

**Database documentation: rlcs
(B) Logbook schema**

N. Walker & K. A. Mackay

NIWA Fisheries Data Management
Database Documentation Series

Revised on 9 January 2014

Table of Contents

1	Database Document Series	4
2	Rock lobster Voluntary Logbook Program.....	4
	2.1 Operational versus non-operational data repository	5
3	Data Structures	5
	3.1 Table Relationships.....	5
	3.2 Database Design.....	8
4	Table Summaries	9
5	rlcs Logbook Tables.....	10
	5.1 Table 1: Fishers	10
	5.2 Table 2: Trips.....	11
	5.3 Table 3: PotLifts	13
	5.4 Table 4: Lobsters.....	16
	5.5 Table 5: Vessels.....	17
	5.6 Table 6: Zones	17
	5.7 Table 7: StatAreas.....	18
	5.8 Table 8: TWC.....	19
6	rlcs Logbook Business Rules	20
	6.1 Introduction to business rules.....	20
	6.2 Summary of rules	21
	Appendix 1 : Data Entry and Validation	27

List of Figures

Figure 1: Entity Relationship Diagram (ERD) for the voluntary logbook schema in the rlcs database.	6
--	---

Revision History

Version	Change	Date	Responsible
1.0	First release	2002 ?	Kevin Mackay
1.1	?	5-Sep-02	Kevin Mackay
1.2	Added lat & long attributes to PotLifts table, & Postcode to Fishers	11-Nov-09	David Fisher
1.3	Changed Trips.Time to time(4) from time(7)	9-Jan-14	David Fisher

1 Database Document Series

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

The Ministry of Fisheries data set incorporates historic research data, data collected more recently by MAF Fisheries prior to the split in 1995 of Policy to the Ministry of Fisheries and research to NIWA, and currently data collected by NIWA and other agencies for the Ministry of Fisheries.

This document is a brief introduction to the rock lobster voluntary logbook sampling programme and its associated data as held in the database **rlcs**, and is a part of the database documentation series produced by NIWA.

Voluntary logbook sampling refers to the sampling of the catch from rock lobster fishing by the fishers themselves, as opposed to catch sampling in which the catch is sampled by observers. The catch sampling is documented in Mackay & George (2002)¹.

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 **rlcs**, and the relationships between these tables and other databases.

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

Access to this database and data is restricted to specific Nominated Personnel as specified in the current Schedule 6 of the Data Management contract between the Ministry of Fisheries and NIWA. Any requests for data should in the first instance be directed to the Ministry of Fisheries.

2 Rock lobster Voluntary Logbook Program

Effective fisheries management relies on information about the current characteristics of the catch. Data on the total catch weight and fishing effort are collected via compulsory returns. Information regarding the composition of the catch must be gained from more detailed sampling. Currently, two methods are used to gather data on the composition of rock lobster catches. Firstly, personnel are employed by NIWA and the Rock Lobster Industry Council (RLIC) to carry out catch sampling. Secondly, there is a voluntary logbook, which is used by the fishermen to record the characteristics of their own catch. Since the 1980's there have been several attempts at running voluntary logbook programmes for rock lobster fishers by MAF Fisheries and the fishing industry. The latest, and most concerted, logbook programme has been running since the New Zealand Fishing Industry Board (NZFIB) established it in 1993. The logbook is currently

¹ MACKAY, K.A. & GEORGE, K., 2002, Database documentation: **rlcs**. (A) Catch sampling schema. *NIWA Internal Report*. 53p.

operating in six of the nine rock lobster quota management areas and has become a Fisheries Required Service in CRA 2 and CRA 8. This means that the management committees in CRA 2 and CRA 8 are required to ensure that at least twenty percent of the fishermen in their management area participate in the logbook programme.

Each participating fisher is asked to sample four of their pots (chosen randomly) each day that they would normally fish these pots. Every lobster in each of the selected pots is counted. Up to twenty-five lobsters from each of these four pots is measured and sexed (females are also assigned a stage) and this forms a sub-sample of the catch from the pot. To enable precise data standardisation and analyses of other variables, participating fishers also complete surveys each year that describe their gear and fishing operations.

2.1 Operational versus non-operational data repository

The operational data (that is, day-to-day processing) generated from the logbook programme is managed by Trophia Research and Consulting. Originally, operational data was entered using a Microsoft® DOS-based programme called **ROCKY** and was then transferred into a Microsoft® Access database **CRAVLB**. As a result of a project review, the logbook programme, data entry forms and the database were updated. Data entry and data storage is now managed by a Microsoft® Access Database called “the Rock Lobster Logbook Database” or **RLLBD**.

The Data Manager, on behalf of the Ministry of Fisheries, carries out long-term data archiving of the **RLLBD**. A separate schema within the **rlcs** database manages these data. This schema within **rlcs** is a duplicate of that held within **RLLBD**. Data are received and loaded by the Data Manager from Trophia at a fixed interval, currently set at once a year.

3 Data Structures

3.1 Table Relationships

This database encompasses several tables. The ERD for the voluntary logbook (VLB) schema within **rlcs** (Figure 1) shows the physical data model structure² 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.

All of the table’s attributes are shown in the ERD. The underlined attributes represent the table’s primary key³. 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 **rlcs** database also contain special attributes, called foreign keys⁴.

² Also known as database *schema*

³ A primary key is an attribute or a combination of attributes that contains an unique value to identify that record.

⁴ 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.

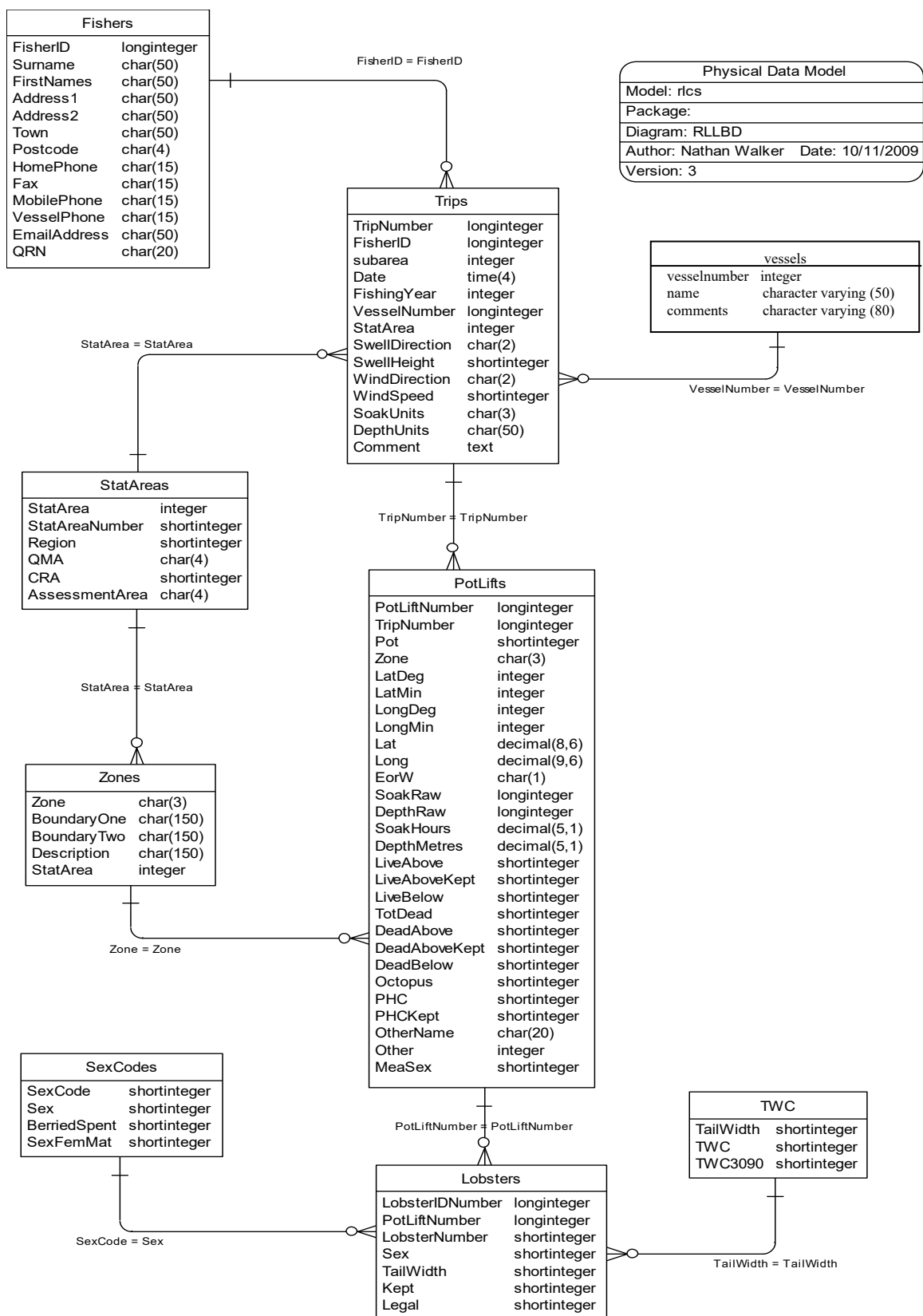


Figure 1: Entity Relationship Diagram (ERD) for the voluntary logbook schema in the **rlcs** database.

Section 5 shows a listing of all the **rlcs** VLB tables as implemented by the Empress 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: UNIQUE index_name ON (attribute [, attributes])

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

The **rlcs** 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 **rlcs** are of the type *one-to-many*⁵. This is shown in the ERD by connecting a fork⁶ (indicating 'many') from the child table; e.g., *PotLifts*, to the parent table; e.g., *Trips*, with a single line (indicating 'one') pointing to the parent. For example, consider the relationship between the tables' *Trips* (the parent table) and *PotLifts* (the child table). Any one trip in *Trips* can have one or more pot lifts in *PotLifts*, but any one pot lift can only be a part of one trip. Note that the word 'many' applies to the possible number of records another is associated with. For a given instance, there might be zero, one, two, or more associated records, but if it is ever possible to have more than one, we use the word 'many' to describe the association.

Note that the one-to-many relationships can be either mandatory or optional. The optional relationship, denoted in the ERD by the symbol "O" at one or both ends of the relationship line, means that a record does not have to have any associated records. Conversely, the mandatory relationship denoted in the ERD by a bar symbol across the relationship line, means that a record has to have at least one associated record. For example, if we consider again the one-to-many relationship between the tables *Trips* and *PotLifts*, which has a mandatory 'one' and an optional 'many'. This means that one sample record can have zero or more (many) pot lift, recorded in the effort tables, but one unit effort in the effort table must have one, and only one, associated record in the trip table.

These relationships are enforced in the database by the use of referential constraints⁷. 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:

- 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.

⁵ 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 fishing trip in *Trips* can have many pot lifts in *PotsLifts* but any one pot lift can only come from one trip.

⁶ Also known as a crow's foot.

⁷ Also known as integrity checks.

All constraints in **rlcs** prevent the latter from occurring. Constraints are shown in the table listings by the following format:

```
Referential:      constraint_name (attribute[, attribute])  INSERT
                                                             DELETE
                    Parent_table (attribute[, attribute])
```

where a constraint can either be INSERT (prevents (iii) and (ii) from occurring) or DELETE (prevents (i) or (ii) from occurring). For example, consider the following constraint found in the table *data_sets*:

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

```
Referential:      No Such Fisher (FisherID) INSERT Fishers (FisherID)
```

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

```
*** User Error: insert constraint 'No Such Fisher' violation
```

All 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:      NORMAL (2, 15) index_name ON (attribute [, attributes])
```

Note that indices may be simple, pointing to one attribute, or composite pointing to more than one attribute. The numbers “...(2, 15)...” in the syntax are Empress DBMS default values relating to the amount of space allocated to index storage.

3.2 Database Design

The VLB schema is modelled around a simple hierarchical/relational construct. Simply put: a Fisher (*Fishers*, table 1) goes out on one or many fishing trips (*Trips*, table 2); on any one trip, up to 4 pots are sampled (*PotLifts*, table 3); and from any one of these pots, all rock lobsters are measured (*Lobsters*, table 4).

Some interesting points to note about the physical table designs:

- In the process of potting, several species can be caught in any one pot, including two species of rock lobster. The designers of the *PotLifts* table have chosen to handle this by defining attributes for three main species, and having a two fields to generic handle any four species that the fisher may have recorded. The majority of the attributes, such as *LiveAbove*, *LiveBelow*, etc., deal exclusively with the red rock lobster (*Jasus edwardsii*). An alternative solution to this might be to construct a Catch table that will handle all the catch information for any number of different species per pot lift, where each species is identified by a species code.
- In the real-world, fishers use different units of measure for sea-bottom depth and soak time. The schema allows to fishers to record these data in their own units, with depth and soak time values stored as *DepthRaw* and *SoakRaw* respectively in the *PotLifts* table. The actual units of measure used are stored in *DepthUnits* and

SoakUnits respectively on the *Trips* table. Upon data entry, these values are used to generate depths in metres and soak times in hours to provide standard units of measure. These data are stored in *DepthMetres* and *SoakHours* respectively in the *PotLifts* table.

In addition to these base tables, there are three tables, which are used to store various codes used in the main tables, and their descriptions. They are *Vessels* (Table 5), *Zones* (Table 6), and *StatAreas* (Table 7).

The last table, TWC (Table 8), provides a simple mapping between the actual tail width (1mm intervals) of a rock lobster with a coarser interval (2mm) for analysis purposes only.

4 Table Summaries

The following is a listing and brief outline of the voluntary logbook tables contained in *rlcs*.

1. **Fishers** : Contains information regarding the fisher including name and contact details.
2. **Trips** : Contains details of the fishing trip, including vessel used, statistical area fished, sea and wind conditions, and units used for soak times and depths.
3. **Potlifts** : Contains details concerning each pot lift, including the soak time and depth and the catch.
4. **Lobsters** : Contains details of individual rock lobsters that were caught, including their sex and stage, tail width and whether or not they were retained.
5. **Vessels** : Contains information relevant to the vessel, including the name, registration number and other characteristics. This information is gathered from the annual survey forms completed by the fishers.
6. **Zones** : Contains a list of the zones, effectively sub-areas within the statistical areas. This includes information on the zones and the corresponding statistical areas and boundaries. Zones have not been described for CRA 1, 6 or 7 as these areas do not utilise the rock lobster logbook.
7. **StatAreas** : Contains the list of statistical areas, including the region and the QMA each statistical area belongs.
8. **TWC** : Contains the list of all possible tail widths in 1mm intervals, their converted values in 2mm intervals, and their 2mm interval values between 30 and 90mm.

5 rlcs Logbook Tables

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

5.1 Table 1: Fishers

Comment: This table contains information regarding the fisher including name and contact details.

Attributes	Data Type	Null?	Comment
FisherID	longinteger	No	An identifying number that is unique for each fisher throughout the country
Surname	character(50,1)	No	The surname of the fisher
FirstNames	character(50,1)		The first names of the fisher
Address1	character(50,1)		First line of the fisher's address
Address2	character(50,1)		Second line of the fisher's address
Town	character(50,1)		The town in which the fisher resides
Postcode	character(4,1)		The post code for the fisher's address
HomePhone	character(15,1)		The home phone number of the fisher
Fax	character(15,1)		The fax number of the fisher
MobilePhone	character(15,1)		The mobile phone number of the fisher
VesselPhone	character(15,1)		The mobile phone number of Fisher's vessel
EmailAddress	character(50,1)		The fisher's email address
QRN	character(20,1)		The fisher's Quota Registration Number

Creator: dba

Indices: UNIQUE BTREE FisherID ON (FisherID)

5.2 Table 2: Trips

Comment: This table contains details of the fishing trip, including vessel used, statistical area fished, sea and wind conditions, and units used for soak times and depths.

Attributes	Data Type	Null?	Comment
TripNumber	longinteger	No	An identifying number that is unique for each fishing trip throughout the country
FisherID	longinteger	No	An identifying number that is unique for each fisher throughout the country (refer to Fishers - <i>FisherID</i>)
subarea	integer		An integer to identify each subarea that the trip went through
Date	time(4)		Date of the fishing trip
FishingYear	integer		Fishing year; e.g., 1 Apr 93 to 31 Mar 94 = Fishing Year 1993
VesselNumber	longinteger		Registration Number of the Vessel
StatArea	integer		The number of the statistical area in which most of the fishing was performed during the fishing trip (refer to StatAreas - <i>StatArea</i>)
SwellDirection	character(2,1)		The direction of the swell during the fishing trip (if there was a North-east swell then it would be NE)
SwellHeight	smallint		The height (m) of the swell during the fishing trip
WindDirection	character(2,1)		The direction of the wind during the fishing trip (if there was a North-east wind then it would be NE)
WindSpeed	smallint		The speed (knots) of the wind during the fishing trip
SoakUnits	character(3,1)		The units in which the soak times are recorded for each of the pots lifted. "HRS" = hours "DAY" = days

5.2 Table 2: Trips (cont...)

Attributes	Data Type	Null? Comment
DepthUnits	character(3,1)	The units in which the depths are recorded for each of the posts lifted. "FEE" = feet "MET" = metres "FAT" = fathoms
comments	text(20,20,20,1)	A field for any text comments that the fishers might write on the datasheets

Creator: dba

Referential: No Such Fisher (FisherID) INSERT Fishers (FisherID)

Indices: UNIQUE BTREE PrimaryKey ON (TripNumber)
NORMAL (2, 15) BTREE FisherID ON (FisherID)
NORMAL (2, 15) BTREE StatAreasTrips ON (StatArea)
NORMAL (2, 15) BTREE TripsVesselNumber ON (VesselNumber)

5.3 Table 3: PotLifts

Comment: This table contains details concerning each potlift, including the soak time, depth, and the catch.

Attributes	Data Type	Null?	Comment
PotLiftNumber	longinteger	No	An identifying number that is unique for each potlift throughout the country
TripNumber	longinteger	No	An identifying number that is unique for each fishing trip throughout the country (refer to Trips - <i>TripNumber</i>)
Pot	smallint	No	The number of the pot lifted (can be only between 1 and 4 inclusive)
Zone	character(3,1)		A code for the zone in which the pot was lifted (refer to Zones - <i>Zone</i>)
LatDeg	integer		Degrees latitude for the position
LatMin	integer		Minutes of latitude for the position
LongDeg	integer		Degrees longitude for the position
LongMin	integer		Minutes of longitude for the position
Lat	decimal(8,6)		Latitude in decimal degrees
Long	decimal(9,6)		Longitude in decimal degrees
EorW	character(1,1)		Meridian for the longitude, east or west of Greenwich
SoakRaw	longinteger		The length of time the pot was in the water (soak time). The units for this field is recorded in the <i>SoakUnits</i> field in the Trips table
DepthRaw	longinteger		The depth that the pot was set in. The units for this field is recorded on the <i>DepthUnits</i> field in the Trips table
SoakHours	decimal(5,1)	No	The soak time in hours. This is calculated from PotLifts - <i>SoakRaw</i> and Trips - <i>SoakUnits</i>

5.3

Table 3: PotLifts (cont...)

Attributes	Data Type	Null?	Comment
DepthMetres	decimal(5,1)	No	The depth in metres. This is calculated from PotLifts - <i>DepthRaw</i> and Trips - <i>DepthUnits</i>
LiveAbove	smallint		The number of live rock lobsters above the Minimum Legal Size (MLS) caught in the pot in addition to those measured and recorded in the Lobsters table
LiveAboveKept	smallint		Whether or not the additional live rock lobsters above MLS were retained: 0 = No 1 = Yes
LiveBelow	integer		The number of live rock lobsters below the MLS caught in the pot in addition to those measured and recorded in the Lobsters table
TotDead	smallint		The total number of dead rock lobsters found in the pot
DeadAbove	smallint		The number of dead rock lobsters above the MLS caught in the pot
DeadAboveKept	smallint		Whether or not the additional dead rock lobsters above MLS were retained: 0 = No 1 = Yes
DeadBelow	smallint		The number of dead rock lobsters below the MLS caught in the pot
Octopus	smallint		The number of octopi caught in the pot
PHC	smallint		The number of packhorse Lobster (PHC) caught in the pot
PHCKept	smallint		Whether or not the packhorse lobsters were retained: 0 = No 1 = Yes
OtherName	character(20,1)		The name of one other species caught, of which whose

		numbers are recorded in the field <i>Other</i>
<i>Other</i>	integer	The number caught of the species named in the field <i>OtherName</i>
		<i>MeaSex</i> smallint
Creator:	dba	
Indices:	UNIQUE BTREE PotLiftNumber ON (PotLiftNumber)	
	NORMAL (2, 15) BTREE PotLiftsZone ON (Zone)	
	NORMAL (2, 15) BTREE TripNumber ON (TripNumber)	

5.4 Table 4: Lobsters

Comment: This table contains details of individual rock lobsters that were caught, including their sex and stage, tail width and whether or not they were retained.

Attributes	Data Type	Null?	Comment
LobsterIDNumber	longinteger	No	An identifying number that is unique for each lobster measured throughout the country
PotLiftNumber	longinteger	No	An identifying number that is unique for each potlift throughout the country (refer to PotLifts - <i>PotLiftNumber</i>)
LobsterNumber	integer		The number of the lobster measured from within each pot
Sex	smallint	No	The sex of the rock lobster. Each number corresponds with a gonad stage code as listed in the table <i>t_gon_stg_meth</i> in the rdb database for the <i>stage_meth</i> "CF": 1 = MM = mature male 2 = IF = immature female 3 = MF = mature female 4 = BF = berried female 5 = SF = spent female 6 = UF = unidentified stage female
TailWidth	integer		The tail width (mm) of the rock lobster
Kept	smallint	No	Whether or not the rock lobster was retained: 0 = no 1 = yes
Legal	smallint	No	Whether or not the rock lobster was legal: 0 = no 1 = yes

Creator: dba

Indices: UNIQUE BTREE LobsterIDNumber ON (LobsterIDNumber)
NORMAL (2, 15) BTREE LobstersTailWidth ON (TailWidth)
NORMAL (2, 15) BTREE PotLiftNumber ON (PotLiftNumber)

5.5 Table 5: Vessels

Comment: This table contains information relevant to the vessel such as vessel name and registration number. This information is gathered from the annual survey forms completed by the fishers.

Attributes	Data Type	Null?	Comment
VesselNumber	longinteger	No	The vessel number supplied by FishServe
Name	character(50,1)		The name of the vessel
comments	character varying (80)		
Creator:	dba		
Indices:	UNIQUE BTREE PrimaryKey ON (VesselNumber)		

5.6 Table 6: Zones

Comment: This table contains a list of the zones. This includes information on the zones and the corresponding statistical areas and boundaries. See appendix 2 for a list of the zones developed to date. Zones have not been described for CRA 1, 6 or 7 as these areas do not utilise the rock lobster logbook

Attributes	Data Type	Null?	Comment
Zone	character(3,1)	No	A code identifying zones, these are unique codes countrywide
BoundaryOne	character(150,1)		Text describing the geographic extent of the first boundary of the zone
BoundaryTwo	character(150,1)		Text describing the geographic extent of the second boundary of the zone
Description	character(150,1)		A further description of the zone (if the boundaries are not descriptive enough)
StatArea	integer		The statistical area in which the zone is located (refer to StatAreas - StatArea)
Creator:	dba		
Indices:	UNIQUE BTREE PrimaryKey ON (Zone) NORMAL (2, 15) BTREE StatArea ON (StatArea)		

5.7 Table 7: StatAreas

Comment: This table contains the list of statistical areas, including the region and the QMA each statistical area belongs.

Attributes	Data Type	Null?	Comment
StatArea	integer	No	A code identifying the statistical area
StatAreaNumber	smallint		A sequential number for each statistical area within a QMA.
Region	smallint		A code identifying the region in which each statistical area is located
QMA	character(4,1)		Quota Management Area code
CRA	smallint		A numerical representation of the QMA code; e.g., 3 = CRA3
AssessmentArea	character(4,1)		A code to identify which stock assessment area each statistical area is assigned to

Creator: dba

Indices: UNIQUE BTREE PrimaryKey ON (StatArea)

5.8 Table 8: TWC

Comment: This table contains the list of all possible tail widths in 1mm intervals, their converted values in 2mm intervals, and their 2mm interval values between 30 and 90mm.

Attributes	Data Type	Null?	Comment
TailWidth	integer	No	1mm intervals representing all possible tail widths
TWC	integer		Tail width converted as 2mm intervals
TWC3090	smallint		Tail width converted as 2mm Intervals between the range of 30 to 90mm

Creator: dba

Indices: UNIQUE BTREE TailWidth ON (TailWidth)

6 rlcs Logbook Business Rules

6.1 *Introduction to business rules*

The following are a list of business rules applying to the **rlcs** database. A business rule is a written statement specifying what the information system (i.e., any system that is designed to handle market sampling data) must do or how it must be structured.

There are three recognised types of business rules:

Fact	Certainty or an existence in the information system.
Formula	Calculation employed in the information system.
Validation	Constraint on a value in the information system.

Fact rules are shown on the ERD by the cardinality (e.g., one-to-many) of table relationships. Formula and Validation rules are implemented by referential constraints, range checks, and algorithms both in the database and during validation.

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

6.2 Summary of rules

Fisher details (Fishers)

FisherID	Must contain a value and that value must be an unique integer.
Surname	Must contain a value.
Town	Should be a valid gazetted geographic location.
HomePhone	Should be a valid combination of 2-digit area code and phone number.
Fax	Should be a valid combination of 2-digit area code and phone number.
MobilePhone	Should be a valid combination of 3-digit cellular provider code and phone number.
VesselPhone	Should be a valid combination of 3-digit cellular provider code and phone number.
EmailAddress	Should be a valid e-mail address in the form of <i>username@domain.name</i>
QRN	Must be a valid Quota Registration Number.

Fishing trip details (Trips)

TripNumber	Must contain a value and that value must be an unique integer.
FisherID	Must contain a value and that value must be a valid Fisher ID number as listed in the <i>Fishers</i> table.
subarea	Must be a integer between 0 and 1000.
Date	Must be a valid date from 1 April 1993.
FishingYear	Must be a valid 4-digit year greater than or equal to 1993.
VesselNumber	Must be a valid vessel number as listed in the <i>Vessels</i> table.
StatArea	Must be a valid rock lobster statistical area as listed in the <i>StatAreas</i> table.
SwellDirection	Must be a valid 2-character direction composed of: N, NE, E, SE, S, SW, W, and NW.
SwellHeight	Must be a positive integer and should be between the reasonable range of 0 to 10.
WindDirection	Must be a valid 2-character direction composed of: N, NE, E, SE, S, SW, W, and NW.
WindSpeed	Must be a positive integer and should be between the reasonable range of 0 to 60.
SoakUnits	Must be a valid 3-character code. The valid codes are: “HRS” and “DAY”.
DepthUnits	Must be a valid 3-character code. The valid codes are: “FEE”, “MET” and “FAT”.

Pot lift details (PotLifts)

PotLiftNumber	Must contain a value and that value must be an unique integer.
TripNumber	Must contain a value and that value must be a valid trip number as listed in the <i>Trips</i> table.
Pot	Must be an integer between 1 and 4 inclusive.
Zone	Must be a valid zone as listed in the <i>Zones</i> table.
SoakRaw	Must be a positive integer greater than zero and should be within the reasonable range of 1 to 240.
DepthRaw	Must be a positive integer greater than zero and should be within the reasonable range of 1 to 130.
SoakHours	Must be a positive number greater than zero and should be between the reasonable range of 1.0 to 1440.0
DepthMetres	Must be a positive number greater than zero and should be between the reasonable range of 1.0 to 200.0
LiveAbove	Must be a positive integer and should be within the reasonable range of 0 to 50.
LiveAboveKept	Must be either 0 or 1.
LiveBelow	Must be a positive integer and should be within the reasonable range of 0 to 50.
TotDead	Must be a positive integer and should be within the reasonable range of 0 to 10.
DeadAbove	Must be a positive integer less than or equal to <i>TotDead</i> and should be within the reasonable range of 0 to 50.
DeadAboveKept	Must be either 0 or 1.
DeadBelow	Must be a positive integer less than or equal to <i>TotDead</i> and should be within the reasonable range of 0 to 50.
Multiple column checks on <i>TotDead</i>, <i>DeadAbove</i> and <i>DeadBelow</i>: The sum of <i>DeadAbove</i> and <i>DeadBelow</i> should equal <i>TotDead</i> .	
Octopus	Must be a positive integer and should be within the reasonable range of 0 to 10.
PHC	Must be a positive integer and should be within the reasonable range of 0 to 10.

Pot lift details (PotLifts) (cont...)

PHCKept Must be either 0 or 1.

Other Must be a positive integer and should be within the reasonable range of 0 to 10.

Lobster details (Lobsters)

LobsterIDNumber Must contain a value and that value must be an unique integer.

PotLiftNumber Must contain a value and that value must be a valid pot lift number as listed in the *PotLifts* table.

LobsterNumber Must be a positive integer greater than zero.

Multiple column checks on *LobsterNumber* and *PotLiftNumber*:
The value of *LobsterNumber* must be unique within the *PotLiftNumber*.

Sex Must be an integer between 1 and 6 inclusive.

TailWidth Must be a positive integer and should be within the reasonable range of 10 to 140.

Kept Must be either 0 or 1.

Legal Must be either 0 or 1.

Fishing Boat details (Vessels)

VesselNumber Must contain a valid vessel registration number.

Fishing Zone details (Zones)

Zone Must contain a value and that value must be a 3-character alphabetic code.

StatArea Must be a valid statistical area code as listed in the *StatAreas* table

Statistical Area details (StatAreas)

StatArea Must contain a value and that value must be a 3-digit code within the range 901 to 943.

StatAreaNumber Must be a integer within the range 1 to 7.

Region Must be a integer within the range 1 to 3.

QMA Must be a valid rock lobster Quota Management Area code. Each code is a 4-character code: the first 3 characters are “CRA”; the last character is an integer between 1 and 9.

CRA Must be an integer between 1 and 9.

Multiple column checks on *QMA* and *CRA*:

The integer in *CRA* must equal the last character in *QMA*.

AssessmentArea Must be a valid rock lobster Quota Management Area code (e.g., “CRA3”) or one of the following stock assessment codes: “NSN”, “STEW”, or “FIOR”.

Appendix 1 : Data Entry and Validation

Trophia on behalf of the Ministry of Fisheries currently carries out data collection and validation. These data collection and validation procedures are applied directly to Trophia's **RLLBD** database. The validated data are subsequently handed to the Ministry of Fisheries Research Data Manager for storage in the **rlcs** database. The procedures listed below apply to data entry and validation for the **RLLBD** only.

Data collection

Data is collected on three forms: Logbook Datasheet; Annual Survey form; Pot Description form. Once completed, these forms and datasheets are forwarded to the appropriate technician for their area, who then forward them on to the appropriate data entry person.

Data entry

Data entry personnel are issued with a satellite copy of the database, which includes tables to store the data and forms for data entry. The data entry form includes validation code that checks the data against predetermined criteria.

Validation code

1. In the main form "fLogbook", the following fields must be entered before subforms can be entered or before they can move between records, using the navigation buttons in the top right corner of the screen: *Date*, *VesselNumber*, *FisherID*, *StatArea*, *SoakUnits*, and *DepthUnits*. If any of these are not entered Microsoft® Access brings up a message box that states which record requires an entry and sets the focus to the field mentioned.
2. In 'fsubPotLifts', the following fields must be entered before 'fsubLobsters' can be entered: *SoakRaw*, *DepthRaw* and *Zone*. If *SoakRaw* or *DepthRaw* are not entered, a message box appears that states which record requires an entry and sets the focus to the field mentioned. If *Zone* is not entered than a message box appears that asks, "You have not entered a Zone. Do you want to correct this?" If the data entry person enters "Yes" the focus will be set on the *Zone* field. If the data entry person enters "No" then they may continue with their data entry. This will be phased out once logbook participants are all entering zones (zones have only recently been introduced to the logbook programme).
3. In 'fsubPotLifts', there are also likely maximum values set for the *SoakHours* and *DepthMetres* fields, these are 600 and 300 respectively. If these values are exceeded a message box will query the user if the value is correct.
4. In 'fsubPotLifts', each pot may only be lifted once per day and therefore may only be recorded once per trip. If there is already a previous entry for the pot number for the trip being entered, then a message box will appear that will say "You have already entered this Pot Number".
5. In 'fsubLobsters', each lobster measured must have both *Sex* and *TailWidth* entered. If either are not entered when the form is updated, a message box will appear saying "Please enter a [Sex code or Tail width]" and will set the focus on the field. If the *TailWidth* is less than 30 or greater than 90mm, a message box will appear that says "Is this Rock Lobster smaller than 30mm or larger than 90mm?" and then set focus on the field if the "Yes" response is received. If the *TailWidth* is less than 20 or larger than 140mm, a message box appears that says "Rock Lobster are very unlikely to be smaller than 20mm or larger than 140mm" and this entry must be corrected before

further data can be entered. Only 25 measurements may be entered per pot, if more than this are entered, a message box appears” You may not enter more than 25 rock lobsters per pot”.

6. For all values that have likely maximum values set, there are also absolute maximums set in the underlying table as well. These are specified in the table summaries.

Data transfer to main database

The data can be sent from the data entry personnel as either an e-mail attachment or on a disk in the post. Either the entire database or just the tables can be copied and sent to Trophia Ltd.

Once Trophia Ltd. has received the data; it can be modified slightly and then entered into the main database. As there are several data entry personnel and many tables have autonumbers as primary key, there will be a problem if the data was directly entered into the database without any changes. To avoid this problem, the autonumber fields will be converted to text fields and have a prefix added, which will indicate which data entry person has sent the data. This will prevent any overlap of the primary key fields and allow for easy transfer into the main database.