Database documentation for Ministry for Primary Industries

‘seamount’ database

NIWA Fisheries Data Management, Database Documentation Series

Version 1.0 September 2023

Climate, Freshwater & Ocean Science

Prepared by:
B Wood, M R Clark, S Carswell, L Hayward

For any information regarding this document please contact:
Lydia Hayward
Manager – Data Services
Fisheries Data Services
fisheriesdataservices@niwa.co.nz

National Institute of Water & Atmospheric Research Ltd
Private Bag 14901
Kilbirnie
Wellington 6241

Phone +64 4 386 0300

Revision	Description	Date
Version 0.1	Draft in preparation/in review	June 2023
Version 1.0	Polished ERDs, code examples	September 2023

Contents

Figures

Database documentation series

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

This document introduces the database **seamount** and is a part of the database documentation series produced by NIWA.

All documents in this series include an introduction to the database design, a description of the main data structures accompanied by an Entity Relationship Diagram (ERD), and a listing of all the main tables. The ERD graphically shows the database tables and their relationships to each other.

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

The data managed in the database, and the processes by which they were derived is described in Clark et al, 2022 (AEBR 291). This report also describes the nature of seamounts and other underwater features which is not repeated here. It is strongly recommended that users of the data familiarise themselves with this report. The report can be found at the following link:

AEBR 291 Underwater Topographic Features in the New Zealand region

This documentation replaces a previous Seamount database held by FNZ which included different tables and columns (see Rowden et al. 2008) and its documentation in Mackay (2007).

Seamount database

This document describes the seamount database developed in 2022, which provides an updated and enhanced database, replacing the 2006 version of the database. It specifically covers the instance of the database provided for access by MPI staff, not NIWA's internal version. The MPI version is automatically refreshed nightly from NIWA's version, including the data from external databases.

The term seamount has various interpretations (Clark et al. 2022), but in the context of this database it refers to UTF's (Underwater Topographic Features) which are over 100m high from the base to the peak. The generic abbreviation UTF is used in this document, and includes seamounts, knolls and hills.

PostgreSQL is the RDBMS (Relational Database Management System) used to implement this database. The PostGIS extension v3.3.2 is used to provide comprehensive spatial data management and query capabilities. The ogr_fdw v1.1 and postgres_fdw v1.1 provide access to remote data sources and allow their presentation as managed Postgres/PostGIS tables, allowing for joins between separately managed datasets irrespective of legacy systems.

Typical database extracts of seamount data are used in conjunction with a wide range of data from other sources, such as:

- political, legal or fishery management boundaries
- various environmental classification areas
- commercial fishing activity
- specimen collections
- research trawl activity and catches

To facilitate such use of UTF data this database includes several related datasets, including research trawl station and catch data as well as specimen records from the NIWA invertebrate collection (NIC). This approach enables the use of SQL queries to directly relate these data to UTF's instead of requiring the analysis of multiple data extracts from various repositories or databases. These remote data are essentially provided live, so local SQL queries will always access current records from these remote datasets. Examples of SQL queries linking UTF data with NIC and research trawl data are provided in Appendix 1.

The seamount database can also be directly accessed by GIS (Geographic Information Software) applications. To support such access, various relevant map related datasets are included in the database, so users do not need to access data from multiple data sources when visualising or querying the data. Some examples using QGIS to access the seamount database are included in Appendix 2.

Functional overview

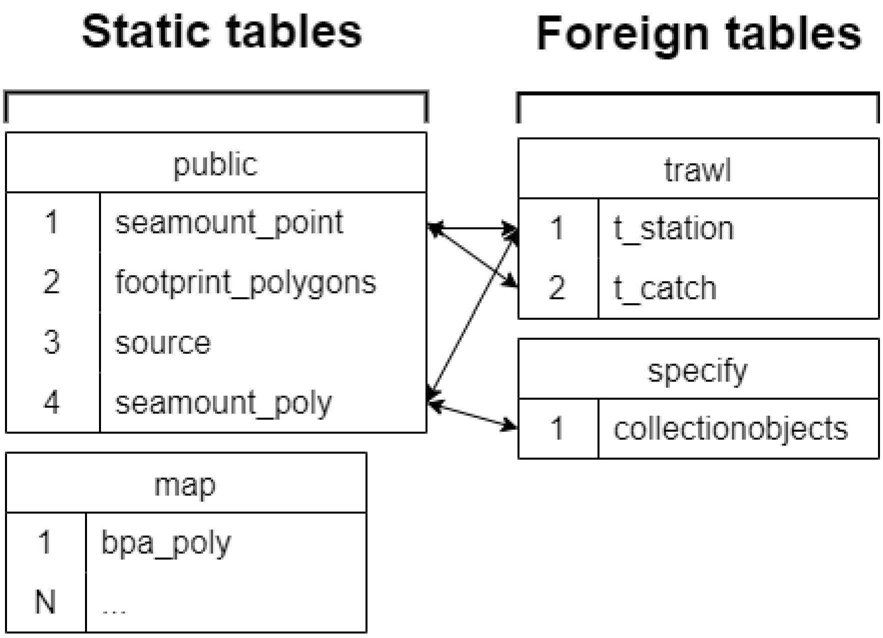

Figure 1: A functional overview of the schemas within the seamount database, including example interfaces between tables.

Database structures

Table relationships

The database comprises tables (representing real world entities), each with columns (storing attribute information describing the entities). Some tables have explicit relationships. These tables are shown diagrammatically in an ERD, Figure 1.

Other tables have been provided for convenience or as layers to be used in GIS applications, such as those within the “map” schema (derived from LINZ), or those within the public schemas (mpi_eez_poly & mpi_bpa_poly), supplied as requested for MPI’s use (see Figure 2).

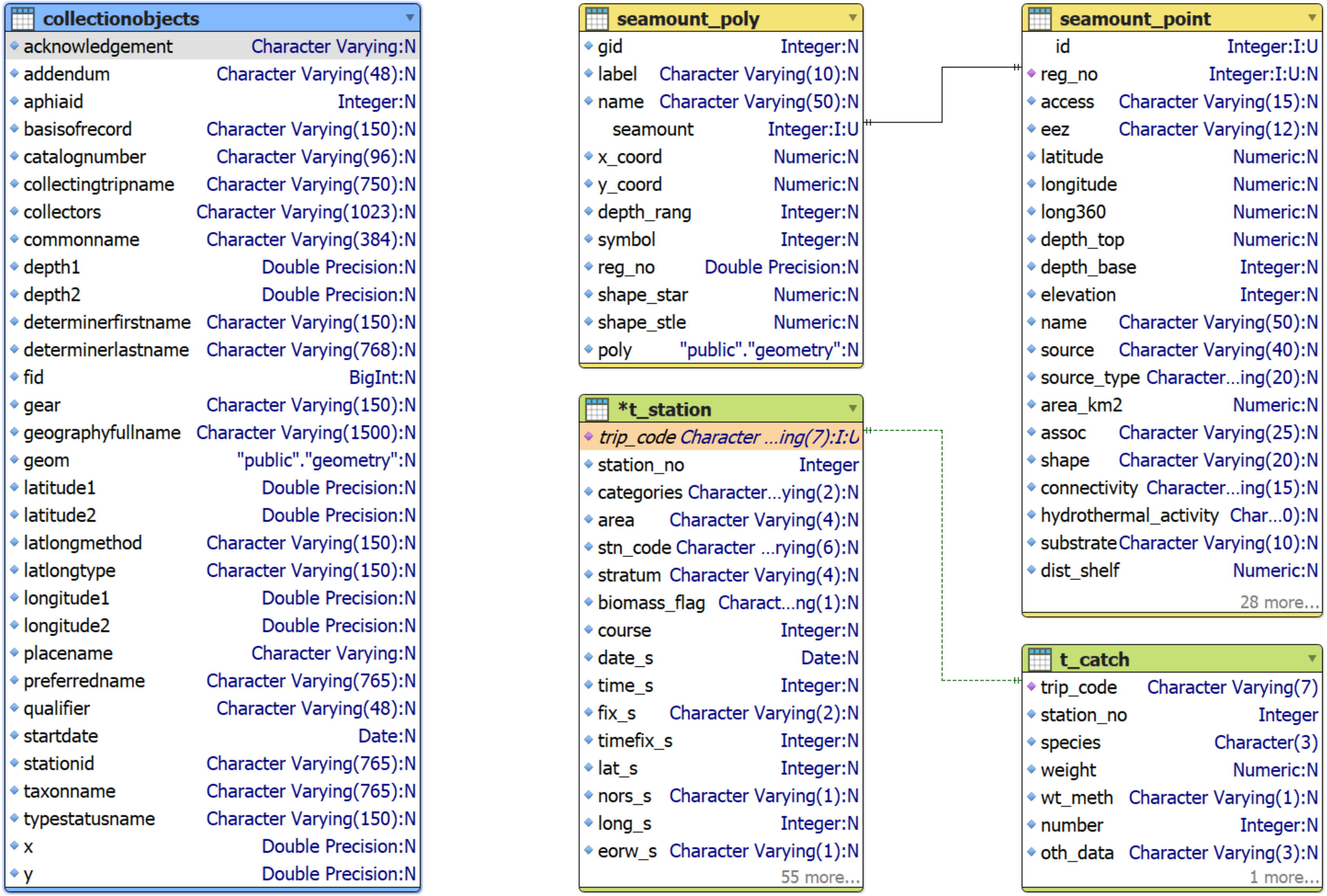

Figure 2: Entity Relationship Diagram (ERD) for seamount_point, seamount_poly, trawl station, trawl catch & specify collection objects tables.

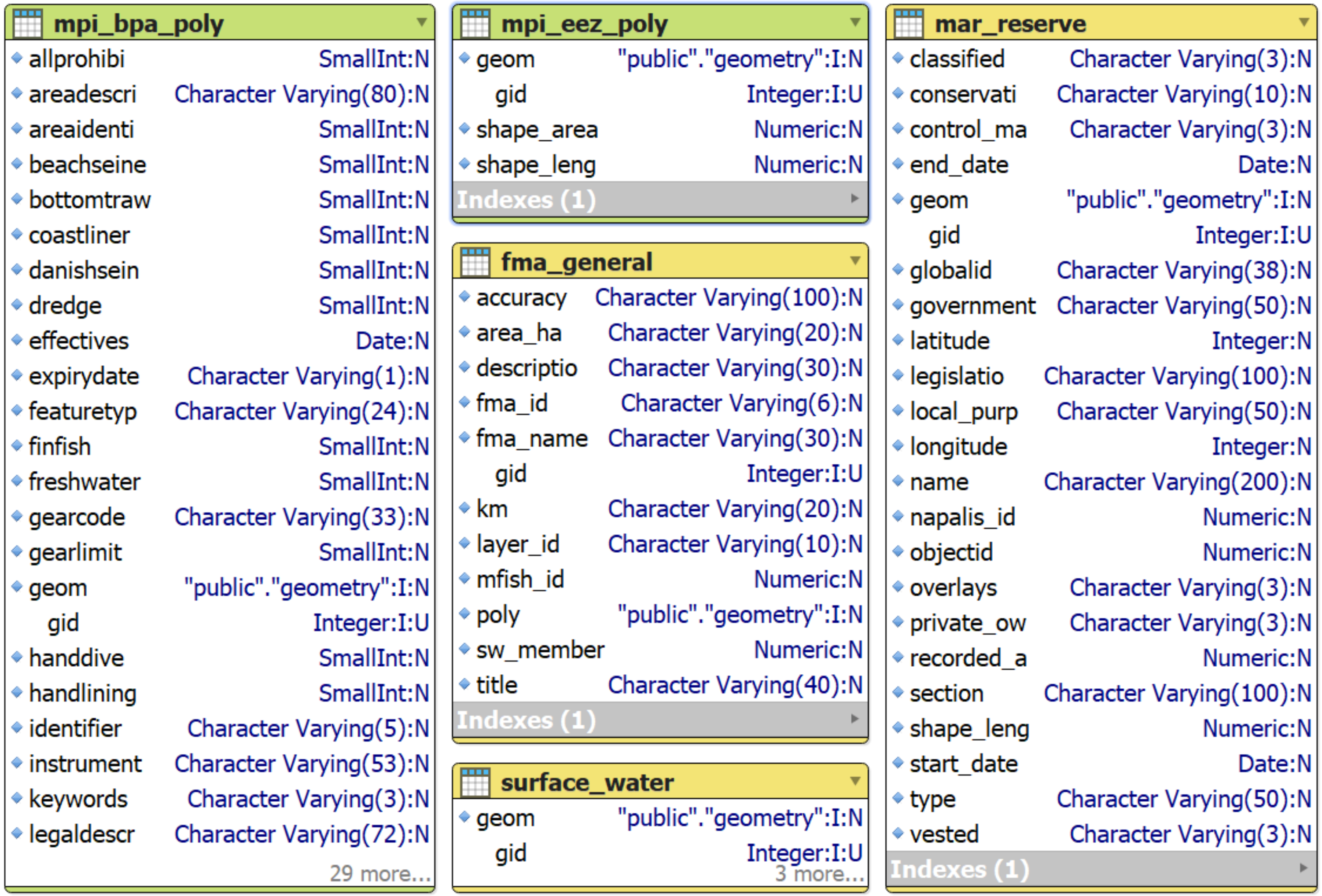

Figure 3: ERD of various "map" schema & MPI convenience map tables.

Table summaries

This database has tables containing seamount point and polygon data, map schemas and foreign tables from the ‘trawl’ and Specify ‘niwainvert’ databases.

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

1. **seamount_point:** This is the primary table of the database. It stores the core information describing individual seamounts
2. **footprint_polygons:** contains commercial trawl footprint polygons, adjusted from the officially reported trawl positions (the vessel positions) to represent the likely footprint of the trawl gear on the seabed.
3. **seamount_poly:** contains seamount polygon data.
4. **source:** Lookup table defining the various source datasets from which each seamount record originates
5. **map schemas:** Multiple tables present in the map schema that provide various spatial context

    5.1 **public.mpi_bpa_poly:** Benthic Protection Areas as defined by Geospatial Management MPI (https://data-mpi.opendata.arcgis.com/datasets/MPI::benthic-protection-areas/) (data updated 28 Jan 2020)

    5.2 **public.mpi_eez_poly:** Table created from r220218_EEZ_BT_Forum\Spatial\Data\EEZ\NZ_EEZ_MPI_2023.shp, provided by MPI. EEZ and TS outer limit line datasets were sourced from the LINZ data service. Converted to polygons then the TS was removed from the EEZ (MPI 2023)

    5.3 **map.fma_general:** Fisheries Management Areas defined using the approximate co-ordinates within the Fisheries Act, 1996.

    5.4 **map.mar_reserve:** Marine areas or reserves defined by the Marine Reserves Act.

Foreign tables: ‘trawl’ database

6. **t_station:** please refer to ‘trawl’ database documentation.
7. **t_catch:** please refer to ‘trawl’ database documentation.

Foreign tables: Specify ‘niwainvert’ database

8. **collectionobjects:** station, voyage, depth and taxonomy information on specimens held at the NIWA Invertebrate Collection (NIC).

Database Tables

The following are listings of the tables in seamount including attribute names, data types and descriptions.

Table 1: seamount_point

The main table is the seamount point table. This stores the core information describing individual seamounts.

Comment: Table to store seamount (UTF) data. Originally imported from a spreadsheet maintained by NIWA (refer Malcolm Clark)'

Column name	Type	Description
id	integer	DB primary key
reg_no	integer	Unique integer identifying an individual seamount
access	character varying	The status of the data for public release. Where defined as "Good" the record is served to MPI/FNZ and made public. "Sensitive" is where the information has been provided by the fishing industry and may be commercially sensitive or where permission has not been granted for data release.
eez	character varying	Identifies if the peak of a UTF lies within the NZ Territorial Sea, the NZ EEZ or is outside the EEZ.
latitude	numeric	The latitude (in decimal degrees) of the peak of the UTF.
longitude	numeric	The longitude (in decimal degrees, +-180) of the peak of the UTF.
long360	numeric	The longitude (in decimal degrees, 0-360) of the peak of the UTF.
depth_top	numeric	The depth (m) of the shallowest part of the peak of the UTF.
depth_base	integer	The depth (m) of the base of the UTF.
elevation	integer	The height (m) from the base to the top of the UTF.
name	character varying	The agreed common name (if any) of the UTF.
source	character varying	The survey or original dataset where the information about the UTF was obtained. From a list defined in the table "source".
source_type	character varying	
area_km2	numeric	The area (sq km) occupied by the UTF basal polygon

Column name	Type	Description
assoc	character varying	The feature type associated with the UTF.
shape	character varying	The shape of the profile of the UTF.
connectivity	character varying	The nature of the UTF with respect to any neighbouring UTF's.
hydrothermal_activity	character varying	Is the UTF hydrothermally active?
substrate	character varying	The existence of substrate data held by NIWA
dist_shelf	numeric	The distance (km) from the UTF to the New Zealand continental shelf.
surf_water	character varying	The nature of the ocean surface water above the UTF.
mec_20	integer	The MEC20 classification for the area the UTF lies within.
mec_33	integer	The MEC33 classification for the area the UTF lies within.
bomec	integer	The BOMEC classification for the area the UTF lies within.
scc	integer	The Seafloor Community Classification for the area the UTF lies within.
curr_speed	numeric	The typical speed of the ocean currents for the water above the UTF.
depth_thermocline	numeric	The typical depth of the thermocline in the water above the UTF.
ann_mean_semi_diur_tide	numeric	Mean semi-diurnal tidal flow (used in calculating meanflow Taylor Cap probability
diurnal_tide	numeric	Mean diurnal tidal flow (used in calculating diurnal Taylor Cap probability)
prob_cap_meanflow	numeric	Probability of Taylor Cap formation in mean current flow
prob_cap_diurnal	numeric	Probability of Taylor Cap formation in mean tidal flow
mean_slope	numeric	The average slope of the sides of the UTF. Calculated by taking the average slope of 8 lines run from the peak to the UTF base at 45° compass intervals.
calc_area	numeric	The calculated area (sqkm) inside the perimeter of the base of the UTF
perimeter	numeric	Perimeter distance (km) of the base of the seamount
nearest_neighbour	integer	The reg_no of the nearest other UTF
dist_nn	numeric	The distance (km) to the nearest neighbouring UTF.

Column name	Type	Description
num_in_100km	integer	The number of other UTF's within 100km of the peak.
bpa	character varying	Is the UTF within a Benthic Protection Area? Calculated using "map".bpa_poly
sca	character varying	Is the UTF within a Seamount Closure Area? Calculated using "map".sca_poly
mpa	character varying	Is the UTF within a Marine Protected Area? Calculated using "map".mar_reserve
geom	geometry (Point,4326)	Seamount summit point data (EPSG:4326), refer to AEBR 291
fished	character	Intersection of trawl footprint and basal polygon
utf_type	character varying	Terminology based on elevation (Seamount, Knoll, Hill)
fma	character varying	Fisheries Management Area
biol_trawl	character varying	Biology trawled

Table 2: footprint_polygons

The second table generated specifically for the seamount database (as described in Clark et al. 2022) contains commercial trawl footprint polygons, adjusted from the officially reported trawl positions (the vessel positions) to represent the likely footprint of the trawl gear on the seabed. This should be used with caution when estimating fished areas relating to UTF's, as they can extend beyond the perimeter of the UTF. For such analyses the footprint polygons should be clipped to the UTF polygons so only the areas inside the UTF perimeter are included. An example SQL carrying this out is provided in Appendix 1

Comment: Table storing adjusted trawl footprints and basic related catch and effort data for trawls in the vicinity of a known UTF. Data generated NIWA (refer Owen Anderson)

Column	Type	Description
gid	integer	unique record identifier generated during the shapefile ingestion process.
objecti	integer	A key to enable matching of records to the raw footprint source data
rownum	integer	A key to enable matching of records to the processed footprint data
fshngek	integer	Fishing event key, matches event_key in FNZ Warehou, etc. catch-effort databases
clnnwcn	character varying	Client name with Client No. Permit holder matching client_name and client_no in FNZ Warehou etc
twdst_m	numeric	Distance along the seabed travelled by the trawl gear, from recorded start and finish positions (m)
fyear	numeric	Fishing year based on recorded start date of the trawl
strtdtt	date	Recorded start date of trawl
target	character varying	Recorded target species for the trawl (limited to the main deep-sea fishery targets ORH, OEO, BOE, SSO, SOR, BYX, BYS, CDL and EPT)
trwl_wd	numeric	Effective width of trawl contact on the seafloor, based on figures for separate gear components reported in Mormede et al. 2017 for UTF-type trawls, adjusted according to the percentage of time each component of the trawl is estimated to be in contact with the seafloor.
ttlctcw	numeric	Total reported catch weight for the trawl (kg)
long_sn	numeric	Longitude at the start of the trawl, adjusted for the offset between the vessel and trawl using the trawl direction obtained from start and finish positions,

Column	Type	Description
		the recorded depth, and an assumed warp:depth ratio
lat_sn	numeric	Latitude at the start of the trawl, adjusted for the offset between the vessel and trawl using the trawl direction obtained from start and finish positions, the recorded depth, and an assumed warp:depth ratio
long_fn	numeric	Longitude at the end of the trawl, adjusted for the offset between the vessel and trawl using the trawl direction obtained from start and finish positions, the recorded depth, and an assumed warp:depth ratio
lat_fn	numeric	Latitude at the end of the trawl, adjusted for the offset between the vessel and trawl using the trawl direction obtained from start and finish positions, the recorded depth, and an assumed warp:depth ratio
poly	geometry	Geometry of trawl footprint polygon (based on start and end positions, and trawl width) projected in Mercator 41 (EPSG:3994)

Table 3: seamount_poly

Table to store seamount (UTF) polygon data. Originally imported from a shapefile maintained by NIWA (refer Kevin Mackay).

This table is currently provided 'as is' with updates forthcoming for usability and interoperability. To elaborate, currently some global IDs (GIDs) are associated with multi-polygon features, rather than individual polygons. Peaks can lie within the features of a larger seamount (e.g. complicated situations involving a second UTF upon a first UTF) that leads to the relationship between the seamount_point reg_no field & the seamount_poly seamount field not working as well as it should.

This table's primary key, seamount, can be joined to the seamount_point table via the reg_no attribute. See the "Querying NIC specimen data" example in Appendix 1.

Comment: Table to store seamount (UTF) polygon data. Originally imported from a shapefile maintained by NIWA (refer Kevin Mackay)

Column	Type	Description
gid	integer	Internal unique polygon identifier. Note: this is not the same as a UTF identifier
label	text	Text label to be used when plotting the polygon on a map (NO LONGER USED)
name	text	Recorded name of the UTF
seamount	integer	Seamount identifier allowing linking to the seamount_point table
x_coord	float	X coordinate of the centroid of the polygon (NO LONGER USED)
y_coord	float	Y coordinate of the centroid of the polygon (NO LONGER USED)
depth_rang	integer	Range of depth (metres) between the lowest and highest contours (NO LONGER USED)
symbol		Symbol to be used when plotting the polygon on a map (NO LONGER USED)
reg_no	integer	Unique identifier for UTF (NO LONGER USED - see column seamount)
shape_star	float	Area (square metres) of polygon
shape_stle	float	Length (metres) of perimeter of polygon
poly	geometry(MultiPolygon,3994)	Seamount Multicolour geometry (ESPG:3994) as derived from a NIWA managed shapefile

Table 4: source

This table is referenced by the main seamount table to provide a set of records describing where the information pertaining to each UTF was originally derived

Comment: Lookup table defining the various source datasets from which each seamount record originates.

Column	Type	Description
id	integer	database primary key for source table
name	character varying	the name of the source dataset for the UTF.
description	text	if present, a verbose description of the source dataset.

The map schemas

There are several tables present in the map schema within the seamount database to provide spatial context. They can be used to:

- provide a lookup capability to determine which region in a map (table) each UTF lies within
- provide contextual map layers for a GIS application rendering maps showing UTF points and/or polygons

The columns in each table are those found in the original dataset, and are not documented here. Each map table has a comment assigned describing the source of the map data.

Three additional tables within the public schema have been provided as convenience layers:

- mpi_bpa_poly, Benthic Protection Areas as definied by Geospatial Management MPI (https://data-mpi.opendata.arcgis.com/datasets/MPI::benthic-protection-areas/) (data updated 28 Jan 2020)
- mpi_eez_poly, Table created from r220218_EEZ_BT_Forum\Spatial\Data\EEZ\NZ_EEZ_MPI_2023.shp, provided by MPI. EEZ and TS outer limit line datasets were sourced from the LINZ data service. Converted to polygons then the TS was removed from the EEZ (MPI 2023)
- linz_12mile_nzts_poly, 12 Mile Territorial Sea Outer Limit as definied by ISO 19115/19139, File Identifier 8de93723-46b5-a99e-183f-bb8448ce75e7

Table	Comment
“map”.bpa_poly	NZ Benthic Protected Area polygons
“map”.eez	EEZ polygon from LINZ, as at 21/11/2021. See data.linz.govt.nz for full metadata.
“map”.fma_general	NZ Fisheries Management Areas polygons, from data.govt.nz
“map”.mar_reserve	NZ marine reserves (type 1 marine protected areas), from data.govt.nz
“map”.mec20	NZ Marine Environment Classification 20 class polygons
“map”.nz_coast	1:50 scale NZ coast & islands polygons, as at 21/11/2021from data.linz.govt.nz
“map”.nzts_poly	NZ Territorial Sea polygons supplied by LINZ as a custom dataset. Contains overlapping polygons derived from coastlines at different scales.
“map”.sca_poly	NZ Seamount Closure Area polygons
“map”.scc	NZ Seafloor Community Classification polygons derived from the raster original (Stevenson, 2021)
“map”.surface_water	Surface water zones around New Zealand. Generalised from NIWA data.

Foreign tables: 'trawl' database

The seamount database includes two tables from the trawl schema: t_catch and t_station. The trawl schema holds trawl survey data, collected by research trawl surveys on both research vessels and chartered commercial fishing vessels.

trawl.t_station

Every trawl survey will have at least one station – a station is the location at which the trawl gear was towed. The table trawl.t_station holds the station details, such as start and finish location, time, depth, gear performance and environmental parameters.

trawl.t_catch

Each station in a trawl survey may produce a catch of several species of fish, and other organisms. A catch from any one station is broken down into the different species, with each species being an individual record in the table trawl.t_catch. Note that not every station may produce a catch, so there is an optional one-to-many relationship between trawl.t_station and trawl.t_catch.

Further trawl documentation

For additional details on trawl survey data and specific information for the trawl.t_catch & trawl.t_station tables, please refer to the 'trawl' database documentation:

Data Holdings Scientific Research Databases - Trawl.pdf

Foreign tables: Specify 'niwainvert' database

The Specify 'niwainvert' relational database holds multiple tables of information about specimens held at the NIWA Invertebrate Collection (NIC). Relevant to the seamount database, a table 'collectionobjects' within the specify schema, collates information about specimens, including taxonomic information, collection and locality information (station, voyage and depth information about where specimens are collected).

The primary columns from the 'specify.collectionobjects' table are presented in the table below.

Column	Type	Description
collectingtripname	character varying	database primary key for source table
stationid	character varying	the name of the source dataset for the UTF.

Acknowledgements

The authors would like to acknowledge Kevin Mackay, Owen Anderson, Alan Hart & Graham Rickard, who contributed data and files to this database and Shaun Carswell for establishing the database pipeline.

References

Clark, M.R., Wood, B., Mackay, K., Anderson, O.F., Hart, A., Rickard, G., Rowden, A.A. (2022) Underwater Topographic Features in the New Zealand region: development of an updated 'SEAMOUNT' database and information on the extent and intensity of deep-sea trawl fisheries on them. *New Zealand Aquatic Environment and Biodiversity Report No. 291*. 28 p.

Mackay, K.A. (2007) Database documentation: seamount. NIWA Internal Report.

Rowden, A.A., Oliver, M., Clark, M.R., Mackay, K. (2008) New Zealand's "SEAMOUNT" database: recent updates and its potential use for ecological risk assessment. *New Zealand Aquatic Environment and Biodiversity Report No. 27*. 49 p.

Appendix A Example SQL queries to retrieve data from the seamount database

Introduction

A modern database is not simply a place to store and retrieve data. It can be a powerful analytical tool in its own right and can be accessed directly from other software applications, such as GIS and statistical or modelling tools. This section provides some example SQL queries to retrieves and analyse data form the seamount database.

The new seamount database includes several datasets which are commonly used with core seamount data for analytical or reporting purposes. This appendix provides examples of SQL queries making use of these extra data. These are intended as indicative examples of the types of queries that the database supports, not an exhaustive list of the queries than can be run.

These datasets are as provided by the appropriate authority, generally as a shapefile which has been imported into the database.

Spatial data coordinates can be stored in (and converted between) various Coordinate Reference Systems (CRS). There is an EPSG (European Petroleum Survey Group) registry for these which provides an integer identifier which is commonly used to identify the CRS.

The language used to query a relational database is SQL (Structured Query Language). All the examples included here are using SQL. Any inline comments in SQL are prefixed with a double hyphen (–). These are used to make the queries more self explanatory and will be ignored by the query parser, so the examples can be run by cut and pasting into a suitable client application.

There are many applications that can be used to connect to and query a PostgreSQL database, Squirrel SQL, Razor SQL, DBeaver and PgAdmin are graphical clients. Psql is a command line client. All send SQL queries to the database and return the result. Of the ones mentioned DBeaver is a free client application with native support for PostGIS spatial data enabling query output to be seen on a map as well as textual output.

Users of the R statistical and modelling application can connect directly to the database from R and run the same SQL queries, with the outputs loaded into R objects in their workplace, ready for further analysis or visualisation.

Any user who has access to the database can run these and similar SQL queries. To connect to a PostgreSQL database, irrespective of the software being used, the following parameters are required:

- Database server (fully qualified name or IP address)
- Port (the network port to use to connect, generally 4326)
- Database name
- Authorised user name
- Password

Point in polygon

A common analysis relates seamounts to a spatial (polygon) dataset, such as one that defines political, fishing effort fisheries management or environmental areas around New Zealand. Typically, the result identifies the particular area each seamount of interest lies within.

Spatial data are stored using a defined Coordinate Reference System (CRS). When relating coordinate datasets in a query, all the datasets need to be using a common CRS. For user convenience, seamount peaks are stored using latitude/longitude coordinates (EPSG:4326) but the area polygons are generally stored as Mercator41 northings and eastings (EPSG:3994). Any query relating seamount peaks to other spatial data needs to transform (reproject) the degree values to the CRS of the polygons.

Seamounts within Seamount Closure Areas

This query identifies the seamounts which lie within the Seamount Closure Areas defined around New Zealand. The test determines where the seamount peak (a point feature) is contained by a SCA polygon.

```
-- SCA point in poly
--
-- use "as" to rename output columns for clarity and removing ambiguity
--  both tables have a column called "name"
select s.reg_no as seamount_reg_no,
       s.name as seamount_name,
       a.id as sca_id,
       a.name as sca_name
from seamount_point s,
     map.sca_poly a
-- define a spatial join where an SCA polygon contains a seamount point
--  with the lat/long point coordinates (EPSG:4326) are transformed to
--  Mercator41 (EPSG:3994)
where ST_Contains(a.poly, ST_Transform(s.geom, 3994));
```

Output:

seamount_reg_no	seamount_name	sca_id	sca_name
2214		15JA2 c	Seamount 375 / Christable
2233		15JA2 c	Seamount 375 / Christable
3975		15JA2 b	Bollons
2253		15JA2 c	Seamount 375 / Christable
2386		15JA2 b	Bollons
447		2B2 f	Seamount 447
2833		6B2 a	Brothers
2228		15JA2 c	Seamount 375 / Christable

Seamounts within Benthic Protected Areas

Similar to the previous example, this SQL retrieves the BPA for each seamount, instead of the SCA.

```
-- BPA point in poly
--
-- use "as" to rename output columns for clarity and removing ambiguity
--  we are requesting more attributes for the BPA polygons
select s.reg_no as reg_no,
       s.name as seamount_name,
       -- convert the id to an integer instead of decimal, don't need
decimal
       b.mfish_id::int as mpi_id,
       b.identifier as bpa_id,
       b.area_name as bpa_name
from seamount_point s,
     map.bpa_poly b
-- define a patial join where an SCA polygon contains a seamount point
--  with the lat/long point coordinates (EPSG:4326) are transformed to
--  Mercator41 (EPSG:3994)
where ST_Contains(b.poly, ST_Transform(s.geom, 3994));
```

Example output:

reg_no	seamount_name	mpi_id	bpa_id
970	Berlin	14	BPA14
210	Brothers	6	BPA6
855	Cole	10	BPA10
1559	Colville Volcano	10	BPA10
834	Cotton	6	BPA6
816	Dons Pinnie	12	BPA12
327	Erik	14	BPA14

Overlapping polygons

This adds some complexity to the simple point in polygon use case, as we are using the seamount polygons rather than the peak point geometry, so we use an intersection function instead of a containment one to define the relationship between the datasets.

One use case is an extension to the previous BPA query, but instead of just identifying the seamounts lying within a BPA, we can determine the total area occupied by seamounts within each BPA.

It can also be used to identify trawls which overlap with a seamount polygon and from this retrieve catches by seamount. A further enhancement is to add fishing year into the output, to trends and changes in fishing patterns can be derived.

Determine the cumulative area of seamounts within a BPA

This is more complex, conducting several operations to merge, aggregate and calculate the result set.

It uses a subquery to pre-process the spatial data for the final query, calculating the BPA polygon areas and intersections with seamount polygons. It then calculates and aggregates the seamount areas by BPA, with all area calculations carried out in the LINZ NZ Continental Shelf 2000 CRS (EPSG:3581) which is an equal area CRS more suitable for this sort of calculation.

The output is then sorted, using the integer component of the BPA name.

```
-- cumulative area of seamounts within each bpa
--
-- create the intersections of all seamount polygons with bpa in a subquery
-- then sum the areas in LINZ equal area CRS by BPA converted to sqkm
-- formatting the numeric output to reduce irrelevant no of decimals
-- ordering by BPA, but by just the numeric value, not char based
select bpa_id,
       bpa_name,
       bpa_area_sqkm,
       (sum(ST_Area(ST_Transform(poly, 3851)))/1000000)::decimal(7,2)
          as seamount_area_sqkm
from (select b.identifier as bpa_id,
             b.area_name as bpa_name,
             ST_Area(ST_Transform(b.poly,3851))::bigint as bpa_area_sqkm,
             ST_Intersection(s.poly,b.poly) as poly
      from seamount_poly s,
           map.bpa_poly b) as my_table
group by bpa_id,
         bpa_name,
         bpa_area_sqkm
order by substr(bpa_id,4)::int;
```

Output:

bpa_id	bpa_name	bpa_area_sqkm	seamount_area_sqkm
BPA1	Challenger North	17543389305	0.18
BPA2	Campbell East	23267194807	0
BPA3	Hikurangi Deep	53835951695	771.59
BPA4	East Chatham Rise	5238245692	140.23
BPA5	Norfolk Deep	45179313606	3033.06
BPA6	Tectonic Reach	13746542103	3232.96
BPA7	Blink	285701	0
BPA8	Puysegur	188904742	37.73
BPA9	Fiordland Transect	40710640042	1283.23
BPA10	Kermadec	6.38994E+11	34444.43
BPA11	Mid Chatham Rise	8715106108	0
BPA12	Antipodes Transect	1.13289E+11	655.68
BPA13	Challenger South	30467294877	0
BPA14	Arrow Plateau	64735405615	2226.12
BPA15	Bounty Heritage	1815075011	0
BPA16	Campbell Heritage	3123999476	0

Trawl footprint cumulative area/catch tabular query.

This example clips commercial trawl footprint data to the relevant seamount polygon (for fished area calculations any trawled area outside of the seamount polygon is removed).

For each seamount which has been fished, the output provides the overall footprint (not the cumulative area, just cumulative footprint) area by seamount by fishing year.

```sql
-- retrieve merged trawl footprint area and catch per seamount per fishing
year
select p.seamount as reg_no,
       p.name as seamount_name,
       f.fyear::int as fishing_year,
       -- sum the catch per trawl per seamount/year
       sum(f.ttlctcw)::int as catch_kg,
       -- get the footprint area
       ST_Area(
          -- merge trawl polygons
          ST_Union(
            -- reproject to equal area
          ST_Transform(
            -- get only the intersection with a seamount (clip footprint)
            ST_Intersection(p.poly, f.poly)
          ,3851))
       )::decimal(10,2) as trawl_area_sqm
from footprint_polygons f,
     seamount_poly p
-- only where a trawl footprint polygon intersects a seamount polygon
where ST_Intersects(f.poly, p.poly)
group by p.seamount,
         p.name,
         f.fyear
order by p.seamount,
         f.fyear;
```

Example output:

reg_no	seamount_name	fishing_year	catch_kg	trawl_area_sqm
230	Colville Knolls	2019	36163	627520.06
463	Ohena Knoll	2019	64	54296.81
464	Clark Seamount	2019	54	63151.2
467	Whakatane	2019	243	100815.26
471	Waiotahi Knoll	2019	2280	214610.54
473	Nukuhou Knoll	2019	3010	245762.74
501	Hill3	2019	42	147302.54
502	Hill4	2019	21029	435099.04
504	Hill6	2019	646	34197.2

Querying NIC specimen data

The ‘collectionobjects’ table describes specimens held in the NIC. This table includes the location where the specimen was captured, so a spatial query can return any specimen information relating to a seamount.

```
-- relate seamounts to specimens held in the NIC
select c.catalognumber,
       c.taxonname::varchar(24),
       s.reg_no,
       s.name::varchar(20) as seamount_name
from specify.collectionobjects c,
     seamount_poly p,
     seamount_point s
-- ensure specimen location is within a seamount
where ST_Intersects(p.poly, ST_Transform(c.geom,3994))
  and p.seamount=s.reg_no;
```

Example output:

catalognumber	taxonname	reg_no	seamount_name
5	Monachometra kermadecens	907	Hinetapeka
39	Semitaspongia pulvinata	432	
77	Micropilina tangaroa	1462	
158	Ircinia turrita	326	
352	Decapoda	1461	
446	Kemphyra corallina	138	Devonport Seamount
450	Nematocarcinus gracilis	751	Mt Ghost
472	Comatulides dawsoni	1478	SM6
473	Comatulides dawsoni	1478	SM6
597	Lepidopora dendrostylus	544	Tuatoru Knoll
682	Inferiolabiata spinosa	2543	
683	Crypthelia studeri	2543	
708	Stephanohelia praecipua	546	Wanganella Pin
709	Stephanohelia praecipua	546	Wanganella Pin
710	Inferiolabiata spinosa	546	Wanganella Pin
711	Inferiolabiata spinosa	546	Wanganella Pin

Querying research trawl station and catch data

The seamount database includes two tables containing data describing research trawls and catches. Much like the above SQL, the incidence and catches of research trawls on seamounts can be retrieved.

```
-- relating research trawl data to seamounts
select s.reg_no,
       s.name,
       st.trip_code,
       st.station_no,
       c.species,
       coalesce(c.weight, 0.100) as weight
from trawl.t_station st,
     trawl.t_catch c,
     seamount_point s,
     seamount_poly p
where ST_Intersects(ST_Transform(st.track, 3994),p.poly)
  and p.seamount = s.reg_no
  and st.trip_code||'_'||st.station_no = c.trip_code||'_'||c.station_no;
```