

Database documentation: squ_ce

K. A. Mackay

NIWA Internal Report
2001

Revised on 30 January 2001

Contents

1	Introduction to the Database Document series	3
2	Arrow squid fishery catch effort data	3
3	Data Structures.....	4
3.1	Table relationships.....	4
3.2	Database design	7
4	Table Summaries	11
5	squ_ce Tables.....	12
5.1	Table 1: t_events.....	12
5.2	Table 2: t_fish_evnts.....	15
5.3	Table 3: t_catches	17
5.4	Table 4: t_proc_evnts	18
5.5	Table 5: t_proc_state.....	20
5.6	Table 6: t_environ.....	21
5.7	Table 7: t_versions.....	23
5.8	Table 8: t_vessels.....	23
5.9	Table 9: t_vess_reg.....	24
5.10	Table 10: t_vess_types.....	24
6	squ_ce Views.....	25
6.1	View 1: v_events	25
6.2	View 2: v_fish_evnts	26
6.3	View 3: v_catches.....	26
6.4	View 4: v_proc_evnts	27
6.5	View 5: v_environ	28
6.6	View 6: v_fish.....	29
6.7	View 7: v_proc	30
6.8	View 8: v_env.....	31
6.9	View 9: v_jigs.....	32
7	References.....	34

List of Figures

Figure 1: Entity Relationship Diagram (ERD) for the squ_ce database.....	5
Figure 2: GENSPEC diagram for the "events" entity.....	7
Figure 3: ERD of the specialisations of the entity "events".	8
Figure 4: ERD of the squid jig view incorporating the v_fish and v_proc views.	9

1 Introduction to the 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 (MFish).

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 arrow squid fishery catch effort database **squ_ce**, and is 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 how all the tables link together and their relationship with other databases.

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

Access to this database 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 Arrow squid fishery catch effort data

The New Zealand arrow squid fishery is based on two related species. *Nototodarus gouldi* is found around mainland New Zealand north of the Subtropical Convergence, where *N. sloanii* is found in and to the south of the convergence zone.

Except for the Southern Islands fishery (around Auckland and Campbell Islands), for which a separate Total Allowable Commercial Catch (TACC) is set, the two species are managed as a single fishery within an overall TACC. Jigging vessels apply most of the fishing effort on the main islands fisheries. The Southern Islands fishery is almost entirely a trawl fishery. Because the Southern Islands fishery is readily accessible to trawlers, and because squid can be caught with little finfish bycatch it is an attractive resource for trawlers.

The New Zealand squid fishery began in the late 1970s and reached a peak in the early 1980s when over 200 squid jigging vessels came to fish in the New Zealand EEZ. The discovery and exploitation of the large squid stocks in the southwest Atlantic substantially increased the supply of squid to the Asian markets causing the price to fall. In the early 1980s, Japanese squid jiggers would fish in New Zealand for a short time before continuing on the southwest Atlantic. In the late 1980s the jiggers stopped transit fishing in New Zealand and the number of jiggers fishing declined from over 200 in 1983 to around 15 in 1994. Over the last 11 years the jig catch has ranged from 9 000-54 000 t and the trawl fishery has fluctuated between about 30 000-60 000 t. A summary of the arrow squid fishery history and stock assessment can be found in Annala *et al.* (1999).

All commercial fishing catch and effort data for the squid fishery are recorded for the Quota Management System and are stored in the Ministry of Fisheries databases. The **squ_ce** database holds a research version of these data. These data are used to calculate biological reference points such the Maximum Sustainable Yield (MSY) in terms of Current Annual Yield (CAY) or Maximum Constant Yield (MCY).

The data in the **squ_ce** database are a copy of all arrow squid fishery catch and effort data from the MFish databases from 1989 to 1994 and are summarised in Gibson (1995).

3 Data Structures

3.1 Table relationships

This database contains several tables. The ERD for **squ_ce** (Figure 1) shows the logical structure¹ of the database and its entities (each entity is implemented as a database *table*) and relationships between these tables and tables in other databases. This schema is valid regardless of the database system chosen, and it can remain correct even if the Database Management System (DBMS) is changed. Each table represents an object, event, or concept in the real world that is selected to be 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 key².

Note that Figure 1 shows the main tables only. Note that most tables contain foreign keys³. These foreign keys define the relationships between the tables in **squ_ce**.

The **squ_ce** database is implemented as a relational database; i.e., 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. There are three types of relationships possible between tables, but only one exists in **squ_ce**: one-to-many⁴. These relationships can be seen in ERDs by connecting a single line (indicating “many”) from the child table; e.g., *t_events*, to the parent table; e.g., *t_vessels*, with an arrowhead (indicating “one”) pointing to the parent.

Every relationship has a mandatory or optional aspect to it. If a relationship is mandatory, then it has to occur at least once, while an optional relationship might not occur at all. For example, in Figure 1, consider that relationship between the table *t_vessels* and its child table *t_events*. The symbol “O” by the child *t_events* means that a vessel can have zero or many events records, while the bar by the parent *t_vessels* means that for every event there must be a matching vessel record.

¹ Also known as a database *schema*.

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

³ A foreign key is an attribute or a combination of attributes that is a primary key in another table.

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 vessel in *t_vessels* can have many events in *t_events* but one event can only come from one vessel.

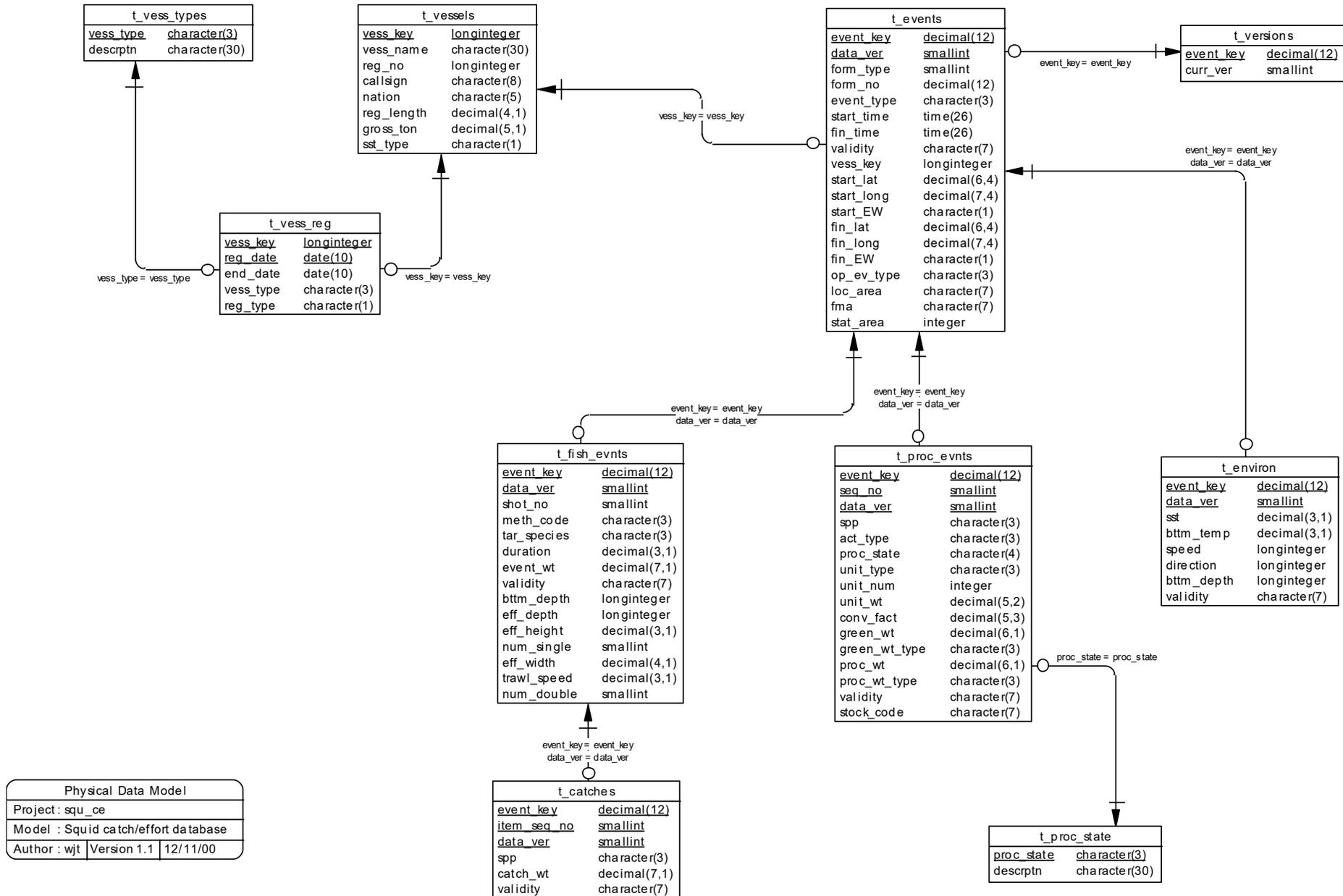


Figure 1: Entity Relationship Diagram (ERD) for the squ_ce database

These links are enforced by referential constraints⁵. 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.

Constraints are shown in the table listings by the following format:

```
Referential:      constraint name (attribute[, attribute]) |INSERT|
                                                           |DELETE|
                   parent table (attribute[, attribute])
```

Note that the typographical convention for the above format is that square brackets “[]” may contain more than one item or none at all. Items stacked between vertical lines “|” are options of which one must be chosen.

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

```
Referential:      Invalid vessel key (vess_key) INSERT
                   t_vessels (vess_key)
```

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

```
*** User Error: insert constraint "Invalid sample location" violation
```

For tables residing in external databases, the parent table name will be prefixed by the name of the database.

Section 5 lists all the **squ_ce** tables as implemented by the Empress RDBMS. 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 following format:

```
Indices:          UNIQUE index_name ON (attribute[, attribute])
```

where attribute(s) make up the primary key and the index name is the primary key name. These prevent records with duplicate keys from being inserted into the tables; e.g., a record with an existing event key.

The database listing (Tables 1-5) show that the tables also have indices on many attributes. 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[, attribute])
```

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 RDBMS default values relating to the amount of space allocated for the index.

⁵ Also known as integrity checks.

3.2 Database design

The *squ_ce* database is built around the premise that all fishing trips are based on events. Where an event is something (anything) that occurs at the particular position on earth and at a certain time onboard a vessel. In the instance of the MFish Catch and Effort system, there are three types of events: a fishing event (when a trawl or jig is made); a processing event (when an amount of fish is processed over a certain time period); and an environmental event (a weather or sea condition measurement). These three event type are specialisations of the generalise entity "events". This is modelled in the *squ_ce* database as a "GENSPEC" structure. Generalisation and specialisation are pictured in Figure 2 using a triangle containing the words "IS A TYPE OF" to connect the components to each other and to the higher-level entity. In a GENSPEC structure, attribute in common with all events are stored in the table *t_events* (Table 1), such as event start and finish date and time, start and finish position, vessel key, and statistical area. Each event, regardless of event type, is identified by an *event_key* attribute.

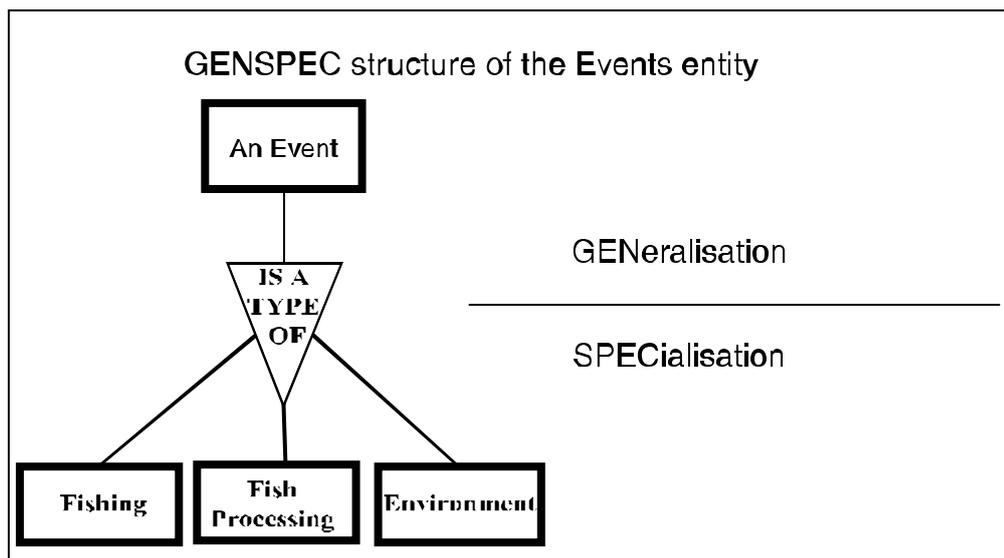


Figure 2: GENSPEC diagram for the "events" entity

Attributes specific to a fishing event are stored in the table *t_fish_evnts* (Table 2). The *event_key* attribute in *t_fish_evnts* can be used to link to the *t_events* table to get the appropriate time and position details. Details stored within *t_fish_evnts* include: target species code, fishing method code, and bottom and fishing gear depth.

The catch of fish as a result of a fishing event is stored in the table *t_catches* (Table 3). Each record in *t_catches* stored the catch weight for one species.

Fishing processing events specific attributes are stored in the table *t_proc_evnts* (Table 4). The *event_key* attribute in *t_proc_evnts* can be used to link to the *t_events* table to get the appropriate time and position details for the processing event. Fish processing events might relate to the processing of a catch from one trawl but is more likely to be the details for fish processing over a 24-hour period. Details recorded in *t_proc_evnts* include species code, processed state, number of units of processed fish, greenweight, and conversion factors (from greenweight to processed weight). Descriptions of the processed state codes can be found in the table *t_proc_stat* (Table 5).

Physical Data Model		
Project :	squ_ce	
Model :	Event views	
Author :	wtj	Version 1.1 1/13/01

v_fish	
e.event_key	decimal(12)
e.data_ver	smallint
form_type	smallint
form_no	decimal(12)
event_type	character(3)
start_time	time(26)
fin_time	time(26)
e.validity print evt_validity	
vess_key	longinteger
start_lat	decimal(6,4)
start_long	decimal(7,4)
start_EW	character(1)
fin_lat	decimal(6,4)
fin_long	decimal(7,4)
fin_EW	character(1)
op_ev_type	character(3)
loc_area	character(7)
fma	character(7)
stat_area	integer
shot_no	smallint
meth_code	character(3)
tar_species	character(3)
duration	decimal(3,1)
event_wt	decimal(7,1)
f.validity print fsh_validity	
bttm_depth	longinteger
eff_depth	longinteger
eff_height	decimal(3,1)
num_single	smallint
eff_width	decimal(4,1)
trawl_speed	decimal(3,1)
num_double	smallint
<input type="checkbox"/> t_events e	
<input type="checkbox"/> t_fish_evnts f	

v_proc	
e.event_key	decimal(12)
e.data_ver	smallint
form_type	smallint
form_no	decimal(12)
event_type	character(3)
start_time	time(26)
fin_time	time(26)
e.validity print evt_validity	
vess_key	longinteger
start_lat	decimal(6,4)
start_long	decimal(7,4)
start_EW	character(1)
fin_lat	decimal(6,4)
fin_long	decimal(7,4)
fin_EW	character(1)
op_ev_type	character(3)
loc_area	character(7)
fma	character(7)
stat_area	integer
seq_no	smallint
spp	character(3)
act_type	character(3)
proc_state	character(4)
unit_type	character(3)
unit_num	integer
unit_wt	decimal(5,2)
conv_fact	decimal(5,3)
green_wt	decimal(6,1)
green_wt_type	character(3)
proc_wt	decimal(6,1)
proc_wt_type	character(3)
p.validity print proc_validity	
stock_code	character(7)
<input type="checkbox"/> t_events e	
<input type="checkbox"/> t_proc_evnts p	

v_env	
e.event_key	decimal(12)
e.data_ver	smallint
form_type	smallint
form_no	decimal(12)
event_type	character(3)
start_time	time(26)
fin_time	time(26)
e.validity print evt_validity	
vess_key	longinteger
start_lat	decimal(6,4)
start_long	decimal(7,4)
start_EW	character(1)
fin_lat	decimal(6,4)
fin_long	decimal(7,4)
fin_EW	character(1)
op_ev_type	character(3)
loc_area	character(7)
fma	character(7)
stat_area	integer
sst	decimal(3,1)
bttm_temp	decimal(3,1)
speed	longinteger
direction	longinteger
bttm_depth	longinteger
v.validity print env_validity	
<input type="checkbox"/> t_events e	
<input type="checkbox"/> t_environs v	

Figure 3: ERD of the specialisations of the entity "events".

Environmental specific attributes are stored in the table *t_environs* (Table 6). The *event_key* attribute in *t_environs* can be used to link to the *t_events* table to get the appropriate time and position details. Such environmental variables recorded include wind speed and sea-surface temperature. For squid trawlers, these variables are recorded for the first trawl of the day; for squid jiggers, these variables are recorded at 0100 each morning.

A fundamental factor of all data derived from the MFish Catch and Effort databases is data versioning. Each record can have multiple versions ranging from the literal version to modified to incorporate error fixes or interpretations. Each record has a sequential version number associated with it, stored in the attribute *data_ver*. This number is incremented as each change is made to the record. The most current version of each record is therefore the one with the highest data version number. The current data version numbers for each event record are stored in the table *t_versions* (Table 7).

Individual vessels are identified in the *t_events* table by a vessel key. The table *t_vessels* (Table 8) stores the individual vessel details, such as vessel name, registration number, radio callsign, and gross tonnage.

Physical Data Model		
Project : squ_ce		
Model : Squid jigging view		
Author : wjt	Version 1.1	1/13/01

v_jigs	
f.event_key	print fsh_event_key
f.data_ver	print fsh_data_ver
f.form_type	
f.form_no	
f.event_type	print fsh_event_type
f.start_time	print fsh_start_time
f.fin_time	print fsh_fin_time
f.evt_validity	print fsh_evt_validity
f.vess_key	print fsh_vess_key
f.start_lat	print fsh_start_lat
f.start_long	print fsh_start_long
f.start_EW	print fsh_start_EW
f.fin_lat	print fsh_fin_lat
f.fin_long	print fsh_fin_long
f.fin_EW	print fsh_fin_EW
f.op_ev_type	print fsh_op_ev_type
f.loc_area	print fsh_loc_area
f.fma	print fsh_fma
f.stat_area	print fsh_stat_area
shot_no	
meth_code	
tar_species	
duration	
event_wt	
fsh_validity	
bttm_depth	
eff_depth	
eff_height	
num_single	
eff_width	
trawl_speed	
num_double	
p.event_key	print prc_event_key
p.data_ver	print prc_data_ver
p.event_type	print prc_event_type
p.start_time	print prc_start_time
p.fin_time	print prc_fin_time
p.evt_validity	print prc_evt_validity
p.vess_key	print prc_vess_key
p.start_lat	print prc_start_lat
p.start_long	print prc_start_long
p.start_EW	print prc_start_EW
p.fin_lat	print prc_fin_lat
p.fin_long	print prc_fin_long
p.fin_EW	print prc_fin_EW
p.op_ev_type	print prc_op_ev_type
p.loc_area	print prc_loc_area
p.fma	print prc_fma
p.stat_area	print prc_stat_area
seq_no	
spp	
act_type	
proc_state	
unit_type	
unit_num	
unit_wt	
conv_fact	
green_wt	
green_wt_type	
proc_wt	
proc_wt_type	
proc_validity	
stock_code	

Figure 4: ERD of the squid jig view incorporating the v_fish and v_proc views.

Vessels in the arrow squid fishery are registered with MFish as either a domestic, foreign licensed, or charter vessel. Over time, the registration status of a vessel may change from one to any of the other registration types. The history of vessel registration types is stored in the table *t_vess_reg* (Table 9). The registration date is usually the date that the vessel was entered in to the MFish registration system; however, this date may not always be known. As the registration date is part of the primary key, a dummy date of 1st January 1900 is entered for those vessels of unknown registration date. The type of vessel (i.e., whether the vessel is a trawler or a jigger) is stored as 3-character attribute *vess_type*. The full descriptions of the vessel type codes are stored in the table *t_vess_type* (Table 10).

In addition to the core tables in **squ_ce**, there are numerous single- and multi-table views to simplify database queries. The simplest of these views allow users to always query the most current version of the records in the main data tables *t_events*, *t_fish_evnts*, *t_catches*, *t_proc_evnts*, *t_environ* with the views *v_events* (View 1), *v_fish_evnts* (View 2), *v_catches* (View 3), *v_proc_evnts* (View 4), and *v_environ* (View 5) respectively.

The next three views allow users to utilise the GENSPEC structure efficiently. The views *v_fish* (View 6), *v_proc* (View 7), and *v_env* (View 8) join the specialised entities of *t_fish_evnts*, *t_proc_evnts*, and *t_environ* to the generalised entity *t_events* (Figure 3).

The last view *v_jigs* (View 9) joins all the tables together to allow users to easily access all data from any night's jigging from any single vessel (Figure 4).

4 Table Summaries

The following is a listing of the tables contained in the **squ_ce** database:

1. **t_events** : contains locations and times of events. Events include fishing, recording of environmental information and daily processing.
2. **t_fish_evnts** : contains effort and overall catch for each fishing event.
3. **t_catches**: contains estimated weights for each species caught. This table is used for trawling only.
4. **t_proc_evnts** : contains daily processing details of each species. Contains processing data and calculated greenweight
5. **t_proc_state** : table describing the codes for processing states used in *t_proc_evnts*.
6. **t_environ** : contains observations of environmental conditions. Time and position for each set of observations (each environmental event) can be found in *t_events*. Jigging environmental observations are made at 0100 hr, and for trawling they are made before the first shot of the day.
7. **t_versions** : contains the current version number for each event. This table can be used to speed up the extraction of data where there are joins between event tables.
8. **t_vessels**: contains details on each vessel, including identification, country of origin and size.
9. **t_vess_reg**: contains details for the types of fishing vessel for each registration period.
10. **t_vess_types**: contains the codes and descriptions for vessel types used in *t_vess_reg*.

The following is a listing of the views contained in the **squ_ce** database:

1. **v_events** : Current version of events as recorded in the *t_events* table, including fishing, recording of environmental information and daily processing events .
2. **v_fish_evnts** : Contains the latest version of effort and overall catch for each fishing event.
3. **v_catches** : Current version of catch weight records for each species caught as recorded in the *t_catches* table.
4. **v_proc_evnts** : Current version of daily processing records of each species. Contains processing data and calculated greenweight
5. **v_environ** : Latest version of observations of environmental conditions. Time and position for each set of observations (each environmental event) can be found in *t_events*. For jigging environmental observations are made at 0100 hr and for trawling they are made before the first shot of the day.
6. **v_fish** : This view joins the latest versions of the tables *t_events* and *t_fish_events*. The result is time and location data associated with data for each fishing event.
7. **v_proc** : This view joins the latest versions of the tables *t_events* and *t_proc_evnts*. The result is time and location data associated with each processing event
8. **v_env** : This view joins the current versions of the tables *t_events* and *t_environ*. The result is time and location data associated with data for each environmental event.
9. **v_jigs** : This view joins tables *t_fish_evnts*, *t_proc_evnts* and *t_environ*. It thus represents all the data from any night's jigging from a single vessel.

5 squ_ce Tables

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

5.1 Table 1: t_events

Comment: Locations and times of events. Events include fishing, recording of environmental information and daily processing. There may be several events recorded on one logbook form (E.g., there may be more than 1 trawl shot per day, as well as one processing event and one environmental event.).

Attributes	Data Type	Null?	Comment
event_key	decimal(12,0)	No	Unique identifier for an event. Each environmental, fishing and processing event for any day has a different key.
data_ver	smallint	No	Version number of data for event; e.g., 1 for first version of data as received from MFish. Each time data are changed the record containing the old data remains unchanged but a new record, containing updated data, is inserted with data_ver having a value of one greater than data_ver for the record containing the old data.
form_type	smallint		Type of form - squid jigging or trawling. This attribute is used to distinguish forms of the same number but of different types. Types are: 1 - Trawling 2 - Jigging forms from 1988 to present day 4 - Jigging forms from 1978 to 1989 19 - Jigging forms from 1989 to 1991 Note though that there is an overlap of jigging forms. The significance of these form types is the distinction between jigging and trawling.
form_no	decimal(12,0)	No	Number of form on which the event was recorded. Provides a link between environmental, fishing and processing events for each day. If more than one form is used in a day, a link will also have to be made using a date (i.e., start_time).
event_type	character(3,1)		Generic type of event - e.g., 0 for operational event such as fishing.

Attributes	Data Type	Null?	Comment
start_time	time(0) range > 19701001000000	No	Date and time at the beginning of event. For a fishing event involving a trawl shot, this is the date and time at the beginning of the shot and for a fishing event involving jigging, this is at 0100 hr as jigging is performed overnight, Format is YYYYMMDDhhmmss - e.g., 2:53:05 pm on 2 August 1994 is 19940802145305.
fin_time	time(0) range > 19701001000000		Date and time at end of a fishing event. Format is YMMDDhhmmss - e.g., 2:53:05 pm on 2 August 1994 is 19940802145305.
validity	character(7,1) match "[CENZ]"		Status flag for the data. C : Data have been checked and are ready for sending to the main database. Editing data will cause this flag to revert to an N. E : Error found during checking. Data will not be sent until errors have been cleared and data re-checked. N : Data not yet sent to the main database. Editing may occur. Z : A data entry person has flagged this record as being in error, even though the computer may not see it as so. This will suspend processing on this record until the flag is changed to something else.
vess_key	longinteger No		Unique identification number for a vessel. Refer t_vessels.
start_lat	decimal(6,4) range 0.0000 to 90.0000		Latitude of starting position for a trawl and of position at 1 am for jigging. Format is DD.dddd (e.g., 45.5000 for 45 degrees 30 minutes).
start_long	decimal(7,4) range 0.0000 to 180.0000		Longitude of starting position for a trawl and of position at 1 am for jigging. Format is DD.dddd (e.g., 45.5000 for 45 degrees 30 minutes).
start_EW	character(1,1) match "[EW]"		Hemisphere of starting position for a trawl and of position at 1 am for jigging - E for East and W for West.

Attributes	Data Type	Null?	Comment
fin_lat	decimal(6,4)		Latitude of finishing position for a trawl. Format is DD.dddd (e.g., 45.5000 for 45 degrees 30 minutes). range 0.0000 to 90.0000
fin_long	decimal(7,4)		Longitude of finishing position for a trawl. Format is DD.dddd (e.g., 45.5000 for 45 degrees 30 minutes). range 0.0000 to 180.0000
fin_EW	character(1,1)		Hemisphere of finishing position for a trawl and of position at 1 am for jigging - E for East and W for West. match "[EW]"
op_ev_type	character(3,1)	No	Type of operational event. E=Environmental F=Fishing P=Processing match "[EFP]"
loc_area	character(7,1)		Although still to be derived, this will describe the type of legal zone (e.g., EZ for within EEZ, 12 for within 12-mile limit).
fma	character(7,1)		Fisheries Management Area, although currently unavailable.
stat_area	integer		Fisheries Statistical Area, although currently unavailable.
Creator:	wjt		
Referential:	Environmental event (event_key) DELETE t_environ (event_key) Fishing event (event_key) DELETE t_fish_evnts (event_key) Processing event (event_key) DELETE t_proc_evnts (event_key) Invalid vessel name (vess_key) INSERT t_vessels (vess_key)		
Indices:	NORMAL (2, 15) BTREE i_ev_vess ON (vess_key) NORMAL (2, 15) BTREE i_evkey ON (event_key) UNIQUE BTREE i_events ON (event_key, data_ver) NORMAL (2, 15) BTREE i_form ON (form_no, form_type) NORMAL (2, 15) BTREE i_evtype ON (event_key, op_ev_type)		

5.2 Table 2: t_fish_evnts

Comment: Effort and overall catch for each fishing event.

Attributes	Data Type	Null?	Comment
event_key	decimal(12,0)	No	Unique identifier for an event. Each environmental, fishing and processing event for any day has a different key.
data_ver	smallint	No	Version number of data for event; e.g., 1 for first version of data as received from MFish. Each time data are changed the record containing the old data remains unchanged but a new record, containing updated data, is inserted with data_ver having a value of one greater than data_ver for the record containing the old data.
shot_no	smallint		Sequential number of a trawl shot within a day of fishing.
meth_code	character(3,1)		Fishing method used. Refer rdb:meth_codes.
tar_species	character(3,1)		Target species for a fishing event. For Arrow Squid this is ASQ prior to 1 Oct 1989, and SQU thereafter.
duration	decimal(3,1)		Duration of fishing event - i.e., time of finish - e.g., 1.5 means 1 hr 30 minutes. Be aware that these times are not always entered on the forms, so there can be effort and catch, but no hours given.
event_wt	decimal(7,1)		Estimated total catch in kilograms of combined species for a fishing event. This does not apply to squid caught by jigging as they are graded into trays. It will apply where squid are caught by other methods.
validity	character(7,1)		Status flag for the data. C : Data have been checked and are ready for sending to the main database. Editing data will cause this flag to revert to an N. E : Error found during checking. Data will not be sent until errors have been cleared and data re-checked. N : Data not yet sent to the main database. Editing may occur. Z : A data entry person has flagged this record as being in error, even though the computer may not see it as so. This will suspend processing on this record until the flag is changed to something else.

match "[CENZ]"

Attributes	Data Type	Null?	Comment
bttm_depth	longinteger		Depth of bottom in meters for trawl and jigging positions.
eff_depth	longinteger		For trawling the depth of the trawl. For jigging, the depth of the deepest hook for the night's fishing. The depth may vary during the night. Depth is in meters.
eff_height	decimal(3,1) range 0.0 to 85.0		Headline height for a trawl shot in meters.
num_single	smallint		Number of jigging machines in use as per the logbook form. It refers to the maximum in use at any one time for the fishing event as the number can change.
eff_width	decimal(4,1)		Gear width for a trawl shot in metres.
trawl_speed	decimal(3,1) range 1.5 to 7.0		Speed of trawling in knots.
num_double	smallint		Number of double reel machines in use. It refers to the maximum in use at any one time for a fishing event as the number can change.
Creator:	wjt		
Referential:	Event key not valid (event_key) INSERT t_events (event_key)		
	Invalid method code (meth_code) INSERT rdb : meth_codes (meth_code)		
	Invalid target species (tar_species) INSERT rdb : curr_spp (code)		
Indices:	NORMAL (2, 15) BTREE i_fishkey ON (event_key)		
	UNIQUE BTREE i_fishing ON (event_key, data_ver)		
	NORMAL (2, 15) BTREE i_fmeth ON (event_key, meth_code)		
	NORMAL (2, 15) BTREE i_fishver ON (data_ver)		

5.3 Table 3: t_catches

Comment: Estimated weights for each species caught. This table is used for trawling only.

Attributes	Data Type	Null?	Comment
event_key	decimal(12,0)	No	Unique identifier for an event. Each environmental, fishing and processing event for any day has a different key.
item_seq_no	smallint	No	Sequential number for each catch species. This is used to as part of the primary key in order to allow for null values of species.
data_ver	smallint	No	Version number of data for event; e.g., 1 for first version of data as received from MFish. Each time data are changed the record containing the old data remains unchanged but a new record, containing updated data, is inserted with data_ver having a value of one greater than data_ver for the record containing the old data.
spp	character(3,1)		Species code of species. Refer rdb:curr_spp.
catch_wt	decimal(7,1)		Weight of catch for each species in kg. For trawling, these are normally estimates. Only 5 species can be recorded on one single form.
validity	character(7,1)		Status flag for the data. C : Data have been checked and are ready for sending to the main database. Editing data will cause this flag to revert to an N. E : Error found during checking. Data will not be sent until errors have been cleared and data re-checked. N : Data not yet sent to the main database. Editing may occur. Z : A data entry person has flagged this record as being in error, even though the computer may not see it as so. This will suspend processing on this record until the flag is changed to something else.

match "[CENZ]"

Creator: wjt

Referential: Event key not valid (event_key) INSERT t_events (event_key)
Not a valid species (spp) INSERT rdb : curr_spp (code)

Indices: NORMAL (2, 15) BTREE i_catchev ON (event_key)
UNIQUE BTREE i_catch ON (event_key, item_seq_no, data_ver)

5.4 Table 4: t_proc_evnts

Comment: Daily processing details of each species. Contains processing data and calculated greenweight.

Attributes	Data Type	Null?	Comment
event_key	decimal(12,0)	No	Unique identifier for an event. Each environmental, fishing and processing event for any day has a different key.
seq_no	smallint	No	Sequential number to make a unique index for the table. For trawling each number corresponds to a different line on the daily processing summary. For jigging, there is a new number for each tray class.
data_ver	smallint	No	Version number of data for event; e.g., 1 for first version of data as received from MFish. Each time data are changed the record containing the old data remains unchanged but a new record, containing updated data, is inserted with data_ver having a value of one greater than data_ver for the record containing the old data.
spp	character(3,1)	No	3-character species code for species being processed. Refer rdb:curr_spp.
act_type	character(3,1)		Type of action - differentiates between tray tallies and weighing. Codes are: ECA - Estimated Catch (No Processed available), GRE - Greenweight, PRO - Processed, PRT - Processed Total, SIN - Single Fish, TTL - Tray Tally (number of squid per tray), TTT - Tray Total (total number of trays). match "[ECA GRE PRO PRT STN TTL TTT]"
proc_state	character(4,1)	No	4-character processing state of species being processed. Refer t_proc_state.

Attributes	Data Type	Null?	Comment
unit_type	character(3,1)		Type of unit being counted. For trawling this will be a tray and for jigging the class of tray (i.e., the range of numbers of squid per tray). Codes are: TRA - Tray T0 - Tray containing 1-10 T1 - Tray containing 11-20 T2 - Tray containing 21-30 T3 - Tray containing 31-40 T4 - Tray containing 41-50 T5 - Tray containing 51-60 T6 - Tray containing 61-70 T7 - Tray containing 71-80 T8 - Tray containing 81-90 T9 - Tray containing 91-100 T10 - Tray containing 101-150 T15 - Tray containing >150 match "T[0-9] T[R,1][A,0,5]"
unit_num	integer		Number of units of unit_type.
unit_wt	decimal(5,2)		Weight per unit.
conv_fact	decimal(5,3)		Conversion factor.
green_wt	decimal(6,1)		Green weight of species in kilograms.
green_wt_type	character(3,1)		Way in which green weight was determined. Codes are: ACT - Actual BAC - Back Calculated FBA - Fisher Back Calculated BAC is calculated by system software after logbook forms have been punched. The fisher on board a vessel calculates FBA green weight. match "[ACT BAC FBA]"
proc_wt	decimal(6,1)		This is the final processed weight for a specific processed state of fish; e.g., hapuku as HGU weight, or as FIL weight.
proc_wt_type	character(3,1)		Way in which processed weight was determined. Codes are: ACT - Actual EST - Estimated BAC - Back Calculated match "[ACT BAC EST]"

Attributes	Data Type	Null?	Comment
validity	character(7,1)		Status flag for the data. C : Data have been checked and are ready for sending to the main database. Editing data will cause this flag to revert to an N. E : Error found during checking. Data will not be sent until errors have been cleared and data re-checked. N : Data not yet sent to the main database. Editing may occur. Z : A data entry person has flagged this record as being in error, even though the computer may not see it as so. This will suspend processing on this record until the flag is changed to something else.
	match "[CENZ]"		
stock_code	character(7,1)		Code describing fish stock - i.e., area and species code. However, this is currently unavailable.
Creator:	wjt		
Referential:	Invalid processing event key (event_key) INSERT t_events (event_key) Invalid species (spp) INSERT rdb : curr_spp (code) No proc_state code (proc_state) INSERT t_proc_state (proc_state)		
Indices:	NORMAL (2, 15) BTREE i_proc_event ON (event_key) NORMAL (2, 15) BTREE i_proc_spp ON (spp) UNIQUE BTREE i_proc ON (event_key, seq_no, data_ver)		

5.5 Table 5: t_proc_state

Comment: Table describing the codes for processing states used in t_proc_evnts.

Attributes	Data Type	Null?	Comment
proc_state	character(3,1)	No	Processing state of species being processed.
descrptn	character(30,1)	No	Description of corresponding code for the processing state.
Creator:	wjt		
Indices:	UNIQUE BTREE i_proc_state ON (proc_state)		

5.6 Table 6: t_environ

Comment: Observations of environmental conditions. Time and position for each set of observations (each environmental event) can be found in t_events. Jigging environmental observations are made at 0100 hr, and for trawling they are made before the first shot of the day.

Attributes	Data Type	Null?	Comment
event_key	decimal(12,0)	No	Unique identifier for an event. Each environmental, fishing and processing event for any day has a different key.
data_ver	smallint	No	Version number of data for event; e.g., 1 for first version of data as received from MFish. Each time data are changed the record containing the old data remains unchanged but a new record, containing updated data, is inserted with data_ver having a value of one greater than data_ver for the record containing the old data.
sst	decimal(3,1) range 3.0 to 24.0		Sea surface temperature in degrees Celsius.
bttm_temp	decimal(3,1) range 0.5 to 24.0		Temperature at sea bottom or at depth of trawl, in degrees Celsius. This is only recorded in trawling.
speed	longinteger range 0 to 20		Wind speed in meters/second. Recorded for jigging only.
direction	longinteger range 0 to 360		Wind direction in degrees. Recorded for jigging only.
bttm_depth	longinteger range 40 to 1500		Bottom depth in meters. Also recorded in t_fish_events.
validity	character(7,1)		Status flag for the data. C : Data have been checked and are ready for sending to the main database. Editing data will cause this flag to revert to an N. E : Error found during checking. Data will not be sent until errors have been cleared and data re-checked. N : Data not yet sent to the main database. Editing may occur. Z : A data entry person has flagged this record as being in error, even though the computer may not see it as so. This will suspend processing on this record until the flag is changed to something else.

match "[CENZ]"

Creator: wjt
Referential: Invalid event key (event_key) INSERT t_events (event_key)
Indices: NORMAL (2, 15) BTREE i_envkey ON (event_key)
UNIQUE BTREE i_env ON (event_key, data_ver)

5.7 Table 7: t_versions

Comment: Current version for each event. This table can be used to speed up the extraction of data where there are joins between event tables.

Attributes	Data Type	Null?	Comment
event_key	decimal(12,0)	No	Unique identifier for an event. Each environmental, fishing and processing event for any day has a different key.
curr_ver	smallint	No	Version number of latest version of data.

Creator: wjt

Indices: UNIQUE BTREE i_curr ON (event_key)

5.8 Table 8: t_vessels

Comment: Information on each vessel, including identification, country of origin and size.

Attributes	Data Type	Null?	Comment
vess_key	longinteger	No	Unique identification number for a vessel.
vess_name	character(30,1)		Vessel name
reg_no	longinteger		Vessel registration number.
callsign	character(8,1)		Callsign of vessel
nation	character(5,1)		Country for fishing registration purposes.
reg_length	decimal(4,1)		Registered length of vessel in metres.
gross_ton	decimal(5,1)		Gross tonnage of a vessel.
sst_type	character(1,1)		Although currently unavailable, this will indicate whether sea surface temperature recorders take continuous or spot readings.

Creator: wjt

Referential: Invalid vessel key (vess_key) DELETE t_vess_reg (vess_key)

Indices: UNIQUE BTREE i_vessel ON (vess_key)

5.9 Table 9: t_vess_reg

Comment: Type of fishing vessel for each registration period.

Attributes	Data Type	Null?	Comment
vess_key	longinteger	No	Unique identification number for a vessel.
reg_date	date(0)	No	Date of registration. If no registration date has been entered from the registration form, this date is the date on which the registration information has been entered into the registration system. If reg_date is 19000101 then no registration date was available. This is used instead of null as reg_date is part of the primary key.
end_date	date(0)		Where a vessel has been registered more than once, this is the date of the last day before the vessel has been registered again.
vess_type	character(3,1)		Fishing type of vessel - i.e., jigger or trawler.
reg_type	character(1,1)		Type of registration. D = Domestic F = Foreign Licensed C = Charter (Joint Venture)
	match "[CDF]"		
Creator:	wjt		
Referential:	Invalid vessel (vess_key) INSERT t_vessels (vess_key)		
	Invalid vesseltype (vess_type) INSERT t_vess_types (vess_type)		
Indices:	NORMAL (2, 15) BTREE i_vreg ON (vess_key)		
	UNIQUE BTREE i_reg ON (vess_key, reg_date)		

5.10 Table 10: t_vess_types

Comment: Table describing the codes for vessel types used in t_vess_reg.

Attributes	Data Type	Null?	Comment
vess_type	character(3,1)	No	Code for vessel type
descriptn	character(30,1)		Description of the corresponding code for the vessel type
Creator:	wjt		

6 squ_ce Views

The following are listings of the single and multi-table views in the **squ_ce** database, including attribute names, data types (and any range restrictions), and comments.

6.1 View 1: v_events

Comment: Contains the current version of events, including fishing, recording of environmental information and daily processing events.

View:

```
select attr 'event_key', attr 'data_ver', attr 'form_type', attr
'form_no', attr 'event_type', attr 'start_time', attr 'fin_time',
attr 'validity', attr 'vess_key', attr 'start_lat', attr
'start_long', attr 'start_EW', attr 'fin_lat', attr 'fin_long', attr
'fin_EW', attr 'op_ev_type', attr 'loc_area', attr 'fma', attr
'stat_area' from 't_events' 'e' where (attr 'data_ver' = (select max
(attr 'data_ver') from 't_events' 'm' where (attr 'm'.'event_key' =
attr 'e'.'event_key')))
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
data_ver	smallint	No
form_type	smallint	
form_no	decimal(12,0)	No
event_type	character(3,1)	
start_time	time(0)	No
fin_time	time(0)	
validity	character(7,1)	
vess_key	longinteger	No
start_lat	decimal(6,4)	
start_long	decimal(7,4)	
start_EW	character(1,1)	
fin_lat	decimal(6,4)	
fin_long	decimal(7,4)	
fin_EW	character(1,1)	
op_ev_type	character(3,1)	No
loc_area	character(7,1)	
fma	character(7,1)	
stat_area	integer	

Creator: wjt

6.2 View 2: v_fish_evnts

Comment: Contains the latest version of effort and overall catch for each fishing event.

View:

```
select attr 'event_key', attr 'data_ver', attr 'shot_no', attr
'meth_code', attr 'tar_species', attr 'duration', attr 'event_wt',
attr 'validity', attr 'bttm_depth', attr 'eff_depth', attr
'eff_height', attr 'num_single', attr 'eff_width', attr
'trawl_speed', attr 'num_double' from 't_fish_evnts' 'f' where (attr
'data_ver' in (select max (attr 'data_ver') from 't_fish_evnts' 'm'
where (attr 'm'.'event_key' = attr 'f'.'event_key')))
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
data_ver	smallint	No
shot_no	smallint	
meth_code	character(3,1)	
tar_species	character(3,1)	
duration	decimal(3,1)	
event_wt	decimal(7,1)	
validity	character(7,1)	
bttm_depth	longinteger	
eff_depth	longinteger	
eff_height	decimal(3,1)	
num_single	smallint	
eff_width	decimal(4,1)	
trawl_speed	decimal(3,1)	
num_double	smallint	

Creator: wjt

6.3 View 3: v_catches

Comment: Current version of catch weights for each species caught.

View:

```
select attr 'event_key', attr 'item_seq_no', attr 'data_ver', attr
'spp', attr 'catch_wt', attr 'validity' from 't_catches' 'c' where
(attr 'data_ver' in (select max (attr 'data_ver') from 't_catches'
'm' where (attr 'm'.'event_key' = attr 'c'.'event_key')))
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
item_seq_no	smallint	No
data_ver	smallint	No
spp	character(3,1)	
catch_wt	decimal(7,1)	
validity	character(7,1)	

Creator: wjt

6.4 View 4: v_proc_evnts

Comment: Current version of daily processing records of each species.
Contains processing data and calculated greenweight.

View:

```
select attr 'event_key', attr 'seq_no', attr 'data_ver', attr 'spp',
attr 'act_type', attr 'proc_state', attr 'unit_type', attr
'unit_num', attr 'unit_wt', attr 'conv_fact', attr 'green_wt', attr
'green_wt_type", attr 'proc_wt', attr 'proc_wt_type', attr
"validity", attr 'stock_code' from 't_proc_evnts' 'p' where (attr
'data_ver' in (select max (attr 'data_ver') from 't_proc_evnts' 'm'
where (attr 'm'. 'event_key' = attr 'p'. 'event_key')))
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
seq_no	smallint	No
data_ver	smallint	No
spp	character(3,1)	No
act_type	character(3,1)	
proc_state	character(4,1)	No
unit_type	character(3,1)	
unit_num	integer	
unit_wt	decimal(5,2)	
conv_fact	decimal(5,3)	
green_wt	decimal(6,1)	
green_wt_type	character(3,1)	
proc_wt	decimal(6,1)	
proc_wt_type	character(3,1)	
validity	character(7,1)	
stock_code	character(7,1)	

Creator: wjt

6.5 View 5: v_environ

Comment: Latest version of observations of environmental conditions. Time and position for each set of observations (each environmental event) can be found in t_events. For jigging environmental observations are made at 0100 hr and for trawling they are made before the first shot of the day.

View:

```
select attr 'event_key', attr 'data_ver', attr 'sst', attr
'bttm_temp', attr 'speed', attr 'direction', attr 'bttm_depth', attr
'validity' from 't_environ' 'v' where (attr 'data_ver' in (select
max (attr 'data_ver') from 't_environ' 'm' where (attr
'm.'.event_key' = attr 'v.'.event_key)))View: v_environ
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
data_ver	smallint	No
sst	decimal(3,1)	
bttm_temp	decimal(3,1)	
speed	longinteger	
direction	longinteger	
bttm_depth	longinteger	
validity	character(7,1)	

Creator: wjt

6.6 View 6: v_fish

Comment: This view joins the latest versions of the tables t_events and t_fish_events. The result is time and location data associated with data for each fishing event.

View:

```
select attr 'e'.'event_key', attr 'e'.'data_ver', attr 'form_type',
attr 'form_no', attr 'event_type', attr 'start_time', attr
'fin_time', attr 'e'.'validity' print 'evt_validity', attr
'vess_key', attr 'start_lat', attr 'start_long', attr 'start_EW',
attr 'fin_lat', attr 'fin_long', attr 'fin_EW', attr 'op_ev_type',
attr 'loc_area', attr 'fma', attr 'stat_area', attr 'shot_no', attr
'meth_code', attr 'tar_species', attr 'duration', attr 'event_wt',
attr 'f'.'validity' print 'fsh_validity', attr 'bttm_depth', attr
'eff_depth', attr 'eff_height', attr 'num_single', attr 'eff_width',
attr 'trawl_speed', attr 'num_double' from 't_events' 'e',
't_fish_evnts' 'f' where (((attr 'op_ev_type' = 'F' and attr
'e'.'data_ver' = (select max (attr 'data_ver') from 't_events' 'm'
where (attr 'm'.'event_key' = attr 'e'.'event_key')))) and attr
'f'.'event_key' = attr 'e'.'event_key') and attr 'f'.'data_ver' =
attr 'e'.'data_ver'))
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
data_ver	smallint	No
form_type	smallint	
form_no	decimal(12,0)	No
event_type	character(3,1)	
start_time	time(0)	No
fin_time	time(0)	
evt_validity	character(7,1)	
vess_key	longinteger	No
start_lat	decimal(6,4)	
start_long	decimal(7,4)	
start_EW	character(1,1)	
fin_lat	decimal(6,4)	
fin_long	decimal(7,4)	
fin_EW	character(1,1)	
op_ev_type	character(3,1)	No
loc_area	character(7,1)	
fma	character(7,1)	
stat_area	integer	
shot_no	smallint	
meth_code	character(3,1)	
tar_species	character(3,1)	
duration	decimal(3,1)	
event_wt	decimal(7,1)	
fsh_validity	character(7,1)	
bttm_depth	longinteger	
eff_depth	longinteger	
eff_height	decimal(3,1)	
num_single	smallint	
eff_width	decimal(4,1)	
trawl_speed	decimal(3,1)	
num_double	smallint	

Creator: wjt

6.7 View 7: v_proc

Comment: This view joins the latest versions of the tables t_events and t_proc_evnts. The result is time and location data associated with each processing event.

View:

```
select attr 'e'.'event_key', attr 'e'.'data_ver', attr 'form_type',
attr 'form_no', attr 'event_type', attr 'start_time', attr
'fin_time', attr 'e'.'validity' print 'evt_validity', attr
'vess_key', attr 'start_lat', attr 'start_long', attr 'start_EW',
attr 'fin_lat', attr 'fin_long', attr 'fin_EW', attr 'op_ev_type',
attr 'loc_area', attr 'fma', attr 'stat_area', attr 'seq_no', attr
'spp', attr 'act_type', attr 'proc_state', attr 'unit_type', attr
'unit_num', attr 'unit_wt', attr 'conv_fact', attr 'green_wt', attr
'green_wt_type', attr 'proc_wt', attr 'proc_wt_type', attr
'p'.'validity' print 'proc_validity', attr 'stock_code' from
't_events' 'e', 't_proc_evnts' 'p' where (((attr 'op_ev_type' =
'P' and attr 'e'.'data_ver' = (select max (attr 'data_ver') from
't_events' 'm' where (attr 'm'.'event_key' = attr 'e'.'event_key'))))
and attr 'p'.'event_key' = attr 'e'.'event_key') and attr
'p'.'data_ver' = attr 'e'.'data_ver'))
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
data_ver	smallint	No
form_type	smallint	
form_no	decimal(12,0)	No
event_type	character(3,1)	
start_time	time(0)	No
fin_time	time(0)	
evt_validity	character(7,1)	
vess_key	longinteger	No
start_lat	decimal(6,4)	
start_long	decimal(7,4)	
start_EW	character(1,1)	
fin_lat	decimal(6,4)	
fin_long	decimal(7,4)	
fin_EW	character(1,1)	
op_ev_type	character(3,1)	No
loc_area	character(7,1)	
fma	character(7,1)	
stat_area	integer	
seq_no	smallint	No
spp	character(3,1)	No
act_type	character(3,1)	
proc_state	character(4,1)	No
unit_type	character(3,1)	
unit_num	integer	
unit_wt	decimal(5,2)	
conv_fact	decimal(5,3)	
green_wt	decimal(6,1)	
green_wt_type	character(3,1)	
proc_wt	decimal(6,1)	
proc_wt_type	character(3,1)	
proc_validity	character(7,1)	
stock_code	character(7,1)	

Creator: wjt

6.8 View 8: v_env

Comment: This view joins the current versions of the tables t_events and t_environ. The result is time and location data associated with data for each environmental event.

View:

```
select attr 'e'.'event_key', attr 'e'.'data_ver', attr 'form_type',
attr 'form_no', attr 'event_type', attr 'start_time', attr
'fin_time', attr 'e'.'validity' print 'evt_validity', attr
'vess_key', attr 'start_lat', attr 'start_long', attr 'start_EW',
attr 'fin_lat', attr 'fin_long', attr 'fin_EW', attr 'op_ev_type',
attr 'loc_area', attr 'fma', attr 'stat_area', attr 'sst', attr
'bttm_temp', attr 'speed', attr 'direction', attr 'bttm_depth',
attr 'v'.'validity' print 'env_validity' from 't_events' 'e',
't_environ' 'v' where (((attr 'op_ev_type' = 'E' and attr
'e'.'data_ver' = (select max (attr 'data_ver') from 't_events' 'm'
where (attr 'm'.'event_key' = attr 'e'.'event_key')))) and attr
'v'.'event_key' = attr 'e'.'event_key') and attr 'v'.'data_ver' =
attr 'e'.'data_ver'))
```

Attributes	Data Type	Null?
event_key	decimal(12,0)	No
data_ver	smallint	No
form_type	smallint	
form_no	decimal(12,0)	No
event_type	character(3,1)	
start_time	time(0)	No
fin_time	time(0)	
evt_validity	character(7,1)	
vess_key	longinteger	No
start_lat	decimal(6,4)	
start_long	decimal(7,4)	
start_EW	character(1,1)	
fin_lat	decimal(6,4)	
fin_long	decimal(7,4)	
fin_EW	character(1,1)	
op_ev_type	character(3,1)	No
loc_area	character(7,1)	
fma	character(7,1)	
stat_area	integer	
sst	decimal(3,1)	
bttm_temp	decimal(3,1)	
speed	longinteger	
direction	longinteger	
bttm_depth	longinteger	
env_validity	character(7,1)	

Creator: wjt

6.9 View 9: v_jigs

Comment: This view joins tables t_fish_evnts, t_proc_evnts and t_environ, using attributes event_key, form_no, form_type and data_ver from the table t_events. It thus represents all the data from any night's jiggling from a single vessel.

View:

```
select attr 'f'.'event_key' print 'fsh_event_key', attr
'f'.'data_ver' print 'fsh_data_ver', attr 'f'.'form_type', attr
'f'.'form_no', attr 'f'.'event_type' print 'fsh_event_type', attr
'f'.'start_time' print 'fsh_start_time', attr 'f'.'fin_time' print
'fsh_fin_time', attr 'f'.'evt_validity' print 'fsh_evt_validity',
attr 'f'.'vess_key' print 'fsh_vess_key', attr 'f'.'start_lat' print
'fsh_start_lat', attr 'f'.'start_long' print 'fsh_start_long', attr
'f'.'start_EW' print 'fsh_start_EW', attr 'f'.'fin_lat' print
'fsh_fin_lat', attr 'f'.'fin_long' print 'fsh_fin_long', attr
'f'.'fin_EW' print 'fsh_fin_EW', attr 'f'.'op_ev_type' print
'fsh_op_ev_type', attr 'f'.'loc_area' print 'fsh_loc_area', attr
'f'.'fma' print 'fsh_fma', attr 'f'.'stat_area' print
'fsh_stat_area', attr 'shot_no', attr 'meth_code', attr
'tar_species', attr 'duration', attr 'event_wt', attr
'fsh_validity', attr 'bttm_depth', attr 'eff_depth', attr
'eff_height', attr 'num_single', attr 'eff_width', attr
'trawl_speed', attr 'num_double', attr 'p'.'event_key' print
'prc_event_key', attr 'p'.'data_ver' print 'prc_data_ver', attr 'p'.'
event_type' print 'prc_event_type', attr 'p'.'start_time' print
'prc_start_time', attr 'p'.'fin_time' print 'prc_fin_time', attr
'p'.'evt_validity' print 'prc_evt_validity', attr 'p'.'vess_key'
print 'prc_vess_key', attr 'p'.'start_lat' print 'prc_start_lat',
attr 'p'.'start_long' print 'prc_start_long', attr 'p'.'start_EW'
print 'prc_start_EW', attr 'p'.'fin_lat' print 'prc_fin_lat', attr
'p'.'fin_long' print 'prc_fin_long', attr 'p'.'fin_EW' print
'prc_fin_EW', attr 'p'.'op_ev_type' print 'prc_op_ev_type', attr
'p'.'loc_area' print 'prc_loc_area', attr 'p'.'fma' print 'prc_fma',
attr 'p'.'stat_area' print 'prc_stat_area', attr 'seq_no', attr
'spp', attr 'act_type', attr 'proc_state', attr 'unit_type', attr
'unit_num', attr 'unit_wt', attr 'conv_fact', attr 'green_wt', attr
'green_wt_type', attr 'proc_wt', attr 'proc_wt_type', attr
'proc_validity', attr 'stock_code' from 'v_fish' 'f', 'v_proc' 'p'
where (((attr 'f'.'meth_code' = 'SQJ' and attr 'p'.'form_no' = attr
'f'.'form_no') and attr 'p'.'form_type' = attr 'f'.'form_type'))
```

Attributes	Data Type	Null?
fsh_event_key	decimal(12,0)	No
fsh_data_ver	smallint	No
form_type	smallint	
form_no	decimal(12,0)	No
fsh_event_type	character(3,1)	
fsh_start_time	time(0)	No
fsh_fin_time	time(0)	
fsh_evt_validity	character(7,1)	
fsh_vess_key	longinteger	No
fsh_start_lat	decimal(6,4)	
fsh_start_long	decimal(7,4)	
fsh_start_EW	character(1,1)	
fsh_fin_lat	decimal(6,4)	
fsh_fin_long	decimal(7,4)	
fsh_fin_EW	character(1,1)	
fsh_op_ev_type	character(3,1)	No

Attributes	Data Type	Null?
fsh_loc_area	character(7,1)	
fsh_fma	character(7,1)	
fsh_stat_area	integer	
shot_no	smallint	
meth_code	character(3,1)	
tar_species	character(3,1)	
duration	decimal(3,1)	
event_wt	decimal(7,1)	
fsh_validity	character(7,1)	
bttm_depth	longinteger	
eff_depth	longinteger	
eff_height	decimal(3,1)	
num_single	smallint	
eff_width	decimal(4,1)	
trawl_speed	decimal(3,1)	
num_double	smallint	
prc_event_key	decimal(12,0)	No
prc_data_ver	smallint	No
prc_event_type	character(3,1)	
prc_start_time	time(0)	No
prc_fin_time	time(0)	
prc_evt_validity	character(7,1)	
prc_vess_key	longinteger	No
prc_start_lat	decimal(6,4)	
prc_start_long	decimal(7,4)	
prc_start_EW	character(1,1)	
prc_fin_lat	decimal(6,4)	
prc_fin_long	decimal(7,4)	
prc_fin_EW	character(1,1)	
prc_op_ev_type	character(3,1)	No
prc_loc_area	character(7,1)	
prc_fma	character(7,1)	
prc_stat_area	integer	
seq_no	smallint	No
spp	character(3,1)	No
act_type	character(3,1)	
proc_state	character(4,1)	No
unit_type	character(3,1)	
unit_num	integer	
unit_wt	decimal(5,2)	
conv_fact	decimal(5,3)	
green_wt	decimal(6,1)	
green_wt_type	character(3,1)	
proc_wt	decimal(6,1)	
proc_wt_type	character(3,1)	
proc_validity	character(7,1)	
stock_code	character(7,1)	

Creator: wjt

7 References

1. Annala, J.H., Sullivan, K.J., and O'Brien, C.J. (*comps.*): 1999. Report from the Fishery Assessment Plenary, April 1999: stock assessments and yield estimates. pp. 35-38.
2. Gibson, D.J.M. 1995: The New Zealand Squid Fishery, 1979-93. *MAF Fisheries Technical Report No 42*. 43p.