Database documentation
for
Deepwater commercial database

dw_cdb

David Fisher
based on Draft documentation by Ralph Coburn

NIWA Fisheries Data Management
Database Documentation Series

Postgres Version 2.1 July 2015

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 for Primary Industries (MPI) formerly the Ministry of Fisheries.

This MPI data set, incorporates historic research data, data collected by MAF Fisheries prior to the split in 1995 of Policy to the Ministry of Fisheries and research to NIWA, and data collected by NIWA and other agencies for the Ministry of Fisheries and subsequently for MPI.

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

2 The deepwater commercial database

2.1 Data sources

The data is sourced from fishing returns completed by commercial fishers, with the early year's data being from the original Fisheries Statistics Unit dataset. This time series starts in April 1978. Subsequent data have been sourced from the MFish and subsequently MPI commercial catch effort database, Warehou.

3 Data structures

The **dw_cdb** database has one main table, *t_all*. It contains tow-by-tow records of commercial fishing where orange roughy or oreo were targeted or caught. Each record has details of the vessel, fishing details including date and position, and catch details of orange roughy or oreo. Oreo may be unspecified (OEO), smooth oreo (SSO) or black oreo (BOE).

The only link is between *t_all* and the master species code table, species_master, on the **rdb** database.

The *t_all* table is the result of a decision to denormalise two earlier tables, t_tow and t_vess to aid query speed for users. *t_all* now supercedes both of these.

This documentation supersedes any previous documentation, including Coburn 1995[1].

[1] Coburn, R. 1995. Introduction to deepwater commercial database (**dw_cdb**). Unpublished MAF Fisheries internal report.

4 Table summaries

The **dw_cdb** database has one official table, *t_all*.

The following is a listing and brief outline of the tables contained in dw_cdb.

1. **t_all**: Records of tows where orange roughy (ORH) or Oreo (OEO, SSO, or BOE) were either targeted or caught.

5 dw_cdb Tables

The following is the only table within the dw-cdb including attribute names, data types (and any range restrictions), and comments.

5.1 Table 1: t_all

Comment: Records of tows where orange roughy (ORH) or Oreo were either targeted or caught. Oreo may be unspecified (OEO), or smooth oreo (SSO) or black oreo (BOE).

Column	Type	Null?	Description
vessel_id	smallint		Vessel identification number used in this database. Used to join to other tables such as t_vess.
vessel_key	smallint		MPI (MFish) assigned number to uniquely identify the vessel.
nation	character varying(3)		Code for vessels home nation: DOM New Zealand, SOV=RUS USSR and its derivatives, JAP Japan, KOR Korea, FRG Federal Republic of Germany, POL Poland, NOR Norway, FAR Faroe Island, etc.
tonnage	smallint		Gross registered tonnage (GRT) of vessel, tons
kw	smallint		Main engine power of vessel, Kilowatts.
type	character varying(7)		Indicates data source: old, fsu, dw, tcepr, updat92, updat93, updat94, updat95, etc.
date_s	date	No	Date at the start of tow.
time_s	smallint		Time at start of tow, 24hr clock.
depth	smallint		Depth under vessel at start of tow, meters.
speed	numeric(2,1)		Speed of vessel during the tow, knots.
target_sp	character(3)		Target species recorded for this tow.
lat_s	integer		Latitude at start of tow, degrees and decimal minutes, one implied decimal.

long_s	integer	Longitude at start of tow, degrees and decimal minutes, one implied decimal.
eorw_s	character varying(1)	Specifies long_s as east or west.
time_f	smallint	Time at end of tow, 24hr clock.
lat_f	integer	Latitude at end of tow, degrees and decimal minutes, one implied decimal.
long_f	integer	Longitude at end of tow, degrees and decimal minutes, one implied decimal.
eorw_f	character varying(1)	Specifies long_f as east or west.
orh_kg	integer	Catch of orange roughy, kgs.
oeo_kg	integer	Catch of unspecified oreo, kgs.
sso_kg	integer	Catch of smooth oreo, kgs.
boe_kg	integer	Catch of black oreo, kgs.
dlat_s	numeric(7,5)	Latitude of vessel at start of the station in decimal degree.
dlon_s	numeric(8,5)	Longitude of vessel at start of the station in decimal degree.
dlat_e	numeric(7,5)	Latitude of vessel at end of the station in decimal degree.
dlon_e	numeric(8,5)	Longitude of vessel at end of the station in decimal degree.
startp	geometry	Position of vessel at start of the station as gis point type.
endp	geometry	Position of vessel at end of the station as gis point type.
track	geometry	Track line of vessel from start position to end position of station as gis line type.

Indexes:
   "nx_t_all_date_s" btree (date_s)
   "nx_t_all_endp" gist (endp)
   "nx_t_all_lat_s" btree (lat_s)
   "nx_t_all_long_s" btree (long_s)
   "nx_t_all_nation" btree (nation)
   "nx_t_all_startp" gist (startp)

```
    "nx_t_all_target_sp" btree (target_sp)
    "nx_t_all_track" gist (track)
    "nx_t_all_vessel_id" btree (vessel_id)
    "nx_t_all_vessel_key" btree (vessel_key)
Check constraints:
    "enforce_dims_endp" CHECK (ndims(endp) = 2)
    "enforce_dims_startp" CHECK (ndims(startp) = 2)
    "enforce_dims_track" CHECK (ndims(track) = 2)
    "enforce_geotype_endp" CHECK (geometrytype(endp) = 'POINT'::text
     OR endp IS NULL)
    "enforce_geotype_startp" CHECK (geometrytype(startp) = 'POINT'::text
     OR startp IS NULL)
    "enforce_geotype_track" CHECK (geometrytype(track) =
     'LINESTRING'::text OR track IS NULL)
    "enforce_srid_endp" CHECK (srid(endp) = 4326)
    "enforce_srid_startp" CHECK (srid(startp) = 4326)
    "enforce_srid_track" CHECK (srid(track) = 4326)

Foreign-key constraints:
    "fk_t_all_species_master" FOREIGN KEY (target_sp)
     REFERENCES rdb.species_master(code)
     ON UPDATE CASCADE ON DELETE CASCADE
```