*v_scyllarus* and *v_jasus*) exist to provide phyllosoma stage data for *Scyllarus* sp. and *Jasus* sp. respectively. A view is a table that does not have any existence in its own right but is instead a snapshot of data derived from one or more base tables.

One of the aims of phyllosoma trawl surveys is to compare abundance data with changes in ocean climate. To this end, on RV Tangaroa surveys, physical oceanographic data such as surface temperature, salinity, and sea currents are logged real-time from the vessel's sensors by the Data Acquisition System (DAS). DAS controls data logging automatically by monitoring the trawl gear warp length starting logging when the trawl gear is let out, stopping when the trawl gear is hauled. Data are recorded into discrete envelopes of time and each envelope is roughly comparable with a station. Each envelope is given a sequential envelope number, with summary information such as matching station number and start/finish time stored in *t_evpsumm* (Table 9). Instances of unmatched stations are assigned a station number of 0. Envelopes were sometimes created when no gear was deployed.

Actual data recorded by the DAS for an envelope are stored in *t_dasdata* (Table 10) with each record having a timestamp (*timestamp*) to say when it was recorded, an identifier (*tally_id*) to say which sensor is being measured, and an observed value (*obs*) from the sensor.

Explanations for all phyllosoma and puerulus developments stages are stored in the table *t_stage_codes* (Table 12). From this table, there are two views, which show explanations of development stage codes for puerulus (*v_puer_stage_codes*) and phyllosoma (*v_phy_stage_codes*) exclusively.

4 Table summaries

This database is arranged as a set of eleven main tables, two views of species-specific data, and a further two views for codes.

The following is a listing and brief outline of the tables contained within **rocklob** pertaining to puerulus settlement and juvenile abundances:

1. **t_locations :** contains the full name of each location code, site within a location and the method of puerulus collector deployment or juvenile survey.
2. **t_catch** : contains details for each puerulus collector check or juvenile abundance survey, including location code, date, and numbers of pueruli caught.
3. **t_puer_lfreq** : contains carapace length data for pueruli and juveniles from collectors or abundance surveys during a check.
4. **t_puer_stage** : contains life history stage data for individual pueruli caught by collectors during a check.

The following is a listing and brief outline of the tables contained within **rocklob** pertaining to phyllosoma trawl surveys:

5. **t_trip** : contains profile information on all trips.
6. **t_station** : contains data on location, gear used and environment at each station within a trip.
7. **t_stat_comm** : contains comments for a station in a trip.
8. **t_phy_stage** : contains codes for levels of life history development for individual phyllosoma.
   (a) **v_scyllarus** : a view of the *t_phy_stage* table containing all scyllarus (SHL) phyllosoma life history stage data.
   (b) **v_jasus** : a view of the *t_phy_stage* table containing all Jasus sp. phyllosoma (PHY) life history stage data.
9. **t_evpsumm :** contains start and finish times and positions, and station numbers contained in, envelopes of DAS data.
10. **t_dasdata :** contains filtered data from the vessel's sensors collected in DAS envelopes

The following are the common code tables used by both puerulus/juvenile and phyllosoma surveys:

11. **t_deploy_meth_codes :** contains codes and their descriptions of the methods used to deploy puerulus and juvenile collectors.
12. **t_stage_codes :** contains crayfish development stage codes and their descriptions, from phyllosoma larvae to juveniles.
    (a) **v_puer_stage_codes :** contains puerulus development stage codes and their descriptions.
    (b) **v_phy_stage_codes :** contains phyllosoma development stage codes and their descriptions.

5 rocklob tables

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

See Appendix 1 for attributes that have comments referring to the Trawl Instructions (unpub. NIWA report).

5.1 Table 1: t_locations

```
Comment: Lists the location used for the deployment of pueruli
         collectors and juvenile surveys, the location code, and the
         data collection method.
```

Column	Type	Null?	Description
location	character varying(6)	No	6-char code for the location of the collector. Format (1) = "AAAnnn" where "AAA" = location abbreviation and "nnn" = site number. Format (2) = "AAAnnJ" where "AAA" = location abbreviation and "nnn" = site number and "J" = juvenile abundance survey.
loc_desc	character varying(25)		Location name or description.
site	character varying(25)		Site name within a location.
deploy_meth	character varying(10)		Method of collector deployment used at the location. Refer t_deploy_meth_codes. e.g. BOAT, DIVER, SHORE, SUSPENDED.

```
Indexes:
   "pk_t_locations" PRIMARY KEY, btree ("location")

Foreign-key constraints:
   "fk_t_locations_t_deploy_meth_codes" FOREIGN KEY (deploy_meth)
    REFERENCES rocklob.t_deploy_meth_codes(deploy_meth)
```

5.2 Table 2: t_catch

Comment: Shows number of pueruli caught on each collector at each check or juveniles counted at each abundance survey, with comments.

Column	Type	Null?	Description
location	character varying(6)	No	6-char code for the location of the collector.
date_checked	date	No	Date the collector was checked.
collector_no	smallint	No	Each collector has been given a unique number.
no_catch	integer		Number of pueruli and/or juveniles caught/counted.
effort	integer		Effort (time in minutes) taken for the catch.
year_collected	integer		Year that the catch data represents.
month_collected	smallint		Month (1-12) that the catch data represents.
leader	character varying(20)		Name of trip leader.
notes	text		Comment for each collector.

```
Indexes:
   "pk_t_catch" PRIMARY KEY, btree
    ("location", date_checked, collector_no)
Check constraints:
   "t_catch_month_collected_check" CHECK (month_collected >= 1 AND
                                          month_collected <= 12)
   "t_catch_year_collected_check" CHECK (year_collected >= 1979)

Foreign-key constraints:
   "fk_t_catch_t_location" FOREIGN KEY ("location")
    REFERENCES rocklob.t_locations("location")
```

5.3 Table 3: t_puer_lfreq

Comment: Table of pueruli and juvenile length frequency data.

Column	Type	Null?	Description
location	character varying(6)	No	6-char code for the location of the collector.
date_checked	date	No	Date the collector was checked.
collector_no	smallint	No	Each collector has been given a unique number.
measurer	character varying(20)		Person who recorded the length of the animals.
m_date	date		Measurement date.
len_p	numeric(4,1)		Carapace length of the peurulus.
no_p	smallint		Number of pueruli.
no_pp	integer		Number of juveniles (post- puerulus)
no_m_pp	integer		Number of male juveniles (post-puerulus).
no_f_pp	integer		Number of female juveniles (post-puerulus).
box	character varying(8)		Box and container number where the animals are kept.
notes	text		

```
Foreign-key constraints:
   "fk_t_puer_lfreq_t_catch" FOREIGN KEY ("location", date_checked,
                                collector_no)
    REFERENCES rocklob.t_catch("location", date_checked,
                                collector_no)
```

5.4 Table 4: t_puer_stage

Comment: Table of pueruli life history stage data.

Column	Type	Null?	Description
location	character varying(6)	No	6-char code for the location of the collector.
date_checked	date	No	Date the collector was checked.
collector_no	smallint	No	Each collector has been given a unique number.
stage	character varying(5)		Code for life history stage. Refer v_puer_stage_codes (stage).
no_a	integer		Number of animals at this stage.

```
Foreign-key constraints:
   "fk_t_puer_stage_t_catch" FOREIGN KEY ("location", date_checked,
                                          collector_no)
    REFERENCES rocklob.t_catch("location", date_checked,
                                          collector_no)
```

5.5 Table 5: t_trip

Comment: Profile information on all trips held in this database.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code - 3 char vessel name, 2 digit year and 2 digit trip number.
proj_code	character varying(6)	No	Project or programme code for this trip.
date_s	date		Start date for the trip.
date_f	date		Finish date for the trip.
leader	character varying(20)		Name of trip leader.
master	character varying(30)		Name of vessel master(s).
areas	character varying(24)		Codes of area(s) surveyed separated by commas (,).
mainspp	character varying(15)		Target species code(s) separated by commas.
gear1	character varying(29)		gear_meth code, codend, liner & cover mesh sizes (mm), ground rope, sweep & bridle lengths (m) separated by commas for 1st gear code used.
gear2	character varying(29)		gear_meth code, codend, liner & cover mesh sizes (mm), ground rope, sweep & bridle lengths (m) separated by commas for 2nd gear code used.
gear3	character varying(29)		gear_meth code, codend, liner & cover mesh sizes (mm), ground rope, sweep & bridle lengths (m) separated by commas for 3rd gear code used.
staff	text		Name(s) of all staff on the trip.

Indexes:
"pk_t_trip" PRIMARY KEY, btree (trip_code)

```
Check constraints:
   "t_trip_gear1_check" CHECK (gear1::text ~ '[0-9,. ]*'::text)
   "t_trip_gear2_check" CHECK (gear2::text ~ '[0-9,. ]*'::text)
   "t_trip_gear3_check" CHECK (gear3::text ~ '[0-9,. ]*'::text)
   "t_trip_mainspp_check" CHECK (mainspp::text ~ '[A-Z,]*'::text)
   "t_trip_proj_code_check" CHECK (proj_code::text ~ '[A-Z][A-Z][A-
                                          Z][A-Z][0-9][0-9]'::text)
   "t_trip_trip_code_check" CHECK (trip_code::text ~ '[a-z0-9][a-z0-
                                          9][a-z0-9][6-9][0-9][0-
                                          9][0-9]'::text)
```

5.6 Table 6: t_station

Comment: Data on location, gear used and environment at each station on a trip.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as defined in the trip table.
station_no	integer	No	Station number - unique within a trip.
categories	character varying(2)		2 separate user-defined categories; definitions should be in trip comments.
area	character varying(4)		Code describing area, refer to rdb.area_codes.
stn_code	character varying(4)		Code for a permanent station occupied repeatedly.
stratum	character varying(4)		Stratum number if trip is a stratified survey, else a transect code.
course	integer		Course of vessel during the shot (course-made-good).
date_s	date		Starting date of the shot.
time_s	integer		Starting time (24hr,NZST) of the shot (hhmm format).
fix_s	character varying(2)		Method of fixing position at start of tow, refer rdb.t_fix_meth_codes.
timefix_s	integer		Time (in minutes) elapsed since last position fix at the start of tow.
lat_s	integer		Latitude of vessel at start of tow (ddmmmm format, d=deg., m=min. to 2 implied dec. pl.).

nors_s	character varying(1)	Tow start position hemisphere.
long_s	integer	Longitude of vessel at start of tow (ddmmmmm format - d=deg., m=min. to 2 implied dec. pl.).
eorw_s	character varying(1)	Tow start position meridian.
gear_s	integer	Depth of lowest part of gear (groundrope) at start of the tow (m).
bot_gs	integer	Depth of sea bottom at gear position at start of the tow (m).
bot_vs	integer	Depth of sea bottom at vessel position at start of the tow (m).
date_f	date	Finishing date of the shot.
time_f	integer	Finishing time (24hr,NZST) of the shot (hhmm format).
fix_f	character varying(2)	Method of fixing position at end of tow, refer rdb:t_fix_meth_codes.
timefix_f	integer	Time (in minutes) elapsed since last position fix at end of the tow.
lat_f	integer	Latitude of vessel at end of tow (ddmmmm format, d=deg., m=min. to 2 implied dec. pl.).
nors_f	character varying(1)	Tow finish position hemisphere.
long_f	integer	Longitude of vessel at end of tow (ddmmmmm format, d=deg., m=min. to 2 implied dec. pl.).
eorw_f	character varying(1)	Tow finish position meridian.

gear_f	integer	Depth of lowest part of gear (groundrope) at end of the tow (m).
bot_gf	integer	Depth of sea bottom at gear position at end of tow (m).
bot_vf	integer	Depth of sea bottom at vessel position at end of tow (m).
max_gdepth	integer	Maximum depth of lowest part of gear (groundrope) during the tow (m).
min_gdepth	integer	Minimum depth of lowest part of gear (groundrope) during the tow (m).
gear_meth	character varying(2)	Gear method code, descriptions in rdb.meth_codes.
gear_code	smallint	Code for set of gear used, details in trip record.
gear_units	smallint	Number of units of gear used in the tow.
gear_perf	smallint	Code for performance of gear during the tow, refer to Appendix 1 of the database documentation.
path	smallint	Code describing configuration of path of shot, refer to Appendix 1 of the database documentation.
speed	numeric(3,1)	Average speed through water during shot (knots).
distance	numeric(4,2)	Distance of gear over bottom (nautical miles).
head_ht	numeric(3,1)	Average headline height (m).

head_code	character varying(1)	Code showing how headline height was determined, refer rdb.t_headline_codes.
dist_wings	numeric(4,1)	Average distance between wings (m).
distwing_code	character varying(1)	Code to indicate how distance between the wings was determined for this tow, refer rdb.t_wing_dist_codes.
dist_doors	numeric(4,1)	Average distance between doors of gear (m).
distdoor_code	character varying(1)	Code to indicate how the distance between the doors was determined for this tow, refer rdb.t_door_dist_codes.
warp_lgth	integer	Length of warp during the tow (m).
fish_locn	character varying(1)	Code to indicate the location of the fish at the net mouth during the shot as observed on the net sonde, refer rdb.t_fish_obs_codes.
wind_dir	integer	Wind direction (degrees true), 999=No wind.
wind_force	smallint	Wind force on Beaufort scale.
air_temp	numeric(3,1)	Air temperature (degrees C).
air_press	numeric(5,1)	Air pressure (millibars).
cloud_cov	smallint	Code describing cloud cover during tow in eighths of sky covered, 0=clear sky.
sea_cond	smallint	Codes describing condition of the sea, refer to Appendix 1 of the database documentation.

sea_col	smallint	Code describing colour of sea, refer to Appendix 1 of the database documentation.
swell_ht	smallint	Code describing height of swell, refer to Appendix 1 of the database documentation.
swell_dir	integer	Direction of the swell (degrees true).
bot_type	smallint	Code describing sea bottom type, refer to Appendix 1 of the database documentation.
bot_cont	smallint	Code describing sea bottom contour, refer to Appendix 1 of the database documentation.
surf_temp	numeric(3,1)	Surface temperature (degrees C).
bot_temp	numeric(3,1)	Temperature at bottom (degrees C).
wind_spd	smallint	Wind speed from anemometer (m/s) (1knot=0.51m/s).
secchi	smallint	Depth at which Secchi disc becomes invisible (m).
moon	integer	Quarters of the moon phase.
mesh_size	numeric(5,3)	Mesh size (cm) of the gear.
other	character varying(10)	Any other details, should be fully commented.
user1	character varying(1)	User-defined field 1. Field should be defined in t_stat_comm.
user2	character varying(1)	User-defined field 2. Field should be defined in t_stat_comm.

user3	character varying(1)	User-defined field 3. Field should be defined in t_stat_comm.
tot_vol	numeric(4,1)	Total volume (litres) of material caught during tow.
vol_samp	numeric(4,1)	Volume (litres) of material sampled.
phy	integer	Number of phyllosomas caught.
puer	integer	Number of pueruli caught.
ibacus	integer	Number of ibacus sp. caught.
scyllarus	integer	Number of scyllarus sp. caught.
dlat_s	numeric(7,5)	Latitude of vessel at start of the station in decimal degree.
dlon_s	numeric(8,5)	Longitude of vessel at start of the station in decimal degree.
dlat_e	numeric(7,5)	Latitude of vessel at end of the station in decimal degree.
dlon_e	numeric(8,5)	Longitude of vessel at end of the station in decimal degree.
startp	geometry	Position of vessel at start of the station as gis point type.
endp	geometry	Position of vessel at end of the station as gis point type.
track	geometry	Track line of vessel from start position to end position of station as gis line type.

Indexes:

```
    "pk_t_station" PRIMARY KEY, btree (trip_code, station_no)
    "nx_t_station_endp" gist (endp)
    "nx_t_station_startp" gist (startp)
    "nx_t_station_track" gist (track)
Check constraints:
    "enforce_dims_endp" CHECK (ndims(endp) = 2)
    "enforce_dims_startp" CHECK (ndims(startp) = 2)
    "enforce_dims_track" CHECK (ndims(track) = 2)
    "enforce_geotype_endp" CHECK (geometrytype(endp) = 'POINT'::text
                                        OR endp IS NULL)
    "enforce_geotype_startp" CHECK (geometrytype(startp) =
                                        'POINT'::text OR startp
                                        IS NULL)
    "enforce_geotype_track" CHECK (geometrytype(track) =
                                        'LINESTRING'::text OR
                                        track IS NULL)
    "enforce_srid_endp" CHECK (srid(endp) = 4326)
    "enforce_srid_startp" CHECK (srid(startp) = 4326)
    "enforce_srid_track" CHECK (srid(track) = 4326)
    "t_station_bot_cont_check" CHECK (bot_cont >= 0 AND bot_cont <=
                                        5)
    "t_station_bot_type_check" CHECK (bot_type >= 0 AND bot_type <=
                                        9)
    "t_station_cloud_cov_check" CHECK (cloud_cov >= 0 AND cloud_cov
                                        <= 8)
    "t_station_course_check" CHECK (course >= 0 AND course <= 359)
    "t_station_eorw_f_check" CHECK (eorw_f::text ~ '[EW]'::text)
    "t_station_eorw_s_check" CHECK (eorw_s::text ~ '[EW]'::text)
    "t_station_gear_perf_check" CHECK (gear_perf >= 1 AND gear_perf
                                        <= 4)
    "t_station_lat_f_check" CHECK (lat_f::text ~ '[2-6][0-9][0-5][0-
                                        9][0-9][0-9]'::text)
    "t_station_lat_s_check" CHECK (lat_s::text ~ '[2-6][0-9][0-5][0-
                                        9][0-9][0-9]'::text)
    "t_station_long_f_check" CHECK (long_f::text ~ '1[5-8][0-9][0-
                                        5][0-9][0-9][0-9]'::text)
    "t_station_long_s_check" CHECK (long_s::text ~ '1[5-8][0-9][0-
                                        5][0-9][0-9][0-9]'::text)
    "t_station_moon_check" CHECK (moon >= 1 AND moon <= 4)
    "t_station_nors_f_check" CHECK (nors_f::text ~ '[NS]'::text)
    "t_station_nors_s_check" CHECK (nors_s::text ~ '[NS]'::text)
    "t_station_path_check" CHECK (path >= 1 AND path <= 8)
    "t_station_sea_col_check" CHECK (sea_col >= 1 AND sea_col <= 8)
    "t_station_sea_cond_check" CHECK (sea_cond >= 0 AND sea_cond <=
                                        9)
    "t_station_swell_dir_check" CHECK (swell_dir >= 0 AND swell_dir
                                        <= 359 OR swell_dir =
                                        999)
    "t_station_swell_ht_check" CHECK (swell_ht >= 1 AND swell_ht <=
                                        3)
    "t_station_time_f_check" CHECK (time_f >= 0 AND time_f <= 2359)
    "t_station_time_s_check" CHECK (time_s >= 0 AND time_s <= 2359)
    "t_station_wind_dir_check" CHECK (wind_dir >= 0 AND wind_dir <=
                                        359 OR wind_dir = 999)
```

```
   "t_station_wind_force_check" CHECK (wind_force >= 0 AND
                                       wind_force <= 12)

Foreign-key constraints:
   "fk_t_station_area_codes" FOREIGN KEY (area)
    REFERENCES rdb.area_codes(code)
   "fk_t_station_meth_codes" FOREIGN KEY (gear_meth)
    REFERENCES rdb.meth_codes(code)
   "fk_t_station_t_door_dist_codes" FOREIGN KEY (distdoor_code)
    REFERENCES rdb.t_door_dist_codes(door_code)
   "fk_t_station_t_fish_obs_codes" FOREIGN KEY (fish_locn)
    REFERENCES rdb.t_fish_obs_codes(fish_obs_code)
   "fk_t_station_t_fix_meth_codes" FOREIGN KEY (fix_f)
    REFERENCES rdb.t_fix_meth_codes(fix_meth_code)
   "fk_t_station_t_fix_meth_codes_2" FOREIGN KEY (fix_s)
    REFERENCES rdb.t_fix_meth_codes(fix_meth_code)
   "fk_t_station_t_headline_codes" FOREIGN KEY (head_code)
    REFERENCES rdb.t_headline_codes(headline_code)
   "fk_t_station_t_trip" FOREIGN KEY (trip_code)
    REFERENCES rocklob.t_trip(trip_code)
   "fk_t_station_t_wing_dist_codes" FOREIGN KEY (distwing_code)
    REFERENCES rdb.t_wing_dist_codes(wing_dist_code)
```

5.7 Table 7: t_stat_comm

Comment: Comments for a station in a trip.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
comments	text	No	Comments for this station - should include comments about catch or any special action taken during tow.

```
Indexes:
   "pk_t_stat_comm" PRIMARY KEY, btree (trip_code, station_no)

Foreign-key constraints:
   "fk_t_stat_comm_t_station" FOREIGN KEY (trip_code, station_no)
    REFERENCES rocklob.t_station(trip_code, station_no)
```

5.8 Table 8: t_phy_stage

Comment: Phyllosoma life history stage table.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Trip code as in the trip table.
station_no	integer	No	Station number as in station table.
species	character(3)		3-char species code, refer rdb.curr_spp.
stage	character varying(3)		Code for phyllosoma larvae development stage. Refer v_phy_stage_codes.
no_a	integer		Number of animals at this stage.

```
Foreign-key constraints:
   "fk_t_phy_stage_t_station" FOREIGN KEY (trip_code, station_no)
    REFERENCES rocklob.t_station(trip_code, station_no)
```

5.9 Table 9: t_evpsumm

Comment: Table storing start and finish times and positions of stations in envelopes of DAS data.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Standard 7 char code for the trip.
evp_no	integer	No	Number to identify an envelope of DAS data.
station_no	integer		Number for station assigned to envelope. Unassigned envelopes are given a station number of 0.
lat_s	double precision	No	Latitude for start of envelope/station.
long_s	double precision	No	Longitude for start of envelope/station.
time_s	timestamp without time zone	No	Time at start of envelope/station.
time_f	timestamp without time zone	No	Time at finish of envelope/station.

```
Indexes:
   "pk_t_evpsumm" PRIMARY KEY, btree (trip_code, evp_no)

Foreign-key constraints:
   "fk_t_evpsumm_t_trip" FOREIGN KEY (trip_code)
    REFERENCES rocklob.t_trip(trip_code)
```

5.10 Table 10: t_dasdata

Comment: Table storing filtered records from das envelopes.

Column	Type	Null?	Description
trip_code	character varying(7)	No	Standard 7 char code for the trip.
evp_no	integer	No	Number to identify an envelope of DAS data.
station_no	integer	No	Number for station assigned to envelope.
tally_id	character varying(12)	No	Label assigned by REACT to identify records.
timestamp	timestamp without time zone	No	Rounded timestamp when observation was made.
obs	double precision	No	Value of observation for tally_id at timestamp.

5.11 Table 11: t_deploy_meth_codes

Comment: Contains codes and their descriptions of the methods used to deploy puerulus and juvenile collectors.

Column	Type	Null?	Description
deploy_meth	character varying(10)	No	Code of the method of collector deployment.
descrptn	character varying(100)		Description of the collector deployment method code.

Indexes:
"pk_t_deploy_meth_codes" PRIMARY KEY, btree (deploy_meth)

5.12 Table 12: t_stage_codes

Comment: Descriptions of crayfish development stages from phyllosoma larvae to juveniles.

Column	Type	Null?	Description
species	character varying(3)	No	3-char species code, refer rdb.curr_spp.
type	character varying(10)	No	Flag to denote whether the stage is for a phyllosoma or a puerulus.
stage	character varying(5)	No	Code for phyllosoma larvae or puerulus development stage.
descrptn	text		Description of phyllosoma larvae or puerulus development code.
key	character varying(10)		Group number in which a stage belongs. Used by some researchers to group similar stages together into one 'super-stage'.
lesser_stage	character varying(10)		Stage, as defined by Lesser, J.H.R. 1978. Phyllosoma larvae of Jasus edwardsii and their distribution off the east coast of North Island. N.Z. Journal of Marine and Freshwater Research 12 (4): 357-70.

```
Indexes:
   "pk_t_stage_codes" PRIMARY KEY, btree (species, stage)
Check constraints:
   "t_stage_codes_type_check" CHECK
    ("type"::text ~ 'PUERULUS|PHYLLOSOMA'::text)
```

5.13 View 1: v_scyllarus

```
Comment: View of all scyllarus (SHL) phyllosoma life history stage
         data.

Column          Type

trip_code       character varying(7)

station_no      integer

species         character(3)

stage           character varying(3)

no_a            integer

View definition:
SELECT t_phy_stage.trip_code, t_phy_stage.station_no,
t_phy_stage.species, t_phy_stage.stage, t_phy_stage.no_a
 FROM rocklob.t_phy_stage
 WHERE t_phy_stage.species::text = 'SHL'::text;
```

5.14 View 2: v_jasus

Comment: View of all jasus (PHY) phyllosoma life history stage data.

Column	Type
trip_code	character varying(7)
station_no	integer
species	character(3)
stage	character varying(3)
no_a	integer

View definition:

```
SELECT t_phy_stage.trip_code, t_phy_stage.station_no,
t_phy_stage.species, t_phy_stage.stage, t_phy_stage.no_a
  FROM rocklob.t_phy_stage
 WHERE t_phy_stage.species::text = 'PHY'::text;
```

5.15 View 3: v_puer_stage_codes

```
Comment: View of descriptions of development stages of rock lobsters
         from puerulus to juveniles.

Column          Type

species         character varying(3)

type            character varying(10)

stage           character varying(5)

descrptn        text

key             character varying(10)

lesser_stage    character varying(10)

View definition:
SELECT t_stage_codes.species, t_stage_codes."type",
t_stage_codes.stage, t_stage_codes.descrptn, t_stage_codes."key",
t_stage_codes.lesser_stage
  FROM rocklob.t_stage_codes
 WHERE t_stage_codes."type"::text = 'PUERULUS'::text;
```

5.16 View 16: v_phy_stage_codes

Comment: View of descriptions of development stages of phyllosoma larvae.

Column	Type
species	character varying(3)
type	character varying(10)
stage	character varying(5)
descrptn	text
key	character varying(10)
lesser_stage	character varying(10)

View definition:

```
SELECT t_stage_codes.species, t_stage_codes."type",
t_stage_codes.stage, t_stage_codes.descrptn, t_stage_codes."key",
t_stage_codes.lesser_stage
  FROM rocklob.t_stage_codes
 WHERE t_stage_codes."type"::text = 'PHYLLOSOMA'::text;
```

6 rocklob business rules

6.1 Introduction to business rules

The following are a list of business rules pertaining to the rocklob database. A business rule is a written statement specifying what the information system (i.e., any system that is designed to handle rock lobster life cycle data) must do or how it must be structured.

There are three recognized 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. Referential constraints, range checks, and algorithms both in the database and during data validation implement the formula and validation type rules.

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

Puerulus collector locations (t_locations)

location	Location code, must be unique.
deploy_meth	Puerulus collector deployment method code must be a valid code as listed in the table *t_deploy_meth_codes*.

Puerulus catch from a collector (t_catch)

location	Must be equal to a location code held in the *t_locations* table.
date_checked	The date a collector is checked must be a legitimate date.
collector_no	Must be a unique number within all collectors checked at a location on a date.
no_catch	Must be a integer greater than or equal to zero.
effort	Must be a integer greater than or equal to zero.
month_collected	Month collected must be an integer ranging from 1 to 12.

Puerulus length frequencies (t_puer_lfreq)

Multiple columns check on location, date_checked and collector_no:
The combination of *trip_code*, *date_checked* and *collector_no* must exist in the *t_catch* table.

m_date Date of measurement of pueruli and/or post-pueruli. Must be a legitimate date on or after *date_checked*.

len_p Carapace length (mm), should fall within the reasonable range of 5-105.

no_p Must be an integer greater than or equal to zero.

Multiple columns check on no_p and len_p:
The associated value of *len_p* should not exceed 14.5.

no_pp Must be an integer greater than or equal to zero.

Multiple columns check on no_pp and len_p:
The value of *len_p* should not be less than 14.5.

no_m_pp Must be an integer greater than or equal to zero.

no_f_pp Must be an integer greater than or equal to zero.

Multiple columns check on no_pp, no_m_pp, and no_f_pp:
The sum *of no_m_pp* and *no_f_pp* cannot exceed *no_pp*.

Puerulus length frequencies (t_puer_stage)

Multiple columns check on location, date_checked and collector_no:
The combination of *trip_code*, *date_checked* and *collector_no* must exist in the *t_catch* table.

stage Puerulus life cycle stage number, must be a valid code as listed in the view *v_puer_stage_codes*.

no_a Number of pueruli at the stage number, should fall within the reasonable range of 1-50.

Phyllosoma survey trip details (t_trip)

trip_code	Trip code, must be unique. Trip codes are in the following format: 3 character vessel code (see the *vessels* table in the **rdb** database for available codes); 2-digit year (e.g., 99 = 1999, 00 = 2000); 2-digit sequential trip number for each vessel each year.
proj_code	Project code must be a valid code within the NIWA project management system.
date_s	The start date of the trip must be a legitimate date.
date_f	The start date of the trip must be a legitimate date.
	Multiple column checks on start and finish dates: The start date must not be later than the finish date.
areas	Each of the listed area codes must be a valid code as listed in the *area_codes* table in the **rdb** database.
mainspp	Each of the listed species codes must be a valid code as listed in the *curr_spp* table in the **rdb** database.

Phyllosoma survey station details (t_station)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

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

course Course must be within the range of 0 – 359 degrees.

date_s The date at the start of a station must be a legitimate date.

Multiple column checks on start date:
The date must fall within the range of the range of the trip 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.

fix_s }
fix_f } The method of position fix code must be valid code as listed in Appendix 1.

lat_s Must be a valid latitude

NorS_s Northern or Southern Hemisphere at station start, must be equal to either "N" or "S".

long_s Must be a valid longitude.

EorW_s Longitude east or west at station start, must be equal to either "E" or "W".

bot_gs Depth of sea bottom must not be less than depth of gear

date_f The date at the finish of a station must be a legitimate date.

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

time_f	Finish time of the station must be a valid 24-hour time and fall within the range of 0 – 2359. **Multiple columns checks on date and time:** The start date must not be later than the finish date and within a reasonable time period.
lat_f	Must be a valid latitude
NorS_f	Northern or Southern Hemisphere at station finish, must be equal to either "N" or "S".
long_f	Must be a valid longitude.
EorW_f	Longitude east or west at station finish, must be equal to either "E" or "W". **Multiple columns checks on position:** The finish position should be within a reasonable distance from the start position.
bot_gf	Depth of sea bottom must not be less than depth of gear
min_gdepth	Minimum gear depth must be less than or equal to the depth of gear at the start and finish of the station.
max_gdepth	Maximum gear depth must be greater than or equal to the minimum gear depth and the depth of gear at the start and finish of the station
gear_meth	Gear method code must be a valid code as listed in the *meth_codes* table in the **rdb** database.
gear_code	Must within the range 1 – 3 to relate to *gear1*, *gear2*, and *gear3* respectively in the *t_trip* table.
gear_perf	The gear performance code must be valid code as listed in Appendix 1.
path	The path code must be valid code as listed in Appendix 1.
speed	The vessel's recorded speed during the station should be within the range 0 – 5 knots and be reasonable for the gear method.

distance The distance traveled during the station should be reasonable for the type of gear method.

Multiple columns check on: distance; start and finish positions; and speed and start/finish times:
The distance traveled during a station as calculated by (1) the difference between start and finish positions; (2) speed * elapsed time; and (3) recorded distance should be in approximate agreement.

head_code Headline height code must be a valid code as listed in the *t_headline_codes* table in the **rdb** database.

distwing_code Distance between trawl wings code must be a valid code as listed in the *t_wing_dist_codes* table in the **rdb** database.

distdoor_code Distance between trawl doors code must be a valid code as listed in the *t_door_dist_codes* table in the **rdb** database.

wind_dir Wind direction must fall within the range of 0-359, 999.

wind_force Wind force must fall within the range of 0 – 12.

air_temp Air temperature should fall within the reasonable range of 5 – 30.

air_press Air pressure should fall within the reasonable range of 960 to 1040.

cloud_cov Cloud cover must fall within the range of 0-8.

sea_cond The sea condition code must be valid code as listed in Appendix 1.

sea_col The sea colour code must be valid code as listed in Appendix 1.

swell_ht The swell height code must be valid code as listed in Appendix 1.

swell_dir Wind direction must fall within the range of 0-359, or equal 999.

bot_type The bottom type code must be valid code as listed in Appendix 1.

bot_cont The bottom contour code must be valid code as listed in Appendix 1.

surf_temp Sea surface temperature should fall within the reasonable range of 5 – 28.

bot_temp Sea bottom temperature should fall within the reasonable range of 3 – 25.

wind_spd Wind speed should fall within the reasonable range of 0 - 30.

secchi Secchi disc distance should fall within the reasonable range of 0 – 40.

moon Moon phase must fall within the range of 1 - 4.

Phyllosoma survey station comments (t_stat_comm)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

Multiple columns check trip code and station number:
The combination of trip code and station number must exist in the *t_station* table.

Phyllosoma survey stage data (t_phy_stage)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

Multiple columns check trip code and station number:
The combination of trip code and station number must exist in the *t_station* table.

species The species code must be valid code as listed in Appendix 1.

stage The phyllosoma life history stage code must be valid code as listed in Appendix 1.

Phyllosoma survey DAS envelope summary (t_evp_summ)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

Multiple columns check trip code and station number:
The combination of trip code and station number must exist in the *t_station* table.

evp_no Must be a unique number within a single trip.

lat_s Must be a valid latitude.

long_s Must be a valid longitude

time_s Must be a valid date and time.

time_f Must be a valid date and time.

Phyllosoma survey DAS data (t_dasdata)

trip_code Must be equal to a trip code as listed in the *t_trip* table.

station_no Must be a unique number within a single trip.

Multiple columns check trip code and station number:
The combination of trip code and station number must exist in the *t_station* table.

evp_no Must be a unique number within a single trip.

tally_id The DAS data tally id code must be valid code as listed in Appendix 1.

timestamp Must be a valid date and time.

Puerulus and juvenile collector deployment codes (t_deploy_meth_codes)

deploy_meth Must contain a value.

descrptn Must contain a value.

Puerulus development stage codes (t_stage_codes)

species Species code must be a valid species code as listed in Appendix 1.

type Must be equal to either "PHYLLOSOMA" or "PUERULUS".

stage Puerulus development stage code must contain a value.

Multiple columns check on species, type, and stage:
The values in the species, type, and stage attributes must be a unique combination.

7 References

1 Booth, J. D., Carruthers, A. D., Bolt, C. D., and Stewart, R. A. 1991: Measuring depth of settlement in the red rock lobster, *Jasus edwardsii. New Zealand journal of marine and freshwater research*.

2 Booth, J. D. & Forman, J. S. 1995: Larval recruitment in the red rock lobster, *Jasus edwardsii. N.Z. Fisheries Assessment Research Document 95/7*. 46p.

3 Booth, J. D., Forman, J. S., and Stotter, D. R. 1998: Abundance of early life history stages of the red rock lobster, *Jasus edwardsii*, with management implications. *N.Z. Fisheries Assessment Research Document 98/10*. 71p.

4. Mackay, K. A. 1998: Database documentation: trawl. *NIWA Internal Report No. 16*. 35p.

Appendix 1 – Reference Code Tables

Position fix method code

01	Radar
02	Dead reckoning
03	Astrofix
04	Transect marks
05	Radio (RDF)
06	Radar and RDF
07	Satnav
08	Global positioning satellite (GPS)
09	Local knowledge
10	GPX

Gear performance code

2.	Excellent
3.	Satisfactory, catch unlikely to be reduced by performance
4.	Unsatisfactory, catch probably reduced by malfunction or damage
5.	Unsatisfactory, catch reduced by malfunction or damage

Path code

1.	Horizontal straight line
2.	Vertical straight line
3.	Closed circle or loop
4.	Closed triangle or square
5.	Zigzag
6.	U-bend
7.	Contour at constant depth
8.	Retrack on straight line

Sea condition code

0	Calm, glassy	0m
1	Calm	0 – 0.1m
2	Smooth	0.1 – 0.5m
3	Slight	0.5 – 1m
4	Moderate	1 – 2.5m
5	Rough	2.5 – 4m
6	Very rough	4 – 6m
7	High	6 - 10m
8	Very high	10 – 15m
9	Huge	over 15m

Sea colour code

01	Deep blue
02	Blue
03	Light blue
04	Greeny blue
05	Bluey green
06	Deep green
07	Green
08	Yellow green

Swell height code

1	Low	0 – 2m
2	Moderate	2 – 4m
3	Heavy	over 4m

Bottom contour code

0	Unknown
1	Smooth/flat
2	Undulating
3	Hillocky
4	Rugged
5	Very rugged

Bottom type code

0	Unknown
1	Mud or ooze
2	Mud with some sand
3	Sand
4	Sand/gravel and shells
5	Shells (broken)
6	Gravel
7	Rock
8	Coral
9	Stone
10	Live shell beds
11	Mud with broken shells
12	Sponge beds

Species code

PHC	Packhorse rock lobster	*Jasus verreauxi*
PHY	Phyllosoma	
PRK	Prawn killer	*Ibacus alticrenatus*
PUE	Puerulus	
CRA	Rock lobster	*Jasus edwardsii*
SHL	Shovelnosed lobster	*Scyllarus* sp
SLO	Spanish lobster	*Arctides* sp

DAS data tally id code

GPLat	Latitude
GPLong	Longitude
GPHDT	GPS Heading
GPVkts	GPS Velocity
KMCdbt	Kaijo Denki Depth
ReqLen	Required warp length
SSSal	Surface salinity
SSMTW	Surface water temperature
BarPressur	Barometric Pressure
AirTemp	Air Temperature
WindDirn	Wind Direction
Nvelocity	Wind Speed
SNSDep	Scanmar gear depth
SNSDi1	Scanmar distance 1
SNSDi2	Scanmar distance 2
SNSDiR1	Scanmar distance 1 change
SNSDiR2	Scanmar distance 2 change
CN22	Depth from Furuno CN22
MagLat	Latitude from Magnavox GPS
MagLong	Longitude from Magnavox GPS
EK5dbt1	EK500 depth 1
EK5dbt2	EK500 depth 2
PlHDT	Plath heading
NKvhw	Naviknot speed
LenStbd	Length of starboard warp
LenPort	Length of port warp
TenStbd	Length of starboard warp
TenPort	Tension in port winch
CI30Depth	Depth of top layer on doppler
CI30Spd	Speed of doppler log current
CI30Dir	Direction of doppler current
CI30L1Dep	Depth of first layer on doppler
CI30L1Cur	Speed of first layer doppler log current
CI30L1Dir	Direction of first layer doppler current
CI30L2Dep	Depth of second layer on doppler
CI30L2Cur	Speed of second layer doppler log current
CI30L2Dir	Direction of second layer doppler current
CI30L3Dep	Depth of third layer on doppler
CI30L3Cur	Speed of third layer doppler log current
CI30L3Dir	Direction of third layer doppler current