Database documentation for the Ministry for Primary Industries Acoustic database

Alexandre Schimel, Yoann Ladroit, Brent Wood, David Fisher, Jeremy Yeoman

NIWA Fisheries Data Management

Database Documentation Series

Updated May 2020

Revision History

Version	*Date*	*Change*	*Responsible*
1.0	January 2006	First release	Gavin Macaulay
1.1	June 2006	Modifications from NIWA internal review	Gavin Macaulay
1.2	June 2016	Update to reflect change from the Ministry of Fisheries to the Ministry for Primary Industries	Adam Dunford
2.0	May 2018	New documentation for the upgraded acoustic database (project SEA201711)	Alexandre Schimel, Yoann Ladroit
2.1	October 2018	Minor modifications in some attributes' comments	Alexandre Schimel
2.2	October 2018	Completed authors list; Finalized attributes descriptions to match database comments; Minor text and figure modifications	Alexandre Schimel
2.3	May 2019	Updated section on the database loading tool	Alexandre Schimel
2.4	May 2020	Updated section on the database loading tool again	Alexandre Schimel

Contents

1 Database Document Series

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

This MPI data set incorporates historic research data, data collected or held 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 provides an introduction to the acoustic database and is a part of the database documentation series produced by NIWA. It supersedes the original documentation on this database, and previous revisions (Macaulay and Dunford, 2016).

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 fit in together.

This document is intended as a guide for users and administrators of the acoustic database. Access to this database is restricted to nominated personnel as specified in the current Data Management contract between the Ministry for Primary Industries and NIWA. Any requests for data should in the first instance be directed to the Ministry.

2 Acoustic data collection

The acoustic database is designed for the storage of fisheries acoustic data. Fisheries acoustic data are collected using echosounders mounted on a vessel, towed by a vessel, set in a fixed location, or installed on another platform. The performance and characteristics of the echosounders used vary widely but generally involve sophisticated electronic equipment and associated software. MacLennan and Simmonds (1992) provide a detailed description of how and why fisheries acoustic data are collected. The echosounders used to collect acoustic data have changed considerably over the years, driven mainly by advances in electronic and computer technologies. However, the basic format of fisheries acoustic data acquisition has remained the same: An echosounder periodically emits a pulse of sound (it 'pings') and then records the echoes of this pulse, scattered by targets in the water medium. The range and amplitude of the echoes are measured and stored and can then be analysed. Associated data such as the date, time, and vessel position are also stored.

NIWA records and processes fisheries acoustics data on behalf of MPI, primarily to estimate changes in the species abundance of selected fisheries within New Zealand's EEZ. Species for which acoustic data have been collected include hoki, hake, smooth and black oreos, orange roughy, and southern blue whiting. Areas from which these data have been collected include the Chatham Rise, the Campbell Plateau, the west coast of the South Island, Cook Strait and various inshore regions around New Zealand. Aside from the acoustic data, this growing dataset includes various ancillary data that provide context to the acoustic data, such as vessel position, speed, and direction of travel. Acoustic data have been collected from acoustic surveys starting in 1984 through to the present day.

The volume and complexity of this dataset reflects this long history along with the various technological developments that occurred within this time frame, whether in acquisition vessels (e.g. research vessel Tangaroa, commercial fishing vessels, etc.), mission types (trawling, acoustic surveys, AOS, etc.), sensors (CREST, SIMRAD EK60) and data processing software (Echoview, ESP, ESP2, ESP3). The primary records for this ever-growing dataset are held on NIWA's Odin server[1].

[1] \\odin.niwa.co.nz\acoustic

3 The *acoustic* database

The *acoustic* database contains and archives standardized metadata for the acoustic dataset. The *acoustic* database is used to record various vital information that may or may not already be contained in the acoustic dataset, for example, information about the project motivating the survey, survey details such as the vessel name and the echosounder equipment and software settings, or results of any equipment calibrations.

The *acoustic* database was first developed at NIWA in the early 2000s by Gavin Macaulay and Roger Coombs (Macaulay and Dunford, 2016). It was a relational database built with PostgreSQL with the addition of geometric elements to allow spatial data (e.g. navigation data) to be incorporated for possible data access through a Geographical Information System (Postgis).

Up until 2015, loading metadata to the *acoustic* database was a significant part of the acoustic data acquisition and processing workflow. This process had the following features (Figure 1):

- Data from CREST sounders were loaded in the *acoustic* database directly, while data from SIMRAD sounders (EK60) required to be converted to the CREST format, prior to loading;
- Both native CREST data and SIMRAD-converted-to-CREST data were loaded into the database using custom-written C++ code;
- The database had a single public schema.

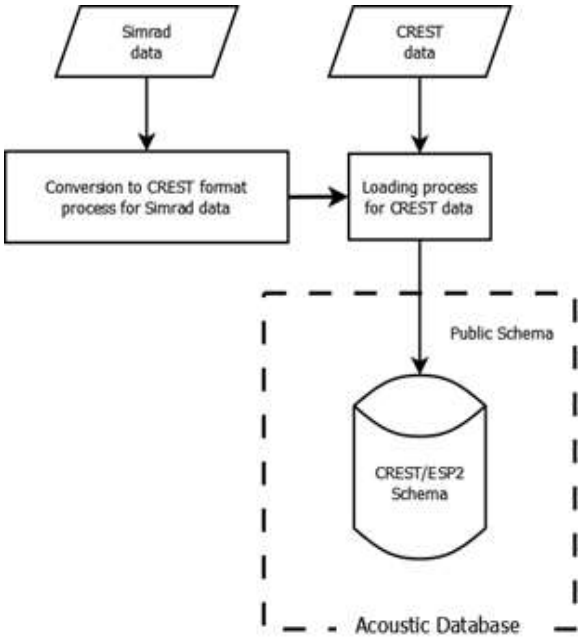

Figure 1: Schematic diagram of the acoustic database (pre-2016).

In 2015, NIWA upgraded its data processing software from the historical ESP2 to the new and more efficient ESP3. In particular, ESP3 was able to process SIMRAD data without needing to convert them to the CREST data format. Since this format conversion was necessary for the loading of metadata in the database, this software upgrade resulted in the technical

impossibility to update the database with new data. This requirement for format conversion in the database loading process was also a hindrance to other research providers (e.g. CSIRO and Deepwater Group Limited) carrying out acoustic surveys for MPI who also experienced difficulties with having data loaded into this database. A new database model was therefore urgently required to accommodate the current and future fisheries acoustics data.

Meanwhile in 2010, the Working Group on Fisheries Acoustics, Science and Technology (WGFAST) of the International Council for the Exploration of the Sea (ICES) brought together international experts into a Topic Group with the objective of standardizing metadata protocols for fisheries acoustics data acquisition, processing, quality control and data dissemination. Through its annual meetings, the group has been developing a metadata convention – effectively, a standard database blueprint for fisheries acoustics datasets (ICES, 2016). Earlier versions of this metadata convention have already been used by other national fisheries agencies, such as the French Institute for the Exploitation of the Sea (IFREMER) to redesign their database Echobase[2], and Australia's CSIRO to package their data for distribution under the IMOS project. In that context, it was evident that a new database model designed and implemented at NIWA would require to be based on the ICES metadata convention to be compliant with international best practice and compatible with that of other international bodies managing fisheries acoustics data – particularly the Integrated Marine Observing System (IMOS) Australian Ocean Data Network (AODN) portal[3]. In 2017, MPI contracted NIWA to design and implement an upgraded acoustic database[4].

The upgraded acoustic database was designed, built in PostgreSQL, and installed on NIWA's "wellfisheriesdb" server in 2018. Its organization is more complex than the original database due to the need to accommodate both the historical and new metadata in their different formats.

In this document, we will use the following terminology:

- A *template* is a relational database design, including table names, attribute names and types, constraints, and formalized relationships between tables. A template does not contain any data.
- A *schema* is an instantiated template within a database. It is an actual collection of tables containing attributes and constraints and linked through formalized relationships. Data within can be added, altered, removed or queried by authorised users.
- A *view* is a virtual table. It is collection of pre-written SQL commands that create a virtual table combining data from one or more tables across several schemas and appearing to the user as if it were an existing table in a database. Its purpose is for display of database data in a more convenient way.
- A *database* is a collection of schemas and views in a same space, allowing straightforward interaction between schemas (e.g. data transfer, querying of data from

[2] http://echobase.codelutin.com/v/latest/en/index.html
[3] https://portal.aodn.org.au/
[4] NIWA project code: SEA201711

different schemas, visualizing views, etc.). A database can be existing as a file (if instantiated with SQLite) or as an object on a server (if instantiated with PostgreSQL). The new acoustic database makes use of both types.

The upgraded acoustic database essentially relies on two different templates:

- "acdb_1": the original design by Macaulay and Dunford (2016). See this reference for the detailed information on this template.
- "acdb_2": a newly designed template for the new acoustic database, inspired from the ICES metadata convention. Detailed information can be found in the present document.

The upgraded acoustic database is installed on the "wellfisheriesdb" server and contains three different schemas (Figure 2):

- "LOAD": a schema based on the acdb_2 template, to hold new data temporarily.
- "ESP3": a schema based on the acdb_2 template, to hold recent and future data.
- "ESP2": a schema that is a direct copy of the old database, and thus based on the acdb_1 template.

The acoustic database has several views in the public schema, which allow users to read data from both the ESP3 and ESP2 schemas.

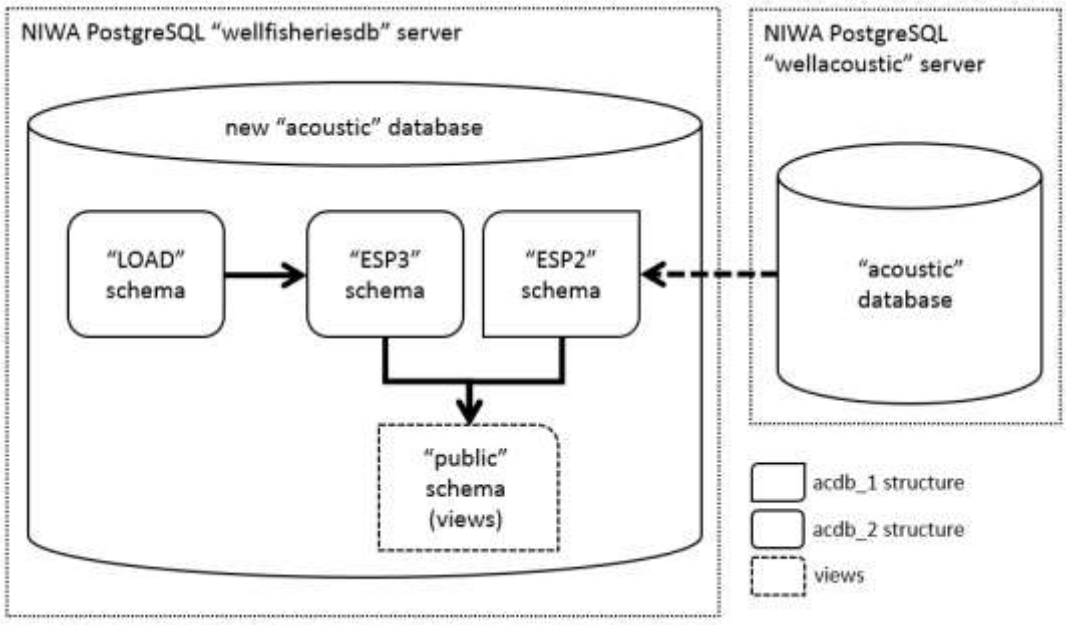

Figure 2: Schematic diagram of the schemas and organization of the new database.

The loading of one or several sets of metadata for new surveys in the acoustic database is done in 3 steps (Figure 3):

- Step 1: Create a "MINIDB" database (built in SQLite from the acdb_2 template) on the local machine containing metadata generated from one or several new acoustic datasets.

- Step 2: Load the contents of one or several “MINIDB” databases into the “LOAD” schema on the server, followed by data check.
- Step 3: Load the contents of the “LOAD” schema into the “ESP3” schema, followed by final data check. This operation also realizes a backup of the metadata being loaded to provide an audit trail.

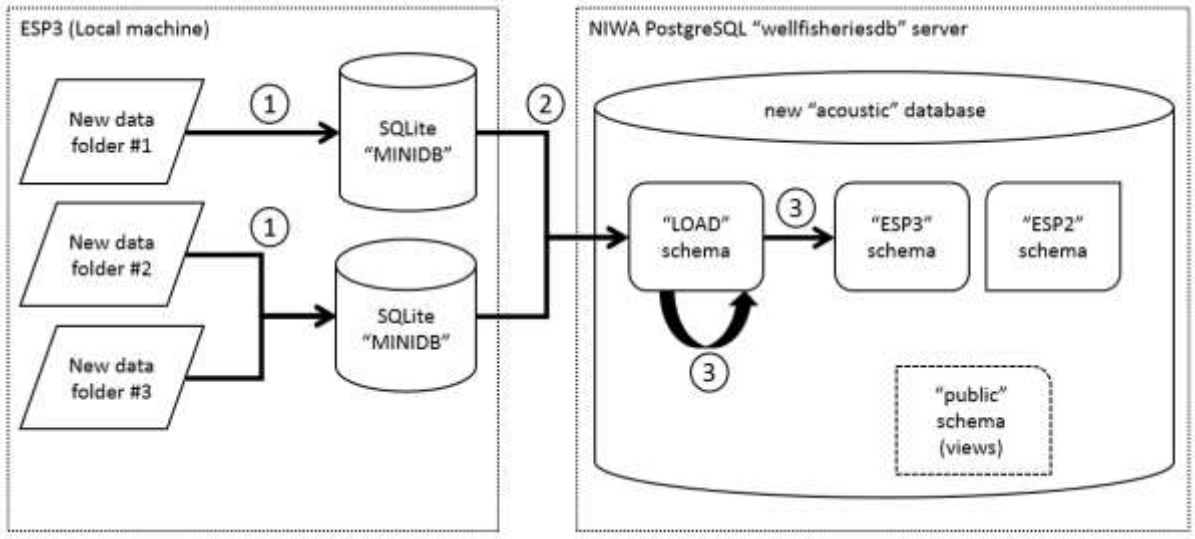

Figure 3: Schematic diagram of the process of loading new metadata in the database, operated in three steps.

4 Structure of the new "acdb_2" template

4.1 Design overview

The philosophy behind the design of the acdb_2 template was to hold the detailed metadata of acoustic data acquisition from the general (e.g. details about the project behind data acquisition) to the more detailed (e.g. survey-specific transect codes, or navigation), as well as all information necessary for data processing (e.g. acquisition parameters and calibration results), but no information on data processing itself (e.g. bottom detection, or parameters for echo integration) since data processing is subject to repeat and modification. The design was to follow the ICES metadata convention as close as possible.

The acdb_2 template was thus designed organically:

- First, each category of the ICES metadata convention was translated into an individual table.
- Then, attributes in each table were added for MPI/NIWA specific needs. For example, NIWA voyage codes and codified setup types such as "hull", "towbody" and "AOS".
- Tables were modified, and other tables created based on current and projected needs. For example, navigation table or creating a generic "deployment" table to incorporate possible glider or drifting buoys deployments where the ICES metadata convention was limited to "cruise" and "mooring".
- Attributes, tables and constraints were added as required for the database. For example, primary keys, foreign keys, controlled vocabulary tables, or join tables to manage many-to-many relationships between two tables.
- Finally, the design was modified to ensure database coherence, remove redundancy, and improve ease of use and understanding. For example, separate tables were created for transducers and transceivers where the ICES metadata convention was limited to a single "instruments" category.

The design was repeatedly tested and modified for consistency, resulting in tables presenting a significant departure from the original list of categories in the metadata convention, but holding all attributes in the convention.

The finalized acdb_2 template is organized in three thematic groups of tables (Figure 4):

- A first group of tables relate to the LOGISTICS behind the acquisition of the dataset. They hold metadata about the mission or project for which the dataset was collected, the cruise, mooring or other deployment used for the survey, and the vessel or vehicle from which data were collected.
- A second group of tables relate to the INSTRUMENTS used to acquire the data. They hold metadata about the transducer(s) and transceivers(s) used, about their installation/setup (e.g. location and orientation), the relevant calibration values, and the parameters of acquisition.

- A third group of tables relate to the DATA themselves. They hold metadata about the files acquired, the transects these files correspond to, information about the ancillary sensors used, the software used for acquisition, and the navigation data.

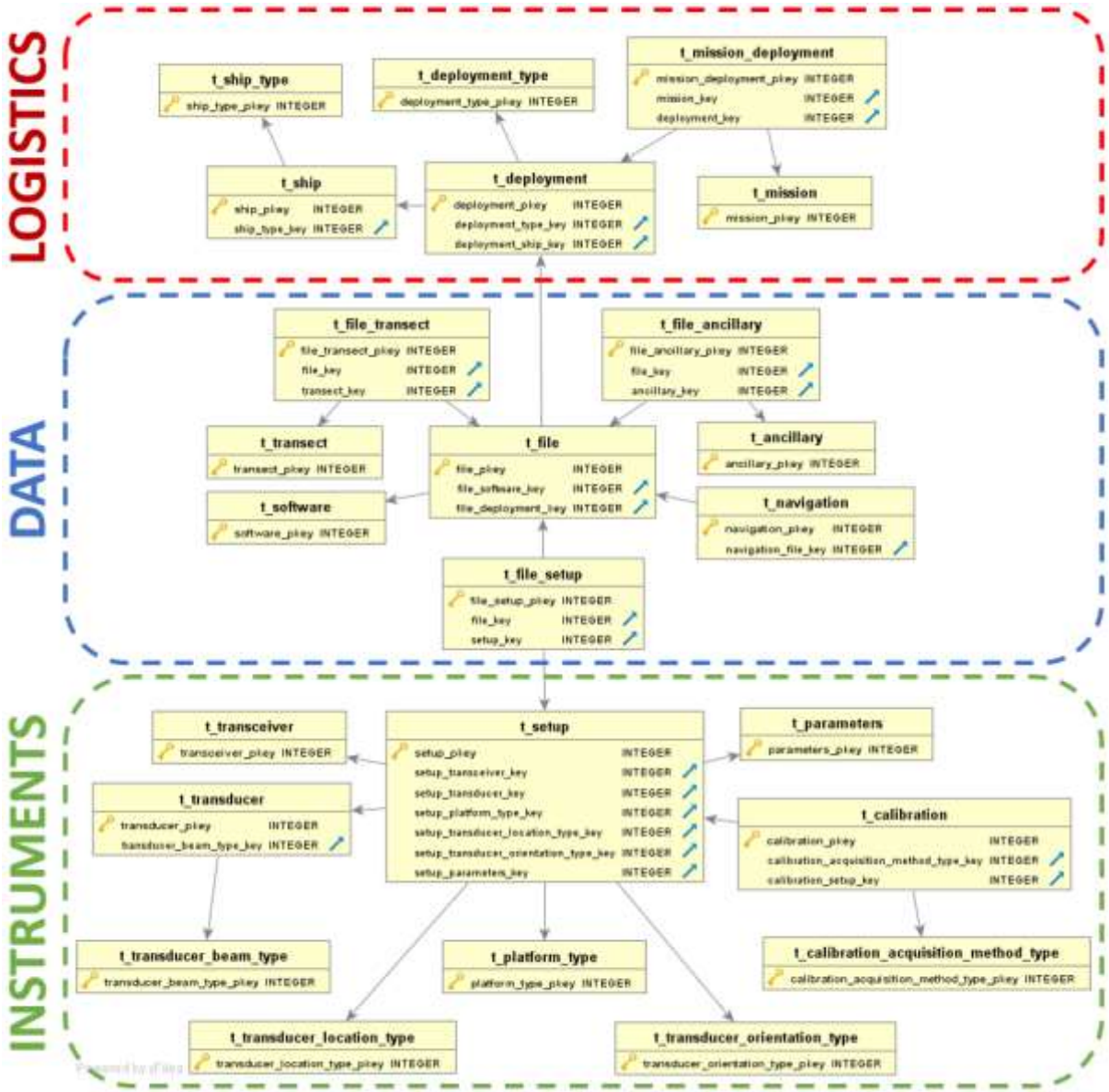

Figure 4: Entity Relationships Diagram (ERD) of the new acdb_2 template.

Irrespective of the group, there are three types of tables: regular, lookup and join tables.

- Regular tables contain the actual metadata. The name of a regular table is always a single word.
- Lookup tables are used to hold a limited number of possible entries for an attribute, i.e. to specify controlled vocabulary. The name of a lookup table always ends up in “type”.
- Join tables are necessary to manage “many-to-many” relationships between two regular tables. The name of a join table is always the combination of the two words making up the name of the regular tables it links.

Following NIWA conventions for database design, all tables start with the prefix "t_". The primary key for all tables is an auto-incrementing SERIAL attribute bearing the name of the table (without the prefix) with the suffix "_pkey" (e.g., attribute mission_pkey is the primary key for table `t_mission`).

4.2 Table relationships

The main difference between the ICES metadata convention and a relational database design is that relationships between tables in a database must be formalized and constrained in the design itself.

In any database, there are three main types of relationships between two tables:

- A “one-to-one” relationship is when an entry in a table A and an entry in a table B can be unambiguously linked to one another.
- A “many-to-one” relationship is when an entry in a table A can be unambiguously linked to a single entry in a table B, but that entry in table B can be linked to several other entries in table A.
- A “many-to-many” relationship is when an entry in a table A can be linked to several entries in a table B, and an entry in a table B can be linked to several entries in a table A.

“One-to-one” relationships

acdb_2 does not have any “one-to-one” relationship as this type of relationship implies that tables A and B can be transparently merged. This was systematically performed in this design to limit the number of tables and complexity of table relationships.

“Many-to-one” relationships

“Many-to-one” relationships are the most common relationship in acdb_2. For example, a deployment of the “cruise” type makes use of a single ship, but any one ship can be used for several deployments, thus resulting in a “many-to-one” relationship between tables `t_deployment` and `t_ship`. Note that it is possible for “many-to-one” relationships to be mandatory or optional. For example, a deployment of type “cruise” will necessarily reference one and only one ship, but a deployment of type “mooring” will not refer to any ship.

In acdb_2, “Many-to-one” relationships are formalized using a foreign key in the table on the "many" side referencing the primary key in the table on the "one" side.

One specific use of “Many-to-one” relationships are lookup tables for controlled vocabulary. For example, according to the ICES metadata convention, a ship can be of a limited number of types: “research”, “fishing” or “other”. Any one ship can be of one and only one type, but any type applies to zero, one or several ships, which can be translated as a “many-to-one” relationships between the table `t_ship`, and a lookup table `t_ship_type` existing solely to hold that controlled vocabulary for that specific attribute.

In acdb_2, a table ending with the word "type" (e.g. `t_ship_type`) is a lookup table for controlled vocabulary for an attribute in a regular table.

“Many-to-Many” relationships

“Many-to-many” relationships are also found through acdb_2. For example, data acquired in a given cruise (in table `t_deployment`) may pertain to several projects (in table `t_mission`), while a given project may have required acquisition of data through several cruises, thus resulting in a “many-to-many” relationship between tables `t_deployment` and `t_mission`. As in “many-to-one” relationships, it is possible for relationships to be mandatory or optional. For example, a mission may be entered for which no deployment was performed yet.

A “many-to-many” relationship between two tables is implemented by the creation of an additional “join” table that performs a “many-to-one” relationship to both tables, and thus allows unambiguously keeping track of which entries in two tables are linked. Join tables are named after the two tables they join, for example `t_mission_deployment` is the join table that performs the “many-to-many” relationship between `t_mission` and `t_deployment`.

4.3 The LOGISTICS group of tables

This group has three regular tables:

- `t_mission`: Missions or projects for which acoustic data were collected.
- `t_ship`: Ships or vessels from which acoustic data were collected.
- `t_deployment`: Individual deployments of one or several acoustic instruments on one platform (cruise, mooring deployment, drifting buoy, glider, autonomous vehicle, etc.) with which acoustic data were collected. Any collection of instruments operated from a same ship are counted as one deployment, with the t_setup table being used to differentiate between platforms (i.e. hull, towbody, AOS).

The relationships between these tables are:

- 1 mission makes use of 0, 1 or several deployments, while 1 deployment allows for acquisition of data for 1 or several missions (i.e. “many-to-many” relationship between `t_mission` and `t_deployment`).
- 1 ship is used in 0, 1 or several deployments, while 1 deployment makes use of 0 or 1 ship (i.e. “many-to-one” relationship between `t_ship` and `t_deployment`).

This group has two lookup tables:

- `t_ship_type`: Controlled vocabulary for type of ship (ship_type attribute in t_ship), as per ICES vocabulary: "Ship, research", "Ship, fishing" or "Ship, other".
- `t_deployment_type`: controlled vocabulary for type of deployment, as per ICES vocabulary: "Ship", "Buoy, moored", "Buoy, drifting", "Glider", "Underwater vehicle, autonomous, motorized", or "Underwater vehicle, autonomous, glider".

This group has one join table:

- `t_mission_deployment`, to manage the “many-to-many” relationship between `t_mission` and `t_deployment`.

Notes about design:

- `t_mission` reflects the ICES category “Mission” closely. All attributes from this category are found in this table, except for attribute “mission_platform_type” which is split between look-up tables `t_ship_type` and `t_deployment_type`. This is because a mission may use several platforms.
- `t_deployment` combine the attributes of ICES categories "Cruise" and "Mooring". This was done because many attributes in these categories were common and can be generalized to other current and future types of deployments (buoy, glider, etc.). Only the attribute mooring_depth is missing from table `t_deployment`, as this is a feature of the `t_setup` table.
- `t_ship` reflects the ICES category "Ship" closely. All attributes in this category are found in this table, with `t_ship_type` used as a look-up table for the ICES attribute “ship_type”.

t_mission

Comment on table: Missions or projects for which acoustic data were collected.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
mission_pkey	integer	Primary key as assigned by the database loading software.	
mission_name	text	Name of mission/project.	Mission/mission_name.
mission_abstract	text	Text description of the mission/project, its purpose, scientific objectives and area of operation. Other instruments and experiments within the mission/project which may or may not relate directly to the acoustic data can be included.	Mission/mission_abstract.
mission_start_date	timestamp without time zone	Start date and time of the mission/project.	Mission/mission_start_date.
mission_end_date	timestamp without time zone	End date and time of the mission/project.	Mission/mission_end_date.
principal_investigator	text	Name of the principal investigator in charge of the mission/project.	Mission/principal_investigator.
principal_investigator_email	text	Principal investigator e-mail address.	Mission/principal_investigator_email.
institution	text	Name of the institute, facility, or company where the original data were produced.	Mission/institution.

data_centre	text	Data centre in charge of the data management or party who distributed the resource.	Mission/data_centre.
data_centre_email	text	Data centre contact e-mail address.	Mission/data_centre_email.
mission_id	text	ID code of mission/project. Use MPI project code here. Remove all slashes and hyphens, e.g. SEA201711	Mission/mission_id.
creator	text	Entity primarily responsible for making the resource.	Mission/creator.
contributor	text	Entity responsible for making contributions to the resource.	Mission/contributor.
mission_comments	text	Free-text field for relevant information not captured by other attributes.	Mission/mission_comments.

t_deployment

Comment on table: Individual deployments of 1 or several acoustic instruments on 1 platform (cruise, mooring deployment, drifting buoy, glider, autonomous vehicle, etc.) with which acoustic data were collected. Any collection of instruments operated from a same ship are counted as one deployment, with the t_setup table being used to differentiate between platforms (i.e. hull, towbody, AOS).

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
deployment_pkey	integer	Primary key as assigned by the database loading software.	
deployment_type_key	integer	Describe type of deployment platform that is hosting the acoustic instrumentation. See controlled vocabulary table t_deployment_type.	Mission/mission_platform.
deployment_ship_key	integer	If this deployment is a cruise, refers to relevant entry in t_ship for ship used.	
deployment_name	text	Formal name of cruise/mooring/deployment as recorded by deployment documentation or institutional data centre.	Cruise/cruise_name.
deployment_id	text	Deployment ID where one exists. Use NIWA trip code here.	Mooring/mooring_code. Cruise/cruise_id.
deployment_description	text	Free text field to describe the cruise/deployment. For a cruise, may include a list of cruise objectives, such as scientific survey, commercial fishing, resupply, or combination of these. For a mooring, describe type of mooring.	Mooring/mooring_description. Cruise/cruise_description.
deployment_area_description	text	For a cruise, list main areas of operation. For a mooring, name of the location where mooring is deployed.	Mooring/mooring_site_name. Cruise/cruise_area_description.

deployment_operator	text	Name of organisation which operates the cruise/deployment.	Mooring/mooring_operator.
deployment_summary_report	text	Published or web-based references that links to the cruise/deployment report.	Cruise/cruise_summary_report.
deployment_start_date	timestamp without time zone	Start date of cruise, or mooring deployment.	Mooring/mooring_deployment_date. Cruise/cruise_start_date.
deployment_end_date	timestamp without time zone	End date of cruise, or retrieval of mooring.	Mooring/mooring_retrieval_date. Cruise/cruise_end_date.
deployment_northlimit	double precision	The constant coordinate for the northernmost face or edge.	Mooring/mooring_northlimit. Cruise/cruise_northlimit.
deployment_eastlimit	double precision	The constant coordinate for the easternmost face or edge.	Mooring/mooring_eastlimit. Cruise/cruise_eastlimit.
deployment_southlimit	double precision	The constant coordinate for the southernmost face or edge.	Mooring/mooring_southlimit. Cruise/cruise_southlimit.
deployment_westlimit	double precision	The constant coordinate for the westernmost face or edge.	Mooring/mooring_westlimit. Cruise/cruise_westlimit.
deployment_uplimit	double precision	The constant coordinate for the uppermost face or edge in the vertical, z, dimension.	Mooring/mooring_uplimit. Cruise/cruise_uplimit.

deployment_downlimit	double precision	The constant coordinate for the lowermost face or edge in the vertical, z, dimension).	Mooring/mooring_downlimit. Cruise/cruise_downlimit.
deployment_units	text	The units of unlabelled numeric values of deployment_northlimit, deployment_eastlimit, deployment_southlimit, deployment_westlimit.	Mooring/mooring_units. Cruise/cruise_units.
deployment_zunits	text	The units of unlabelled numeric values of deployment_uplimit, deployment_downlimit).	Mooring/mooring_zunits. Cruise/cruise_zunits.
deployment_projection	text	The name of the projection used with any parameters required.	Mooring/mooring_projection. Cruise/cruise_projection.
deployment_start_port	text	Commonly used name for the port where cruise/deployment started. Irrelevant for mooring.	Cruise/cruise_start_port.
deployment_end_port	text	Commonly used name for the port where cruise/deployment ended. Irrelevant for mooring.	Cruise/cruise_end_port.
deployment_start_bodc_code	text	Name of port from where cruise/deployment started. Recommended use of British Oceanographic Data Centre (BODC) port gazetteer. Irrelevant for mooring.	Cruise/cruise_start_BODC_code.
deployment_end_bodc_code	text	Name of port from where cruise/deployment ended. Recommended use of British Oceanographic Data Centre (BODC) port gazetteer. Irrelevant for mooring.	Cruise/cruise_end_BODC_code.
deployment_comments	text	Free-text field for relevant information not captured by other attributes.	Mooring/mooring_comments. Cruise/cruise_comments.

t_ship

Comment on table: Ships or vessels from which acoustic data were collected.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
ship_pkey	integer	Primary key as assigned by the database loading software.	
ship_name	text	Name of the ship.	Ship/ship_name.
ship_type_key	integer	Describe type of ship that is hosting the acoustic instrumentation. See controlled vocabulary table `t_ship_type`.	Ship/ship_type. Mission/mission_platform.
ship_code	text	For example, in-house code associated with ship. At NIWA, the three letters code (e.g. TAN for Tangaroa).	Ship/ship_code.
ship_platform_code	text	ICES database of known ships.	Ship/ship_platform_code.
ship_platform_class	text	ICES controlled vocabulary for platform class.	Ship/ship_platform_class.
ship_callsign	text	Ship call sign.	Ship/ship_callsign.
ship_alt_callsign	text	Alternative call sign if the ship has more than one.	Ship/ship_alt_callsign.
ship_imo	integer	Ships International Maritime Organization ship identification number.	Ship/ship_IMO.
ship_operator	text	Name of organization of company which operates the ship.	Ship/ship_opeartor.
ship_length	double precision	Overall length of the ship (m).	Ship/ship_length.

ship_breadth	double precision	The width of the ship at its widest point (m).	Ship/ship_breadth.
ship_draft	double precision	The draft of the ship (m).	
ship_tonnage	double precision	Gross tonnage of the ship (t).	Ship/ship_tonnage.
ship_engine_power	double precision	The total power available for ship propulsion.	Ship/ship_engine_power.
ship_noise_design	text	For example, ICES 209 compliant (Mitson, 1995). Otherwise description of noise performance of the ship.	Ship/ship_noise_design.
ship_aknowledgement	text	Any users (include re-packagers) of these data are required to clearly acknowledge the source of the material in this format.	Ship/ship_aknowledgement.
ship_comments	text	Free-text field for relevant information not captured by other attributes.	Ship/ship_comments.

Lookup table `t_ship_type`

Comment on table: Controlled vocabulary for ship_type attribute in `t_ship`.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
ship_type_pkey	integer	Primary key as assigned by the database loading software.	
ship_type	text	Describe type of ship that is hosting the acoustic instrumentation: "Ship, research", "Ship, fishing" or "Ship, other".	Ship/ship_type. Mission/mission_platform.

Lookup table `t_deployment_type`

Comment on table: Controlled vocabulary for deployment_type attribute in `t_deployment`.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
deployment_type_pkey	integer	Primary key as assigned by the database loading software.	
deployment_type	text	Describe type of deployment platform that is hosting the acoustic instrumentation: "Ship", "Buoy, moored", "Buoy, drifting", "Glider", "Underwater vehicle, autonomous, motorized", or "Underwater vehicle, autonomous, glider". ICES entry "Underwater vehicle, towed" was removed from the list as towed bodies, AOS and other systems that need a ship to be operated fall under the "Ship" category.	Mission/mission_platform.

Join table `t_mission_deployment`

Comment on table: Join table to manage the many-many relationship between `t_mission` and `t_deployment`.

Table attribute	*Type*	*Description*
mission_deployment_pkey	integer	Primary key as assigned by the database loading software.
mission_key	integer	Refers to relevant entry in `t_mission`.
deployment_key	integer	Refers to relevant entry in `t_deployment`.
mission_deployment_comments	text	Free-text field for relevant information not captured by other attributes.

4.4 The INSTRUMENTS group of tables

This group has five regular tables:

- `t_transducer`, describing each (physical) transducer (sonar head);
- `t_transceiver`, describing each (physical) transceiver (General Purpose Transceiver);
- `t_calibration`, describing each result for a calibration session;
- `t_parameters`, describing each set of acquisition parameters;
- `t_setup`, describing each individual combination of the above tables, as well as location/orientation;

The relationships between these tables are:

- 1 setup makes use of 1 transducer, 1 transceiver, and 1 set of parameters, while
- 1 transducer can be used in 0, 1 or several setups (i.e. “many-to-one” relationship between `t_transducer` and `t_setup`).
- 1 transceiver can be used in 0, 1 or several setups (i.e. “many-to-one” relationship between `t_transceiver` and `t_setup`).
- 1 set of parameters can be used in 0, 1 or several setups (i.e. “many-to-one” relationship between `t_parameters` and `t_setup`).
- 1 setup refers to 0, 1 or several calibration results; and 1 calibration result refers to 1 setup (i.e. “many-to-one” relationship between `t_setup` and `t_calibration`.

This group has five lookup tables:

- `t_calibration_acquisition_method_type`: controlled vocabulary for type of calibration, as per ICES vocabulary: "Standard sphere, in-situ", "Standard sphere, tank", "Standard sphere, other", "Reciprocity", "Hydrophone", "Seafloor reflection", "Nominal", or "Intership".
- `t_transducer_beam_type`: controlled vocabulary for type of sonar, as per ICES vocabulary: "Single-beam", "Single-beam, split-aperture", "Multibeam", or "Multibeam, split-aperture".

- `t_platform_type`: controlled vocabulary for transducer location as per NIWA vocabulary: "Hull", "Towbody" or "AOS".
- `t_transducer_location_type`: controlled vocabulary for transducer location, as per ICES vocabulary: "Hull, keel", "Hull, lowered keel", "Hull, blister", "Hull, gondola", "Towed, shallow", "Towed, deep", "Towed, deep, trawlnet attached", or "Ship, pole".
- `t_transducer_orientation_type`: controlled vocabulary for transducer orientation, as per ICES/NIWA vocabulary: "Downward-looking", "Upward-looking", "Sideways-looking, forward", "Sideways-looking, backward", "Sideways-looking, port", "Sideways-looking, starboard", "Other"

This group does not use any join table.

Notes about design:

- The ICES metadata convention has a single "Instrument" category containing information for transducer, transceiver, and transducer setup (location such as hull, towbody, AOS, etc. or orientation such as downward/upward/etc.). A table based on such category would need a new entry every time a different transceiver/transducer combination is used, and every time an instrument is deployed on another location, implying extensive redundancy in the transducer and transceiver metadata. Table `t_setup` closely follows this concept, but additional tables were created to hold information about the transducers and transceivers to reduce redundancy. The `t_setup` table thus lists the individual combinations of 1 transducer and 1 transceiver, on 1 platform/location, with 1 orientation, and 1 set of acquisition parameters, to which one channel in an acoustic file can be uniquely linked.
- The tables `t_transducer` and `t_transceiver` contains all attributes of ICES category "Instrument", except "instrument_transducer_location", "instrument_transducer_depth" and "instrument_transducer_orientation", which are features of the setup and thus can be found in table `t_setup`.
- Frequency is a feature of a transducer/transceiver combination, and thus a single attribute in the ICES "Instrument" category, but here split in two separate attributes in the `t_transducer` and `t_transceiver` tables (frequency_nominal).
- Also, attributes were added for lower and upper bounds of frequency ranges for wide-band systems.

- setup_transducer_depth in `t_setup` is a mean depth for the setup (single value). This attribute is relevant for hull systems, moorings, drifting buoys or autonomous surface vehicles, but not for systems traveling up and down the water-column such as towbodies, AOS, gliders, etc.
- The table `t_parameters` is meant to contain all possible combination of parameters so that the parameters for a given setup can be summarized to one reference to the appropriate entry in `t_parameters`.
- The table `t_calibration` is meant to contain all results for all calibration sessions. One entry in that table thus refers to the setup that was calibrated during the session.

t_transducer

Comment on table: Transducers (sonar heads).

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
transducer_pkey	integer	Primary key as assigned by the database loading software.	
transducer_manufacturer	text	Name of transducer manufacturer.	Instrument/instrument_transducer_manufacturer.
transducer_model	text	Transducer model.	Instrument/instrument_transducer_model.
transducer_beam_type_key	integer	For example, "single-beam, split-aperture". See controlled vocabulary table `t_transducer_beam_type`.	Instrument/instrument_transducer_beam_type.
transducer_serial	text	Transducer serial number.	Instrument/instrument_transducer_serial.
transducer_frequency_lower	integer	For wide-band systems, lower bound of frequency range (kHz).	
transducer_frequency_nominal	integer	Frequency of the transceiver/transducer combination (kHz).	Instrument/instrument_frequency.
transducer_frequency_upper	integer	For wide-band systems, upper bound of frequency range in (kHz).	
transducer_psi	double precision	Manufacturer specified transducer equivalent beam angle, expressed as 10*log10(psi), where psi has units of steradians.	Instrument/instrument_transducer_psi.

transducer_beam_angle_major	double precision	Major beam opening, also referred to athwartship angle (deg).	Instrument/instrument_transducer_beam_angle_major.
transducer_beam_angle_minor	double precision	Minor beam opening, also referred to alongship angle (deg)	Instrument/instrument_transducer_beam_angle_minor.
transducer_comments	text	Free-text field for relevant information not captured by other attributes.	Instrument/instrument_comments.

t_transceiver

Comment on table: Transceivers (GPT).

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
transceiver_pkey	integer	Primary key as assigned by the database loading software.	
transceiver_manufacturer	text	Name of transceiver manufacturer.	Instrument/instrument_transceiver_manufacturer.
transceiver_model	text	Transceiver model.	Instrument/instrument_transceiver_model.
transceiver_serial	text	Transceiver serial number.	Instrument/instrument_transceiver_serial.
transceiver_frequency_lower	integer	For wide-band systems, lower bound of frequency range in (kHz).	
transceiver_frequency_nominal	integer	Frequency of the transceiver/transducer combination in (kHz).	Instrument/instrument_frequency.
transceiver_frequency_upper	integer	For wide-band systems, upper bound of frequency range in (kHz).	
transceiver_firmware	text	Transceiver firmware version.	Instrument/instrument_transceiver_firmware.
transceiver_comments	text	Free-text field for relevant information not captured by other attributes.	Instrument/instrument_comments.

t_calibration

Comment on table: Calibration results

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
calibration_pkey	integer	Primary key as assigned by the database loading software.	
calibration_date	timestamp without time zone	Date of calibration.	Calibration/calibration_date.
calibration_acquisition_method_type_key	integer	Describe the method used to acquire calibration data. See controlled vocabulary table `t_calibration_acquisition_method_type`.	Calibration/calibration_acquisition_method.
calibration_processing_method	text	Describe method of processing that was used to generate calibration offsets.	Calibration/calibration_processing_method.
calibration_accuracy_estimate	text	Estimate of calibration accuracy. Include a description and units so that it is clear what this estimate means (e.g. estimate might be expressed in dB or as a percentage).	Calibration/calibration_accuracy_estimate.
calibration_report	text	URL or references to external documents which give a full account of calibration processing and results may be appropriate.	Calibration/calibration_report.

calibration_setup_key	integer	Refers to relevant entry in `t_setup` for the setup to which the calibration refers to.	
calibration_frequency	double precision	Frequency at which calibration parameters apply to for this setup (Hz).	
calibration_gain	double precision	Estimated on-axis gain (dB).	
calibration_sacorrect	double precision	Sa correction as applied on Simrad systems (dB).	
calibration_phi_athwart	double precision	Estimated athwart beam angle (deg).	
calibration_phi_along	double precision	Estimated along beam angle (deg).	
calibration_phi_athwart_offset	double precision	Estimated athwart beam angle offset (deg).	
calibration_phi_along_offset	double precision	Estimated along beam angle offset (deg).	
calibration_psi	double precision	Estimated equivalent beam angle (dB).	
calibration_comments	text	Free-text field for relevant information not captured by other attributes.	Calibration/calibration_comments.

t_parameters

Comment on table: Software acquisition parameters for a setup

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
parameters_pkey	integer	Primary key as assigned by the database loading software.	
parameters_pulse_mode	text	Pulse mode, CW or FM.	
parameters_pulse_length	double precision	Pulse length (s). Applies to both CW and FM pulses.	
parameters_pulse_slope	double precision	Pulse slope (see description of this parameter in Simrad documentation). Applies to both CW and FM pulses.	
parameters_fm_pulse_type	text	Shape of pulse in case of FM: "linear up-sweep", "linear down-sweep", "exponential up-sweep", "exponential down-sweep", or other to be described.	
parameters_frequency_min	integer	Minimum frequency of pulse (Hz). Start frequency if FM pulse type is upsweep. End frequency if downsweep.	
parameters_frequency_max	integer	Maximum frequency of pulse (Hz). Start frequency if FM pulse type is downsweep. End frequency if upsweep.	
parameters_power	double precision	Transmit power (W).	
parameters_comments	text	Free-text field for relevant information not captured by other attributes.	

t_setup

Comment on table: Individual combinations of 1 transducer and 1 transceiver, on 1 platform/location, with 1 orientation, and 1 set of acquisition parameters, to which 1 channel in an acoustic file can be unambiguously linked.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
setup_pkey	integer	Primary key as assigned by the database loading software.	
setup_transceiver_key	integer	Refers to relevant entry in `t_transceiver` for the transceiver used in this setup.	
setup_transducer_key	integer	Refers to relevant entry in `t_transducer` for the transducer used in this setup.	
setup_platform_type_key	integer	Location of installed transducer following NIWA vocabulary: “Hull”, “Towbody” or “AOS”. See controlled vocabulary table `t_platform_type`.	
setup_transducer_location_type_key	integer	Location of installed transducer following ICES vocabulary: "Hull, keel", "Hull, lowered keel", "Towed, shallow", etc. See controlled vocabulary table `t_transducer_location_type`.	Instrument/instrument_transducer_location.

setup_transducer_location_x	double precision	For transducer installed on fixed position on ship, the position of the transducer on the X-axis, defined as running alongship, positive towards the bow, and referenced to the ship's reference point (m).	
setup_transducer_location_y	double precision	For transducer installed on fixed position on ship, the position of the transducer on the Y-axis, defined as running athwartship, positive towards starboard, and referenced to the ship's reference point (m).	
setup_transducer_location_z	double precision	For transducer installed on fixed position on ship, the position of the transducer on the Z-axis, defined as running vertically, positive downwards, and referenced to the ship's reference point (m).	
setup_transducer_depth	double precision	Mean depth of transducer face beneath the water surface (m).	Instrument/instrument_transducer_depth.
setup_transducer_orientation_type_key	integer	Direction perpendicular to the face of the transducer. See controlled vocabulary table `t_transducer_orientation_type`.	Instrument/instrument_transducer_orientation.
setup_transducer_orientation_vx	double precision	Coordinate along the X-axis (defined as running alongship, positive towards the bow) of the normal vector to the transducer face.	

setup_transducer_orientation_vy	double precision	Coordinate along the Y-axis (defined as running athwartship, positive towards starboard) of the normal vector to the transducer face.	
setup_transducer_orientation_vz	double precision	Coordinate along the Z-axis (defined as running vertically, positive downwards) of the normal vector to the transducer face.	
setup_parameters_key	integer	Refers to relevant entry in `t_parameters` for the set of parameters used in this setup.	
setup_comments	text	Free-text field for relevant information not captured by other attributes.	Instrument/instrument_comments.

Lookup table `t_calibration_acquisition_method_type`

Comment on table: Controlled vocabulary for calibration_acquisition_method_type attribute in `t_calibration`.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
calibration_acquisition_method_type_pkey	integer	Primary key as assigned by the database loading software.	
calibration_acquisition_method_type	text	Describe the method used to acquire calibration data: "Standard sphere, in-situ", "Standard sphere, tank", "Standard sphere, other", "Reciprocity", "Hydrophone", "Seafloor reflection", "Nominal", or "Intership".	Calibration/calibration_acquisition_method.

Lookup table `t_transducer_beam_type`

Comment on table: Controlled vocabulary for transducer_beam_type attribute in `t_transducer`.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
transducer_beam_type_pkey	integer	Primary key as assigned by the database loading software.	
transducer_beam_type	text	Beam type following ICES vocabulary: "Single-beam", "Single-beam, split-aperture", "Multibeam", or "Multibeam, split-aperture".	Instrument/instrument_transducer_beam_type.

Lookup table `t_platform_type`

Comment on table: Controlled vocabulary for platform_type attribute in `t_setup`.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
platform_type_pkey	integer	Primary key as assigned by the database loading software.	
platform_type	text	Location of installed transducer following NIWA vocabulary: "Hull", "Towbody" or "AOS". Redundant with `t_transducer_location_type` which reflects ICES category Instrument, attribute instrument_transducer_location.	

Lookup table `t_transducer_location_type`

Comment on table: Controlled vocabulary for transducer_location_type attribute in `t_setup`.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
transducer_location_type_pkey	integer	Primary key as assigned by the database loading software.	
transducer_location_type	text	Location of installed transducer following ICES vocabulary: "Hull, keel", "Hull, lowered keel", "Hull, blister", "Hull, gondola", "Towed, shallow", "Towed, deep", "Towed, deep, trawlnet attached", or "Ship, pole".	Instrument/instrument_transducer_location.

Lookup table `t_transducer_orientation_type`

Comment on table: Controlled vocabulary for transducer_orientation attribute in `t_setup`.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
transducer_orientation_type_pkey	integer	Primary key as assigned by the database loading software.	
transducer_orientation_type	text	Direction perpendicular to the face of the transducer: "Downward-looking", "Upward-looking", "Sideways-looking, forward", "Sideways-looking, backward", "Sideways-looking, port", "Sideways-looking, starboard", "Other".	Instrument/instrument_transducer_orientation.

4.5 The DATA group of tables

The regular tables in this group are:

- `t_software`, describing each instance of a new software install on a new computer (or updating to new version);
- `t_file`, describing each acoustic data file;
- `t_navigation`, describing simplified acoustic data file navigation;
- `t_transect`, describing each acoustic data transect;
- `t_ancillary`, describing any other instruments - GPS, pitch roll sensor, anemometer, etc.

The relationships between these tables are:

- 1 file was acquired with 1 software, while 1 software is used to acquire 0, 1 or several files (i.e. “many-to-one” relationship between `t_software` and `t_file`).
- 1 file has several navigation entries, while 1 navigation entry applies to 1 file (i.e. “many-to-one” relationship between `t_file` and `t_navigation`).
- 1 file can contain 0, 1 or several transects, while 1 transect can span 1 or several files (i.e. “many-to-many” relationship between `t_file` and `t_transect`).
- 1 file contain data from 0, 1, or several ancillary instruments, while 1 ancillary instrument can have its data recorded in 0, 1 or several files (i.e. “many-to-many” relationship between `t_file` and `t_ancillary`).

There are no controlled vocabulary tables in this group.

The join tables within this group are:

- `t_file_transect`, to manage the many-many relationship between `t_file` and `t_transect`.
- `t_file_ancillary`, to manage the many-many relationship between `t_file` and `t_ancillary`.

Notes about design:

- The ICES metadata convention category "Transect" only contains info about time (start/end) and space (XYZ) bounds for a "transect", and then "Data" and "Dataset" for the metadata of finalized, processed data. What we needed was one table listing the individual files recorded, and one table listing the "set of consecutive pings to be processed together whether they are a subset of the pings in one file, or span several consecutive files", which we called "transect" to follow ICES, even though it might be confusing with NIWA's past use of "transect" as a field for the transect number in a series of transect data acquisition.
- By listing which transects are in which files, the `t_file_transect` join table allows identifying the files that span a given transect. Then the start and end times in relevant transect/files can be used to find the exact boundaries of a given transect.
- transect_id in ICES category "Transect" was removed as it was redundant with transect_pkey.
- ICES category "Transect" has geographic information (northlimit, eastlimit, etc.) attributes. These were not used as attributes in the table `t_transect`, as all necessary information can be retrieved from `t_navigation`.

t_software

Comment on table: Instances of new acquisition software install on a new computer (or updating to new version).

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
software_pkey	integer	Primary key as assigned by the database loading software.	
software_manufacturer	text	Name of acquisition software manufacturer.	
software_name	text	Name of acquisition software.	
software_version	text	Version of acquisition software.	
software_host	text	Name of host machine where acquisition software is installed.	
software_install_date	timestamp without time zone	Date of software installation.	
software_comments	text	Free-text field for relevant information not captured by other attributes.	

t_file

Comment on table: Raw acoustic data files.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
file_pkey	integer	Primary key as assigned by the database loading software.	
file_name	text	Name of the raw acoustic data file.	
file_path	text	Path/location/folder of the primary record of the raw acoustic data file.	
file_start_time	timestamp without time zone	Start time of data in raw acoustic data file.	
file_end_time	timestamp without time zone	End time of data in raw acoustic data file.	
file_software_key	integer	Refers to relevant entry in `t_software` table for the software used to record the file.	
file_deployment_key	integer	Refers to relevant entry in `t_deployment` table for the deployment where file was recorded.	
file_comments	text	Free-text field for relevant information not captured by other attributes.	

`t_navigation`

Comment on table: Simplified navigation for a raw acoustic data file.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
navigation_pkey	integer	Primary key as assigned by the database loading software.	
navigation_time	timestamp without time zone	Date and time for this navigation entry.	
navigation_latitude	double precision	Latitude (decimal degrees).	
navigation_longitude	double precision	Longitude (decimal degrees).	
navigation_depth	double precision	Detected depth under transducer depth, if relevant (m).	
navigation_file_key	integer	Refers to relevant entry in `t_file` for the file for which this navigation data record is relevant.	
navigation_comments	text	Free-text field for relevant information not captured by other attributes.	

t_transect

Comment on table: Acoustic data transects.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
transect_pkey	integer	Primary key as assigned by the database loading software.	Transect/transect_id.
transect_name	text	Name of the transect.	Transect/transect_name.
transect_description	text	Description of the transect, its purpose, and main activity. NOTE: this will be the same as transect_type for now.	Transect/transect_description.
transect_related_activity	text	Describe related activities that may occur on the transect. NOTE: this should be linked to the transect_station.	Transect/transect_related_activity.
transect_start_time	timestamp without time zone	Start time of the transect.	Transect/transect_start_time.
transect_end_time	timestamp without time zone	End time of the transect.	Transect/transect_end_time.
transect_snapshot	integer	Snapshot number, to identify time repeats. Typically for statistical purposes, in acoustic surveys.	
transect_stratum	text	Geographical stratum number or name, to identify geographically different areas. Typically for statistical purposes, in trawl surveys.	

transect_station	text	Name or number of station, to identify separate targets. Typically, station numbers in trawl survey, or marks in acoustic surveys.	
transect_type	text	Type/purpose of transect, to identify categorically different data purposes, e.g. transit/steam/trawl/junk/test/etc.	
transect_number	integer	Incremental number of the transect for a same snapshot/type.	
transect_comments	text	Free-text field for relevant information not captured by other attributes.	Transect/transect_comments.

t_ancillary

Comment on table: Any other instruments or equipment potentially relevant to the acoustic dataset, e.g. GPS, pitch & roll sensor, anemometer, net systems, CTD, ADCP, etc.

Table attribute	*Type*	*Description*	*Relevant ICES category/attribute*
ancillary_pkey	integer	Primary key as assigned by the database loading software.	
ancillary_type	text	Type of ancillary instrument or equipment.	
ancillary_manufacturer	text	Name of the manufacturer of the ancillary instrument or equipment.	
ancillary_model	text	Model of the ancillary instrument or equipment.	
ancillary_serial	text	Serial number of the ancillary instrument or equipment.	
ancillary_firmware	text	Firmware version of the ancillary instrument or equipment, if applicable.	
ancillary_comments	text	Free-text field for relevant information not captured by other attributes.	

Join table `t_file_transect`

Comment on table: Join table to manage the many-many relationship between `t_file` and `t_transect`.

Table attribute	*Type*	*Description*
file_transect_pkey	integer	Primary key as assigned by the database loading software.
file_key	integer	Refers to relevant entry in `t_file`.
transect_key	integer	Refers to relevant entry in `t_transect`.
file_transect_comments	text	Free-text field for relevant information not captured by other attributes.

Join table `t_file_ancillary`

Comment on table: Join table to manage the many-many relationship between `t_file` and `t_ancillary`.

Table attribute	*Type*	*Description*
file_ancillary_pkey	integer	Primary key as assigned by the database loading software.
file_key	integer	Refers to relevant entry in `t_file`.
ancillary_key	integer	Refers to relevant entry in `t_ancillary`.
file_ancillary_comments	text	Free-text field for relevant information not captured by other attributes.

4.6 Relationships between the three main groups

As illustrated in the ERD of the acdb_2 template in Figure 4, the three groups are linked through:

- Table `t_deployment` for the LOGISTICS group.
- Table `t_setup` for the INSTRUMENTS group.
- Table `t_file` for the DATA group.

The rules between these tables are:

- 1 file contains 1 or several channels, that is, relate to 1 or several setups, while 1 setup is used for the acquisition of 0, 1 or several files (i.e. “many-to-many” relationship between `t_file` and `t_setup`).
- 1 file was acquired for 1 deployment, while 1 deployment resulted in the acquisition of 1 or several files (i.e. “many-to-one” relationship between `t_deployment` and `t_file`).

These rules imply one join table:

- `t_file_setup`, to manage the many-many relationship between `t_file` and `t_setup`.

Since any file is acquired during one deployment only, the relationship between `t_file` and `t_deployment` was implemented using a foreign key in the `t_file` table.

Join table `t_file_setup`

Comment on table: Join table to manage the many-many relationship between `t_file` and `t_setup`.

Table attribute	*Type*	*Description*
file_setup_pkey	integer	Primary key as assigned by the database loading software.
file_key	integer	Refers to relevant entry in `t_file`.
setup_key	integer	Refers to relevant entry in `t_setup`.
file_setup_comments	text	Free-text field for relevant information not captured by other attributes.

5 The “public” views

The public views reconcile information from the new ESP3 schema and the old ESP2 schema. They are designed as a view of other tables. Their main purpose is to give easy access to consult the data for any user, without putting any constraints on the database design.

The public views are virtual tables that can be categorized in three groups, following the terminology used for the ESP3 schema.

The public views are subject to change as the database is used and populated, as they can be created to fit any need. Basic views have been created to provide an overview of data to users, in a structure close to the ICES metadata convention.

5.1 LOGISTICS views in public schema

v_mission

This view is a simplified display of the `t_mission` table, gathering all attributes defined for the ICES category “Mission”:

- mission_name
- mission_abstract
- mission_start_date
- mission_end_date
- principal_investigator
- principal_investigator_email
- institution
- data_centre
- data_centre_email
- mission_id
- mission_platform
- creator
- contributor
- mission_comments

v_cruise

This view is a simplified display of the `t_deployment` table applicable to a cruise, gathering all attributes defined for the ICES category “Cruise”:

- cruise_name

- cruise_description
- cruise_summary_report
- cruise_area_description
- cruise_start_date
- cruise_end_date
- cruise_id
- cruise_northlimit
- cruise_eastlimit
- cruise_southlimit
- cruise_westlimit
- cruise_uplimit
- cruise_downlimit
- cruise_units
- cruise_zunits
- cruise_projection
- cruise_start_port
- cruise_end_port
- cruise_start_BODC_code
- cruise_end_BODC_code
- cruise_comments

v_mooring

This view is a simplified display of the `t_deployment` table applicable to a mooring deployment, gathering all attributes defined for the ICES category "Mooring":

- mooring_description
- mooring_depth
- mooring_northlimit
- mooring_eastlimit
- mooring_southlimit
- mooring_westlimit
- mooring_uplimit

- mooring_downlimit
- mooring_units
- mooring_zunits
- mooring_projection
- mooring_deployment_date
- mooring_retrieval_date
- mooring_code
- moorin_site_name
- mooring_operator
- mooring_comments

v_mission_deployment

The v_mission_deployment view is a simplified view joining the basic attributes of missions to the basic attributes of corresponding deployments. This view gives the user a rapid overview of the entire contents of the acoustic database (Figure 5). It has the following attributes

- mission_name
- deployment_name
- deployment_id
- deployment_start_date

	mission_name text	deployment_name text	deployment_start_date timestamp without time zone
1	Cook Strait and Chatham Rise juvenile hoki	tan9504 POAC01	1995-04-01 00:00:00
2	Cook Strait and Chatham Rise juvenile hoki	tan9610 POAC01	1996-09-21 00:00:00
3	Bounty Platform southern blue whiting	tan9910 SBW9801	1999-08-12 00:00:00
4	West Coast South Island hake	tan9911 HAK9801	1999-08-30 00:00:00
5	West Coast South Island hoki	tan0007 HOK9903	2000-06-26 00:00:00
6	Mernoo Bank scampi	kah0102 SCI2000/02	2001-01-29 00:00:00
7	Scampi / bottom classification	kah0102 SCI2000/02	2001-01-29 00:00:00
8	Lake Rotoiti trout	par0101 NZF01504	2001-02-01 00:00:00
9	Southern Lakes trout	par0101 NZF01504	2001-02-01 00:00:00
10	Chatham Rise seamounts	tan0104 SFB013	2001-04-13 00:00:00
11	Eastern Bay of Plenty seamounts	tan0107 SFA0133	2001-05-18 00:00:00
12	Mernoo Bank scampi	kah0109 SCI2000/02	2001-10-25 00:00:00
13	Scampi / bottom classification	kah0109 SCI2000/02	2001-10-25 00:00:00
14	Antartic toothfish	jan0201 ANT2002/01	2002-12-28 00:00:00
15	East Coast North Island scampi	kah0302 SCI2002/01	2003-02-17 00:00:00
16	East Coast North Island orange roughy	tan0310 ORH2002/01A	2003-06-16 00:00:00
17	East Coast South Island hoki	thh0301 IFL02301	2003-08-09 00:00:00
18	Campbell Rise southern blue whiting	aok0301 HMC04301	2003-09-01 00:00:00
19	East Coast Scampi	kah0401 SCI2003/02	2004-01-25 00:00:00
20	Chatham Rise orange roughy	tan0408 ORH2003/01	2004-07-07 00:00:00

Figure 5: The first 20 entries of the v_mission_deployment view table in the public schema.

5.2 LOGISTICS/INSTRUMENTS views in public schema

v_deployment_setup

This view shows the basic attributes of a setup and parameters for a given deployment. It allows a quick overview of past deployments and the systems that were used. It has the following attributes:

- deployment_id
- transceiver_manufacturer,
- transceiver_model,
- transceiver_serial,
- transducer_manufacturer,
- transducer_model,
- transducer_serial,
- platform_type,
- transducer_location_type,
- totransducer_orientation_type,
- parameters_pulse_mode,

- parameters_pulse_length,
- parameters_frequency_min,
- parameters_frequency_max,
- parameters_power

5.3 LOGISTICS/DATA view in public schema

The three following views allow a user to rapidly identify the files corresponding to transects in a given mission, to extract basic navigation for all transects within a mission, and to extract basic navigation for all files in a mission.

v_mission_file_transect

- mission_name
- deployment_id
- transect_name
- file_path
- file_name
- file_start_time

v_deployment_transect_navigation

- deployment_id
- transect_name,
- snapshot_number
- stratum_name
- station
- type
- number
- lat
- lon
- time

v_deployment_file_navigation

- deployment_id
- file_name,
- lat

- lon
- time

6 Loading data into the acoustic database

As previously discussed and illustrated in Figure 3, loading data to the database is operated in three consecutive steps:

- Step 1: Create a MINIDB database (built in SQLite from the acdb_2 template) on the local machine to hold metadata for the new entry.
- Step 2: Load the contents of a MINIDB database into the LOAD schema on the server, followed by data check.
- Step 3: Load the contents of the LOAD schema into the ESP3 schema, followed by final data check.

An application was created in MATLAB to assist a user to perform this sequence (Figure 6). It can be accessed as a compiled version (aka, not needing MATLAB to run) through the software ESP3 by clicking on the "Survey Data" tab of the menu and selecting "Acoustic DB tool".

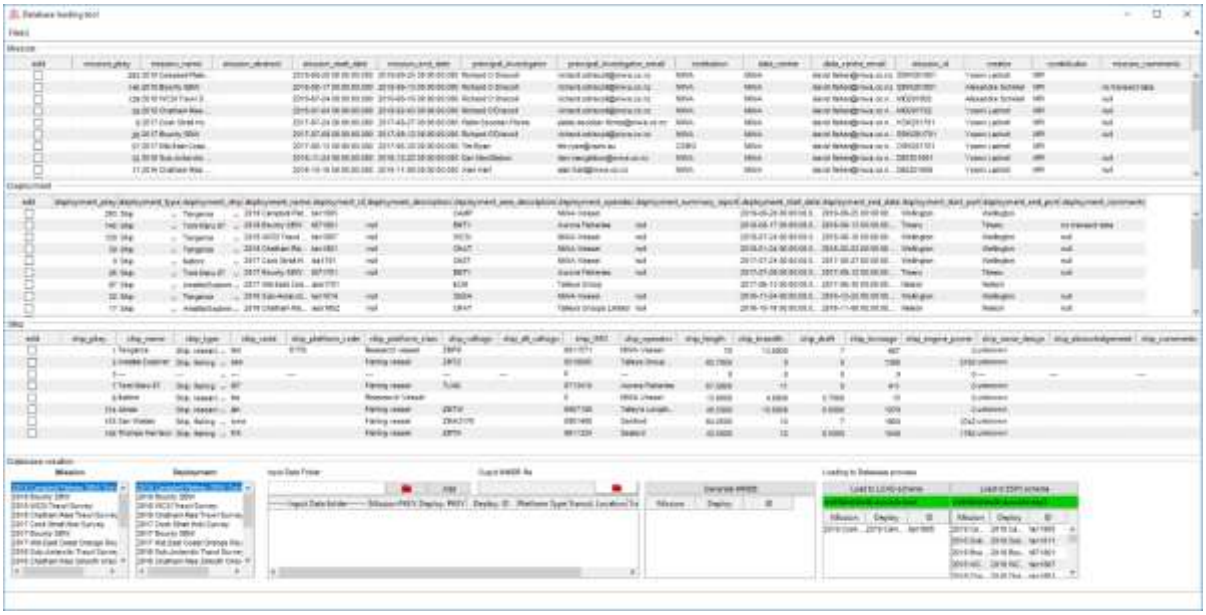

Figure 6: The user interface of the acoustic database loading software.

Step 1.1 (optional): Load existing entries from acoustic database

If you have an active connection to the acoustic database on the wellfisheriesdb server (right column in the "Loading to database process" section is showing green, see Figure 7), you can load the entries that are currently in the database, so you can reuse them. In the menu bar, choose **File(s)** and select **Import db file from ESP3 database** (Figure 7).

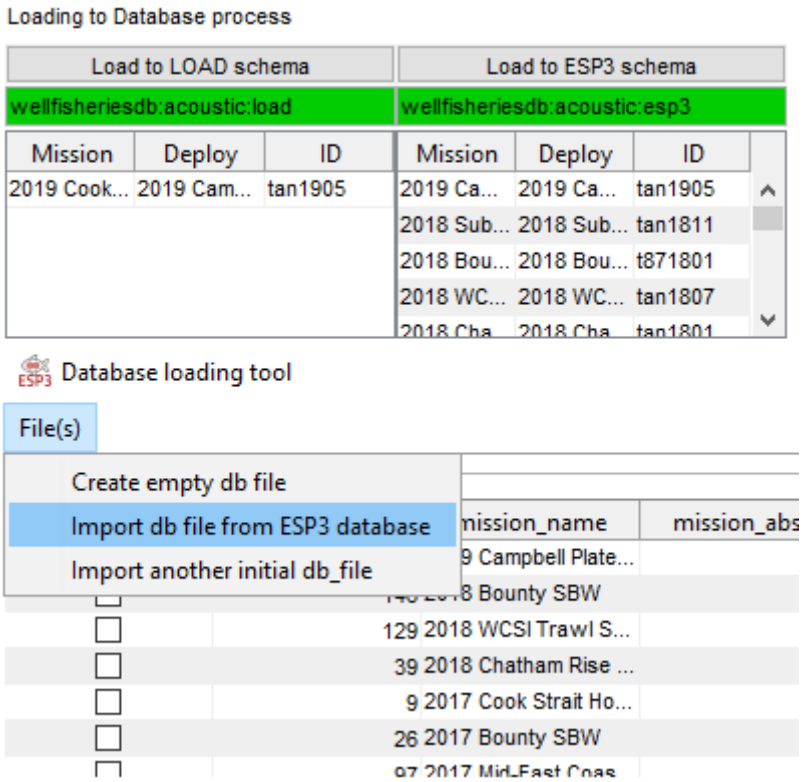

Figure 7: If the connection to the acoustic database is active (top), import the db file from the SP3 database from the Menu (bottom).

Step 1.2: Manually complete the necessary Ship, Deployment, and Mission entries

The three top panels in the interface show the entries for the Mission, Deployment and Ship tables that the software already know about, and which you may have just updated. Each table must contain the appropriate entries, so you must check if the desired entries are already listed, and if not, add it.

Note that a Deployment entry has a Ship entry as an attribute, so you must define the Ship entry before defining a Deployment entry that references it.

To add a new entry, right-click on the appropriate panel, select “add” to add a new entry, then check the “edit” box on the left, and complete the attributes (Figure 8).

Note that you can load more than just one deployment or mission to the acoustic database in a single step, so fill in all mission, deployment, and ship entries you want.

When the desired entries for those three tables exist, move on to the next step.

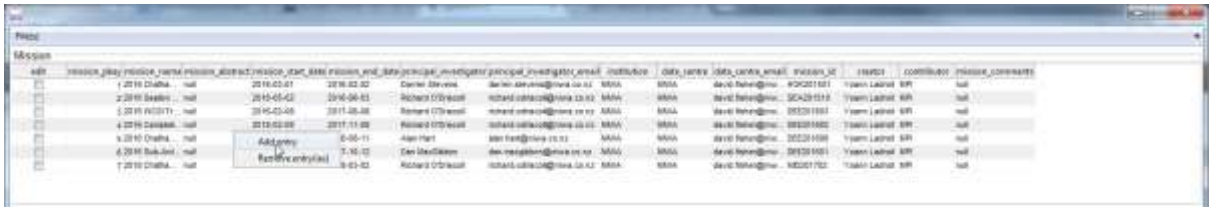

Figure 8: Adding a new entry to the Mission panel

Step 1.3: List the corresponding missions, deployments, and data folders

A MINIDB database is a single-schema SQLite database based on the acdb_2 template that can be created from one or several datasets and saved into a file with extension .db on the local machine running the software. It does not require to establish a connection to the acoustic database and can thus be run while in the field.

Once you have ensured that all needed mission, deployment and ship entries exist, go to the bottom panel "*Database creation*" of the interface to associate the corresponding Missions, Deployments and data folders.

For each folder of raw data (Figure 9):

- Select the corresponding mission and deployment in the lists on the left;
- Click on the folder button of the **Input Data Folder** section and navigate to the data folder (the one containing the raw data and logbook file);
- Click "Add";
- In the new line added in the table, specify in the drop-down menus the platform type, transducer location, and transducer orientation.

Since a typical cruise survey running several systems on different platforms (Hull, towbody, AOS) would have data in several folders, add all relevant data folders this way.

You can also add data folders from different pairs of Mission/Deployments too.

Figure 9: Selecting the appropriate folder of data

Step 1.4. Generate the MINIDB file

When all data folders are listed and their platform type, transducer location and orientation informed, specify an output MINIDB file and click the "*Generate MINIDB*" button to create the MINIDB database (Figure 10).

Note that if the output file you have specified already exists and contains data, that data will appear in the table under the "*Generate MINIDB*" button, and that if you press the generate button, the file will be overwritten and any data contained in it will be lost.

Upon pressing the "*Generate MINIDB*" button, the program creates the empty database file, and goes through the data folder(s) specified to find the raw data files, load them, and extract the relevant metadata from them. The transect information data will be extracted from the logbook file that is in the folder if it exists. If not, transect information will not be populated for your dataset.

Note that this process may take a long time as the software needs to open all raw data files to retrieve relevant information. A progress bar at the right of the bottom panel shows the progress of the operation.

The MINIDB database creation process can be followed in the command window and will show warning and error reports if any of the tables cannot be loaded properly. If any warning or error occurs, these MUST be fixed before moving on to the next step.

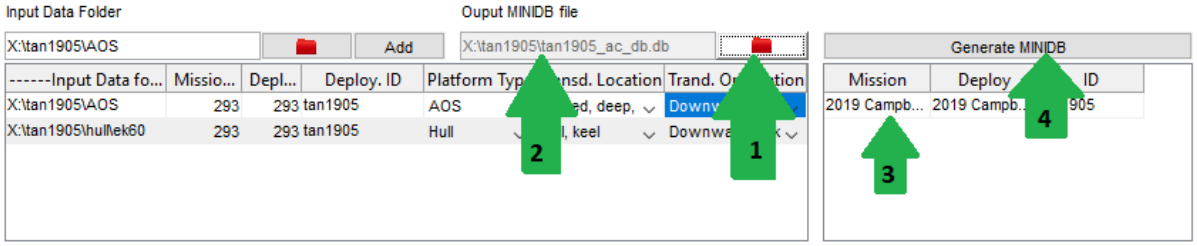

Figure 10: Select the output MINIDB file location. If the file already exists, the contents will show in the right panel. To generate (or overwrite) the MINIDB, click Generate MINIDB

Step 1.5. Inform the missing serial numbers in the MINIDB

For each line in the MINIDB, right-click and select **Edit/View Setups** and complete the missing serial numbers for the transceivers and/or transducers (Figure 11).

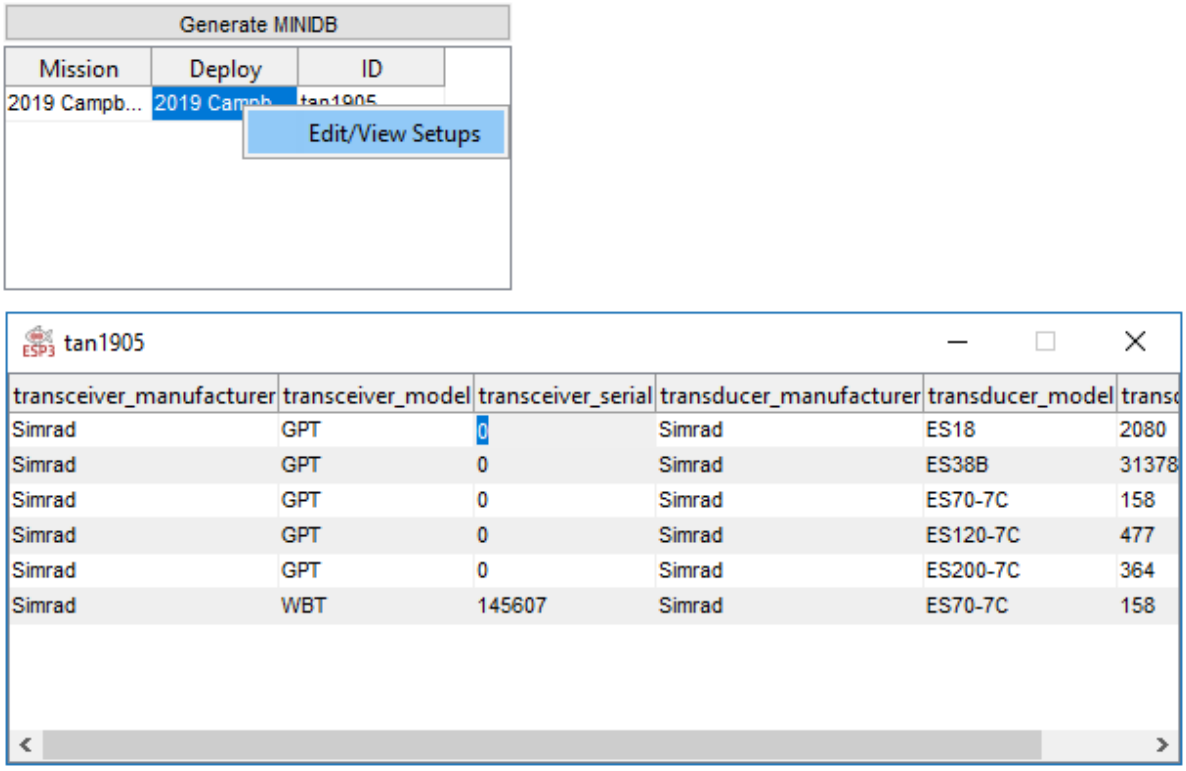

Figure 11: Inform the missing serial numbers in the MINIDB

Step 2.1. Load to the LOAD schema

The LOAD schema on the server is the intermediary schema in the acoustic database. Its location on the server is indicated under the "*Load to LOAD schema*" button. If the connection to the server is operational, the text will be shown on a green background and the loading button will be enabled. If the connection to the server cannot be made, the text will be displayed on a red background and the loading button will be unavailable (Figure 12).

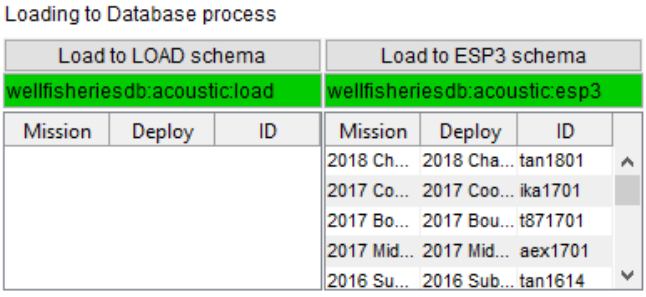

Figure 12: Right part of the bottom panel showing the contents of the LOAD and ESP3 schema on th server, connection status (green for active), and buttons at the top to start loading.

If the connection to the server is active, a table shows under the connection box displaying the Mission/Deployment combination that have already been loaded in the database.

If the connection to the LOAD schema on the server is active (green background), press the "*Load to Load schema*" to load the contents of the MINIDB listed in the "*Output MINIDB file*" box into the LOAD schema. The software starts with discarding any data that may be remaining

in the LOAD schema from a previous load on the destination tables whose location is specified in the text box under the button. Then the software proceeds with the loading of the data.

Note that the data loading process can be followed in the command window and will show warning and error reports if any of the tables cannot be loaded properly. If any warning or error is given during this step, the issue must be fixed and the process repeated before moving to the next step.

Step 2.2. Verify the contents of the LOAD schema

Before moving onto the next step, it is good practice to open a few of the tables as well to check that the data have gone through properly and are mirroring the initial database file. The local MINIDB file can be opened using the open source software sqlitebrowser[5], while the contents of the LOAD schema on the PostgreSQL database can be accessed using the software pgAdmin[6].

Step 3.1: Load to the ESP3 schema

The process for loading in the ESP3 schema is similar to the previous one. If the connection to the ESP3 schema is active (green light), press the "*Load to ESP schema*" button to load the contents of the LOAD schema into the ESP3 schema.

The main difference with the previous step is that the software will not clear the destination schema before starting the loading. Also, once the loading is done, a backup of the table that has just been loaded will be created in the LOAD schema under the name `t_[original-table-name]_[load-identifier]`. This identifier will then be added to the `t_load` table in the LOAD schema alongside the user who did the load and the date/time at which it was done. During the loading process, the software updates primary keys of all tables, and propagates the changes to foreign keys of other tables as well as to join tables before making the final insert in the ESP3 schema, to ensure the coherence of the entire contents of the ESP3 schema.

Step 3.2: Verify the contents of the ESP3 schema

As before, use a software to verify the updated contents of the ESP3 schema.

[5] http://sqlitebrowser.org/
[6] https://www.postgresql.org/

7 References

ICES. 2016. A metadata convention for processed acoustic data from active acoustic systems. Version 1.10, SISP 4 TG-AcMeta, ICES WGFAST Topic Group, TG-ACMeta.

Macaulay G, Dunford A. 2016. Database documentation for the Ministry of Fisheries Acoustics database. NIWA documentation. Version 1.2 June 2016.

MacLennan DN, Simmonds EJ. 1992. Fisheries Acoustics. Chapman & Hall, London. Fish and Fisheries Series 5.325 p.