



# Dam database

DIASPARA WP3.3 final report

Briand Cédric, Oliviero Jules,  
Helminen Jani

February 12, 2026



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Obstruction place</b>	<b>3</b>
<b>3</b>	<b>Physical obstruction</b>	<b>7</b>
<b>4</b>	<b>Assessment of passability</b>	<b>10</b>
<b>5</b>	<b>Hydro Power Plants</b>	<b>13</b>
<b>6</b>	<b>Turbines</b>	<b>16</b>
<b>7</b>	<b>Annex 1 : sql code users</b>	<b>19</b>
<b>8</b>	<b>Annex 2 : Nomenclature tables</b>	<b>21</b>
<b>9</b>	<b>Annex 3 : Dam tables</b>	<b>70</b>
<b>10</b>	<b>Annex 4 : Country dam table example</b>	<b>100</b>
<b>11</b>	<b>Annex 5 : Database ER diagram</b>	<b>108</b>

Foreword : this document is best viewed on the [diaspara website](#) in html

# Chapter 1

## Introduction

River fragmentation induced by human obstacles is a main threat to diadromous species. By impairing their migrations in rivers, they hinder their access to critical growth or reproduction habitats and potentially impair the achievement of their life cycle. Turbines at hydropower stations, pumps are also a direct source of mortality of fish during their downstream migration, or their move within the rivers.

In view of this and in a context of an ecosystem approach of fisheries management, there is a clear need to account for the impact of obstacles on the status of the species and to develop tools to prioritize restoration actions. To do so, a prerequisite is the availability of data on the location of dams as well as a set of characteristics related to their impacts, but those data are scattered, not easily available and not standardised.

This database proposes a database structure that could be used to collect and store a minimal set of data across Europe to quantify the impact of dams and weirs on diadromous fish stocks.

The work is based on an the DBEEL data database from the the POSE project (Walker et al. 2011) that was developed to handle all threats for eel, and was latter extended during the (SUDOANG project)[<https://sudoang.eu/en>]. Here, we develop a simplified elements by removing all elements that were judged not necessary in the database structure to answer the core of the question. This simplification is critical to then simplify the data collection at a large spatial scale. The code of the table in the national database is provided.

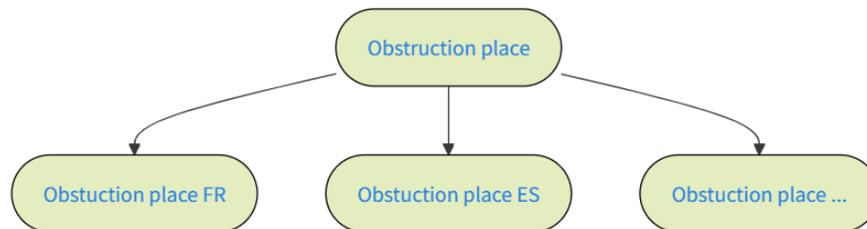
## Chapter 2

# Obstruction place

Each dam is characterized by a location which forms the top table (obstruction\_place) (Figure 2.1). For some complex river systems (braided rivers, branches ...) it is sometimes necessary to link the dams together, thereby identifying the main dam structure used to calculate cumulated impact. Indeed if two dams are in two parallel branches of the river, for upstream migration the cumulated impact is only based on the height and characteristics of the main dam as otherwise we would enter similar height twice. For downstream migration, the flow is split within the dam complex, using the main dam, and all dams attached to it.

### Inherited tables

To handle the updates of data, the database uses inheritance. The data are entered in the country schema, in this schema all tables inherit from the tables in the main dam schema (?@fig-dam\_db\_opin). When updates are collected, one can simply upload all tables again in the country schema. Other columns can be added at the national R/Diaspara\_WP3.3\_dam\_report\_html.qmd level to collect additional information.



Currently the data reporters is identified in EDMO (European Directory of Marine Organisations) (Table Table 2.1), as for the first data call the source of data will probably be the institute reporting at the national level. The

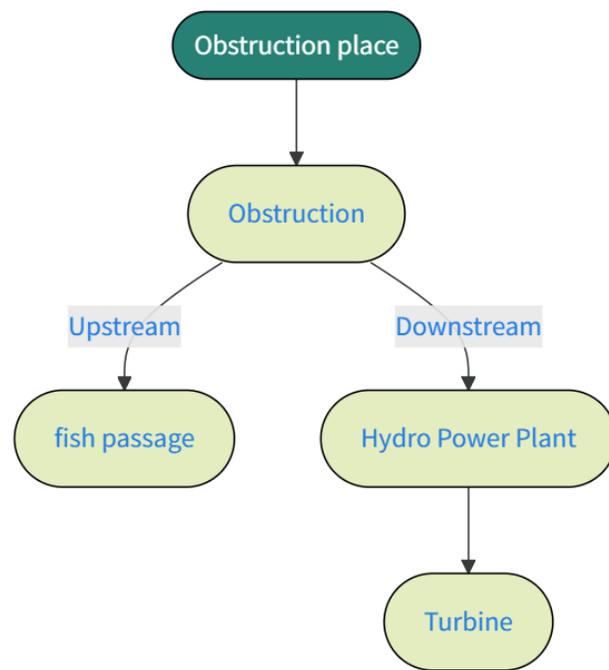


Figure 2.1: Obstruction place

development of a dam database in the Environment Agency will probably change this.

```
EDMO <- getCodeList("EDMO")|>
slice_head(n=10) |>
flextable() |>
  bold(bold = TRUE, part = "header") |>
  autofit() |>
  theme_booktabs(x, bold_header = FALSE)
EDMO
```

Id	Guid	Key	Description
144,695	9529db5e-91fa-4a64-ba9e-30cd22bd9f5a	1	University of Birmingham,
144,704	ff15bb71-0081-45c2-a1ae-de11c75e0161	10	University of Liverpool, Oc
144,794	d0a18e16-e2e9-4eb2-ad9d-e8b2653b7466	100	Leibniz Institute for Baltic S
146,664	7003daa4-5fff-407b-88ad-484f6098bf9e	1000	CQEL Dieppe, Service Ter
146,665	9001fbed-c862-4f11-a8a7-5ec696eb1b19	1001	Ifremer, Station d'Arcachon
146,666	6d9e408e-11dc-4439-a8b7-e487ad256542	1002	Aquitaine Observatory of S
146,667	9ae57011-5369-4218-bacc-803a6ee2b8f3	1003	Department of Agriculture
146,668	98c1edf3-dab4-4412-b6ae-d1b80eaaf19d	1004	Hellenic Navy Hydrograph
146,669	1754ec0a-b13b-4a4f-a890-e3f9f26ad508	1006	APAT of Venice (ex Istituto
146,670	b41cd346-355e-4e40-84ca-d1d919427a74	1007	Comune di Venezia, Centr

The obstruction place table is illustrated in Table Table 2.2.

Table 2.2: Dam obstruction place table

op_placename	op_op_id	op_id_original	op_country
ancien moulin de Gizaucourt	NA	ROE1	FR
Barrage du plan d'eau de Terville	NA	ROE3	FR
déversoir ancien moulin de Dampierre	NA	ROE4	FR
Ancien moulin de Somme Yèvre	NA	ROE5	FR
RADIER LE TERME LA NOISE	NA	ROE6	FR
VANNAGE DE LA FERME PLAGNICOURT	NA	ROE7	FR
seuil de roussy	NA	ROE9	FR
NA	NA	ROE10	FR
écluse n°14 de Thionville	NA	ROE12	FR

Ancien moulin des Ormes	NA	ROE15	FR
Ancien moulin de Dompierre	NA	ROE16	FR
Ancien moulin du bas de Verrières	NA	ROE17	FR
Passage busé	NA	ROE18	FR
Ancien seuil du moulin de Guinglange	NA	ROE19	FR
Ancien seuil du moulin d'Elvange	NA	ROE20	FR
moulin de petite hettange 2	NA	ROE37	FR
ANCIEN MOULIN DE ST MARD-SUR-AUVE	NA	ROE21	FR
Barrage de Homécourt	NA	ROE26	FR
Enrochements sur l'Orne amont Aboué	NA	ROE28	FR
Seuil de Serry	NA	ROE29	FR

---

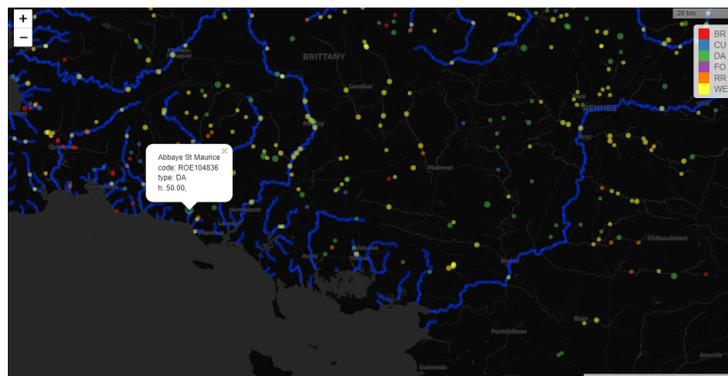


Figure 2.2: Map of a subset of dams in France

## Chapter 3

# Physical obstruction

This table (Table 3.1) is linked to the physical location of the dam and store meta-data on the dam characteristics.

It can be summarized as following :

- Physical obstruction has date of start and end
- Any changes are linked to Events (Erasurement, change of height change dam period)
- There might be several fishway types
- It includes information on downstream water depth (a key parameter to assess the passability of an obstacle)

ob_id	ob_starting_date	ob_ending_date	ob_op_id
f13052f3-db3b-48a8-9993-d31964091f87	2020-12-31	2035-01-01	9fe63686-7fcd-491d-b5ab-
969afa28-a661-41e5-93ab-21de9009739c	1800-01-01	2020-12-31	9fe63686-7fcd-491d-b5ab-
a25944ac-e77d-4663-a468-4364f04f9c12	2020-12-31	2020-12-31	9fe63686-7fcd-491d-b5ab-
8e41a7f2-453c-4b05-8ac4-ccd22931b661	1800-01-01	2035-01-01	06164ff9-7a9e-45ee-a71c-e
77f76cdd-1c8b-4680-8d30-a70102bef8e8	1800-01-01	2021-12-31	f973324d-6fbe-4888-8bb2-
1b08bac0-df44-4fcb-b17a-01a9092e8226	2021-12-31	2035-01-01	de3e0a09-72c5-49b6-9e85-
8e4dfa18-0cc6-4960-b326-90c53edfaa8e	1800-01-01	2021-12-31	de3e0a09-72c5-49b6-9e85-
5efff8a2-262a-464e-94a3-b59a0659ac75	1800-01-01	2021-12-31	698627bb-2417-44b8-8717-
bfa7758c-c869-4422-833b-9ebda93ae96d	1800-01-01	2020-12-31	93bf81fa-c753-4a61-8a34-c
9513f182-f5aa-4918-9083-52f7edcae4f3	1800-01-01	2035-01-01	4531c03c-ec8e-4032-be89-
21842ac7-adf3-4eca-b474-7fe7cc6a7a09	1800-01-01	2035-01-01	2ed703c0-e691-43e4-ac34-
a6643122-cbf9-42be-916d-798d14b92b4b	1800-01-01	2035-01-01	b6df6160-af5b-4e20-aaaf-9

2ef87036-bcfc-4902-a070-18db38d62a0c	1800-01-01	2035-01-01	e4a91d10-beee-488c-810a-
d87cb912-2e87-4ad7-aa4a-e60ab7b23c9f	1800-01-01	2035-01-01	385c0896-d21a-46f9-b7b5-
5eba1864-8c3e-4aa2-bea4-2d1e43ee3ef2	1800-01-01	2035-01-01	478244dd-973c-4693-9010-
bc50629a-8f58-47e7-a5fd-e7bbec8ab730	1800-01-01	2035-01-01	5c8c2cb6-e7b7-4e42-97f2-a
c56a6339-3b5e-4349-b5da-9125ffdd6812	1800-01-01	2035-01-01	c90fc976-489c-4538-8540-c
0d3b053e-af49-45ac-9e8a-61bc1715f605	1800-01-01	2035-01-01	4a503882-fa70-47cb-8ad4-
e12e3137-ce04-48cd-ae62-0dc482854d4d	1800-01-01	2035-01-01	74c18d6f-9a26-4843-900d-
81d22ebb-b6f3-4c32-8bb4-5eed36882e6f	1800-01-01	2035-01-01	a636cf04-79f1-4b4d-926b-

---

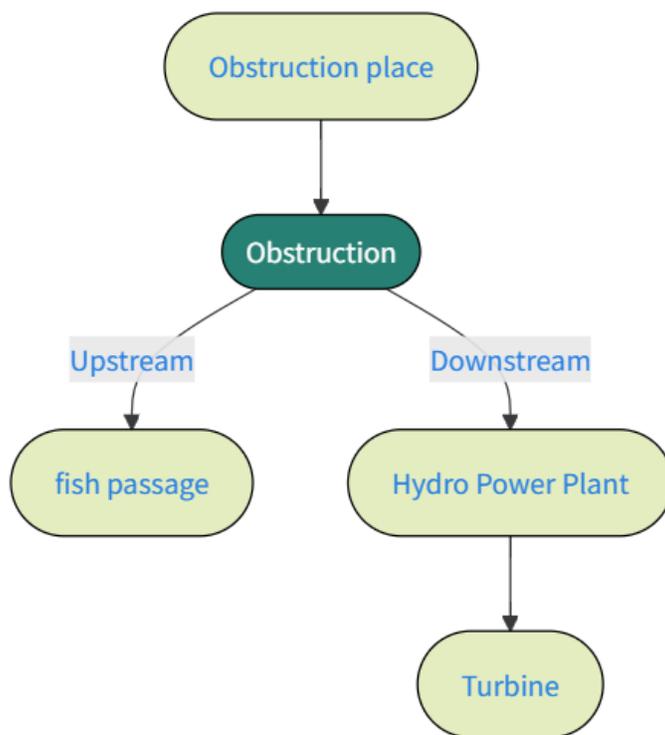


Figure 3.1: Simplified structure of the dam database, the physical obstruction

## Chapter 4

# Assessment of passability

The table (Table 4.1) is called fishway but in practise refers to fish passage, as it evaluates for each species whether the dam is passable, and this is often related to an expertise of fishway type. Basically it says whether a fishway is efficient for each species, a fishway designed for Salmon might not be suitable for Shad (Figure 4.1).

Table 4.1

fi_id_original	no_name	fi_presence_pass	fi_date
ROE23460	Alosa alosa	FALSE	NA
ROE23460	Anguilla anguilla	FALSE	NA
ROE23460	Anguilla_anguilla (glass eel)	FALSE	NA
ROE23460	Petromyzon marinus	FALSE	NA
ROE23460	Salmo salar	FALSE	NA
ROE108809	Alosa alosa	FALSE	NA
ROE108809	Anguilla anguilla	FALSE	NA
ROE108809	Anguilla_anguilla (glass eel)	FALSE	NA
ROE108809	Petromyzon marinus	FALSE	NA
ROE108809	Salmo salar	FALSE	NA
ROE22132	Alosa alosa	FALSE	NA
ROE22132	Anguilla anguilla	FALSE	NA
ROE22132	Anguilla_anguilla (glass eel)	FALSE	NA
ROE22132	Petromyzon marinus	FALSE	NA
ROE22132	Salmo salar	FALSE	NA
ROE74657	Alosa alosa	FALSE	NA
ROE74657	Anguilla anguilla	FALSE	NA
ROE74657	Anguilla_anguilla (glass eel)	FALSE	NA

ROE74657	<i>Petromyzon marinus</i>	FALSE	NA
ROE74657	<i>Salmo salar</i>	FALSE	NA

---

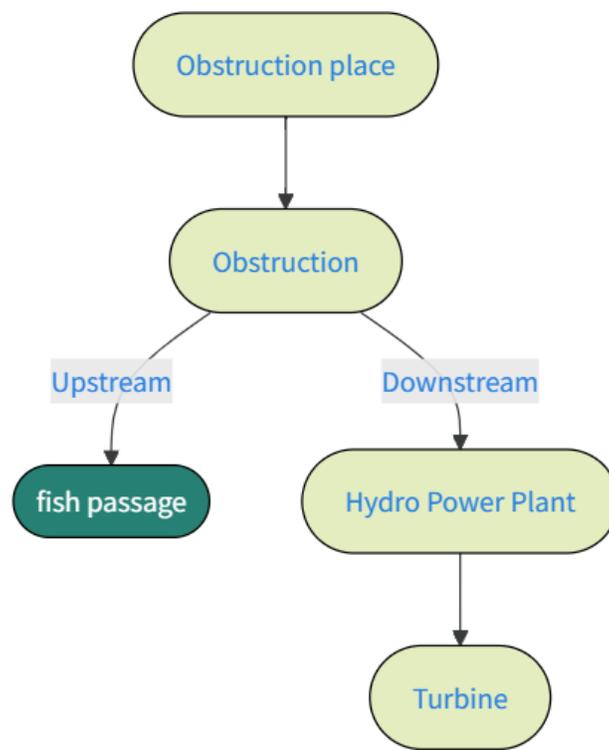


Figure 4.1: Simplified structure of the dam database, assessment of passability

## Chapter 5

# Hydro Power Plants

The elements in hydro power plants (Figure 5.1, Table 5.1) are necessary to make a calculation of the mortality at the dam level, according to the type of turbine, equipment flow and scenarios of flows at migration (see deliverable d2.2.1 from sudoang here [https://sudoang.eu/wp-content/uploads/2021/04/Deliverables-2-1-1\\_2-2-3-French-2.pdf](https://sudoang.eu/wp-content/uploads/2021/04/Deliverables-2-1-1_2-2-3-French-2.pdf)).

hpp_id	hpp_ob_id	hpp_name
632afe4c-fb4b-45c0-85f3-072c77c0a29b	3e537ead-1fad-4f36-9c7d-39764f5d066c	usine de REVIN
d984b863-8c28-4e47-a097-b615fcc8f8fa	61a4bb6e-6e3f-4c9d-bcc9-6043f9cbaa06	usine de REVIN
8a6547a2-1479-4f52-8a0d-621186ee7dcd	11ba38c9-cb55-482a-8f56-15b97e8ab6f8	D - LE VIEUX MOUL
298bd134-b3c9-487f-a547-4feb4202529f	c83ed548-0443-485c-9617-f1e452414ed0	D - LE VIEUX MOUL
6ad6e72a-d4e9-49b0-8d41-6efbba5f4393	6e054baa-b17e-4dc2-b93f-47fa5acbb2ca	Barrage de Saverdun (s
95058cdb-e8ee-49fa-8129-4a3a7f56fe85	fc6ba6f3-8b8c-4d0c-a178-4f21537d89f0	Barrage de Saverdun (s
f778a93c-3d37-44d9-a09c-66be2f112a8a	fe5307b9-dba4-4109-94c3-439b2b7863da	Barrage de Saverdun (s
35cc2907-294e-41ff-99b6-a66ab6653729	783c8967-b564-405e-b951-887804f69e4e	Usine de Mazères
6611162f-8992-444a-92f9-351b1fe85afc	7d618980-c810-47e9-8a2e-c2694d657856	Usine de Mazères
962a55b8-0710-4f80-a064-00a217887b6c	963bcb42-75e1-4b62-8905-404f26c35f79	Usine de Mazères
40d900e3-0e45-4deb-af4c-f5b81afb636f	3a82c16e-cecf-4da4-9927-ebd27c3abba5	chaussée de Moulin Ne
c9d45057-9713-4646-922a-d9b1bf86b9b3	66171ce4-1dff-4eb2-934a-cd1a9d4998b1	Seuil de Villemur sur T
a3e69aaa-2e85-4f64-83c4-12ae9f6dbcc7	3ade113f-455c-4484-a408-1ad31edbe275	Seuil de Villemur sur T
5f4d0907-906c-4790-9823-9f2b7f97df4c	1adceb2f-b0f0-444f-90dd-ebba7246191e	Grépiac
ef08eb27-7296-4c39-84b9-c00a214894db	62fe9ba5-15b1-4b5d-8f57-b73969042149	Centrale des Chatelles
3fb5d6e5-f865-4ce8-93af-50446aa79e56	353ad55c-f9c5-4b4c-90b1-abfeb182173c	Centrale des Chatelles
989973d0-4ee4-43c5-bd5a-50469b24403b	768894d9-83fd-4edf-85ff-a7b4ca5a9aaa	Moulin de Kery
ff98c644-57e1-487c-ba6a-f5da386e952a	441e7546-594c-49c8-9b1e-c1749e34d77b	Seuil de l'île d'Amour

e7af4287-d7e0-4e50-88ac-abf5d02ea799  
dfcbb806-73cf-4124-be35-f9b3254eac6f

8b1c1fbb-bf75-4915-a409-37d79f8cdbf1  
a5d1a251-4c69-4add-abc9-c95362848bab

Seuil de l'île d'Amour  
Seuil de l'île d'Amour

---

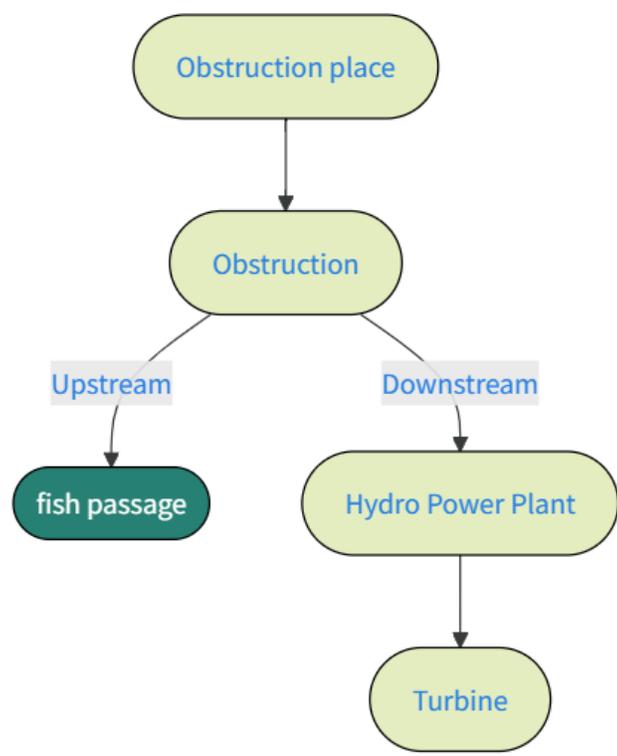


Figure 5.1: Simplified structure of the dam database, hydro power plants

## Chapter 6

# Turbines

In addition to the data stored in the hydropower plant (Figure 5.1), the turbine database store key characteristics of turbines that control the level of induced mortality.

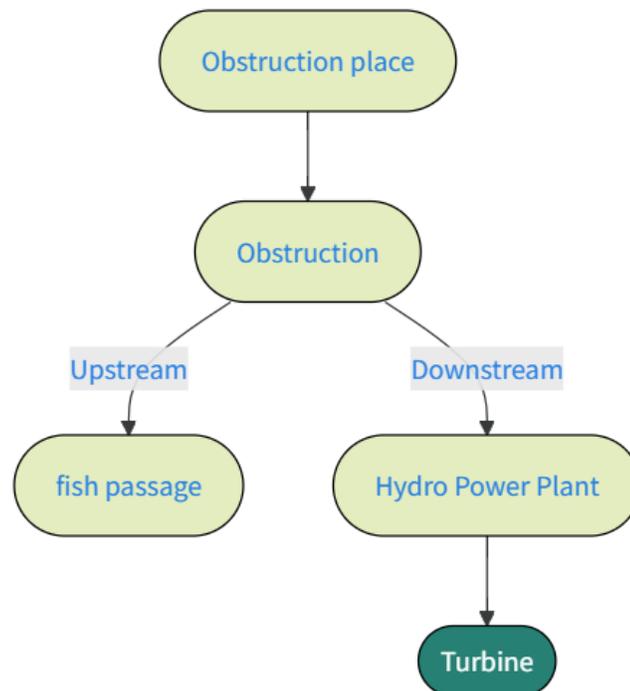


Figure 6.1: Simplified structure of the dam database, Turbine table

turb_id	turb_hpp_id	turb_turbine_type_no
86c7ef69-0c0d-4ddb-947e-91f94908e3f0	50231df6-eda6-40ca-8fb7-d28f0556c324	
5cfe89db-7960-4390-8fe6-0818e1f256ae	50231df6-eda6-40ca-8fb7-d28f0556c324	
d8f5aa76-e3a0-4433-9d49-ae906ebd0740	fb419f34-c603-4c09-92bb-686965efc8f5	
4a2aff8d-3a61-4a4e-bc26-62be93951742	76742279-71c0-4cec-9baa-f204b4d0b9ad	
a42dca50-c7c2-4393-9c41-13a1a31b067b	83829fc2-abc3-4647-9b4b-2620e94def20	
baa0ffb9-3386-4d39-86b6-dae2aa33a359	83829fc2-abc3-4647-9b4b-2620e94def20	
9a0fac44-43b6-40c6-802d-1b0aa9e5e7c4	1c823c34-5b3f-4ef3-8665-de4c538b7c3e	
6293913c-2e57-4dfd-bd0f-355cf63ce8f5	632afe4c-fb4b-45c0-85f3-072c77c0a29b	
206e75a8-b0c0-416c-b28d-46f394715b67	d984b863-8c28-4e47-a097-b615fcc8f8fa	
67999f8b-217b-4942-9906-981e65a6cc87	632afe4c-fb4b-45c0-85f3-072c77c0a29b	
98864bd6-d36b-418d-bfae-d5010cbae8da	d984b863-8c28-4e47-a097-b615fcc8f8fa	
f513c41d-fac7-4fd4-8e6e-29f31fbeatf4	632afe4c-fb4b-45c0-85f3-072c77c0a29b	
a19f4777-f82f-497b-b0ca-56f6c3a5cfe8	d984b863-8c28-4e47-a097-b615fcc8f8fa	
6dd3ce2b-02e0-4242-a7e4-69b1824bebdd	adcf6693-b8f0-43cf-9dcc-1ba7d1a9b47c	
dd2a1236-90bf-42ef-87f5-e71d3472c3d6	078fd54c-b70f-413e-a50a-054b3fb9bc94	
60cc34f7-060e-4e96-ac40-aad6916e2a2c	b8328629-6a19-40b0-ae4e-f702acdb991e	
66dd5d66-de45-4105-a109-921c7b7da9e1	b97e657a-82d5-4151-945a-a6771e9c8b48	
dfaaae23-8835-41b2-ba69-574c22159783	0b3eaf92-63eb-477f-977e-badddb9b55e3f	
dd98aee0-ea81-40b5-aa88-b80bae19e050	0b3eaf92-63eb-477f-977e-badddb9b55e3f	
4d754e59-3708-4b81-8878-e8e2aeff1ae8	33e10d42-7c6b-46cc-a7b2-9de7300d1ac2	

Table 6.2: Turbine type table

no_id	no_code	no_type	no_name
242	NA	Turbine type	Horizontal axis Kaplan (bulb)
243	NA	Turbine type	Other (please specify)
244	NA	Turbine type	Double Francis (spiral case)
245	NA	Turbine type	Francis unspecified
246	NA	Turbine type	Turbine with fixed blade propeller and vertical axis
247	NA	Turbine type	Kaplan not specified
248	NA	Turbine type	Pelton
249	NA	Turbine type	Reversible
250	NA	Turbine type	Kaplan (model of S-turbine)
251	NA	Turbine type	Turbine with fixed blade propeller and horizontal axis
252	NA	Turbine type	Unknown
253	NA	Turbine type	Vertical axis Kaplan
254	NA	Turbine type	Francis without volute
255	NA	Turbine type	Francis (spiral case)

256	NA	Turbine type	Banki-Michell (cross-flow)
257	NA	Turbine type	VLH
258	NA	Turbine type	Archimedean screw turbine
259	NA	Turbine type	Water wheel turbine (aqualienne)
260	NA	Turbine type	Water wheel turbine (others)
261	NA	Turbine type	Propeller unspecified

---

## Chapter 7

# Annex 1 : sql code users

SQL code to create user rights

```
-- Should be created only once
-- each role groups will appear and all databases
-- note the roless might already have been created on the server... so check ...

createDB -U postgres migdb

# TODO ON SERVER MODIFY pg_hba.conf

--- extension management
CREATE EXTENSION "uuid-oss" SCHEMA "public";
CREATE EXTENSION postgis SCHEMA "public";

GRANT ALL PRIVILEGES ON SCHEMA public TO diaspara_admin ;

-- Should be done each time the database is created
--- role management
GRANT CONNECT ON DATABASE damdb TO diaspara_read;
ALTER DATABASE damdb OWNER TO diaspara_admin;

-- grant roles for diaspara_read
GRANT USAGE
```

```
    ON SCHEMA dam
    TO diaspara_read;

GRANT EXECUTE
    ON ALL FUNCTIONS
    IN SCHEMA dam
    TO diaspara_read;

GRANT SELECT
    ON ALL TABLES
    IN SCHEMA dam
    TO diaspara_read;

GRANT USAGE
    ON SCHEMA nomenclature
    TO diaspara_read;

GRANT EXECUTE
    ON ALL FUNCTIONS
    IN SCHEMA nomenclature
    TO diaspara_read;

GRANT SELECT
    ON ALL TABLES
    IN SCHEMA nomenclature
    TO diaspara_read;

GRANT USAGE
    ON SCHEMA dam_france
    TO diaspara_read;

GRANT EXECUTE
    ON ALL FUNCTIONS
    IN SCHEMA dam_france
    TO diaspara_read;

GRANT SELECT
    ON ALL TABLES
    IN SCHEMA dam_france
    TO diaspara_read;
```

## Chapter 8

# Annex 2 : Nomenclature tables

The nomenclature has been adapted to the DIASPARA project using the vocab in stockDB.

SQL code to create nomenclature

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 14.20 (Ubuntu 14.20-0ubuntu0.22.04.1)
-- Dumped by pg_dump version 17.5

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET transaction_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: nomenclature; Type: SCHEMA; Schema: -; Owner: diaspara_admin
--
```

```

CREATE SCHEMA nomenclature;

ALTER SCHEMA nomenclature OWNER TO diaspara_admin;

--
-- Name: nomenclature_id_insert(); Type: FUNCTION; Schema: nomenclature; Owner: postgres
--

CREATE FUNCTION nomenclature.nomenclature_id_insert() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM no_id FROM "nomenclature".nomenclature WHERE no_id = NEW.no_id;
    IF FOUND THEN
        RAISE EXCEPTION '1- Invalid no_id (%)', NEW.no_id;
    ELSE
        RETURN NEW;
    END IF;
END
$$;

ALTER FUNCTION nomenclature.nomenclature_id_insert() OWNER TO postgres;

--
-- Name: nomenclature_id_update(); Type: FUNCTION; Schema: nomenclature; Owner: postgres
--

CREATE FUNCTION nomenclature.nomenclature_id_update() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM no_id FROM "nomenclature".nomenclature WHERE no_id = NEW.no_id;
    IF FOUND THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION '1- Invalid no_id (%)', NEW.no_id;
    END IF;
END
$$;

ALTER FUNCTION nomenclature.nomenclature_id_update() OWNER TO postgres;

```

```

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: nomenclature; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.nomenclature (
    no_id integer NOT NULL,
    no_code character varying(3),
    no_type character varying(60),
    no_name character varying(80)
);

ALTER TABLE nomenclature.nomenclature OWNER TO diaspara_admin;

--
-- Name: age; Type: TABLE; Schema: nomenclature; Owner: postgres
--

CREATE TABLE nomenclature.age (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.age OWNER TO postgres;

--
-- Name: biological_characteristic_type; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.biological_characteristic_type (
    bc_label character varying(100),
    bc_unit character varying(25),
    bc_data_type character varying(25)
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.biological_characteristic_type OWNER TO diaspara_admin;

--
-- Name: downstream_mitigation_measure; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

```

```

--
CREATE TABLE nomenclature.downstream_mitigation_measure (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.downstream_mitigation_measure OWNER TO diaspara_admin;

--
-- Name: effort_type; Type: TABLE; Schema: nomenclature; Owner: postgres
--

CREATE TABLE nomenclature.effort_type (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.effort_type OWNER TO postgres;

--
-- Name: electrofishing_mean; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.electrofishing_mean (
    ef_definition character varying(500)
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.electrofishing_mean OWNER TO diaspara_admin;

--
-- Name: event_change; Type: TABLE; Schema: nomenclature; Owner: postgres
--

CREATE TABLE nomenclature.event_change (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.event_change OWNER TO postgres;

--
-- Name: fishway_type; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin

```

```

--
CREATE TABLE nomenclature.fishway_type (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.fishway_type OWNER TO diaspara_admin;

--
-- Name: gear_type; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.gear_type (
    isscfg_code character varying(6),
    main_gear character varying(60),
    eel_specific_gear character varying(60)
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.gear_type OWNER TO diaspara_admin;

--
-- Name: maturity; Type: TABLE; Schema: nomenclature; Owner: postgres
--

CREATE TABLE nomenclature.maturity (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.maturity OWNER TO postgres;

--
-- Name: migration_direction; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.migration_direction (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.migration_direction OWNER TO diaspara_admin;

```

```

--
-- Name: nomenclature_no_id_seq; Type: SEQUENCE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE SEQUENCE nomenclature.nomenclature_no_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE nomenclature.nomenclature_no_id_seq OWNER TO diaspara_admin;

--
-- Name: nomenclature_no_id_seq; Type: SEQUENCE OWNED BY; Schema: nomenclature; Owner: diaspara_admin
--

ALTER SEQUENCE nomenclature.nomenclature_no_id_seq OWNED BY nomenclature.nomenclature_no_id_seq;

--
-- Name: obstruction_type; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.obstruction_type (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.obstruction_type OWNER TO diaspara_admin;

--
-- Name: TABLE obstruction_type; Type: COMMENT; Schema: nomenclature; Owner: diaspara_admin
--

COMMENT ON TABLE nomenclature.obstruction_type IS 'This table used to contain only physical

--
-- Name: orient_flow; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.orient_flow (
)

```

```

INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.orient_flow OWNER TO diaspara_admin;

--
-- Name: scientific_observation_method; Type: TABLE; Schema: nomenclature; Owner: diaspara_a
--

CREATE TABLE nomenclature.scientific_observation_method (
    sc_observation_category character varying(30),
    sc_definition character varying(500)
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.scientific_observation_method OWNER TO diaspara_admin;

--
-- Name: sex; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.sex (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.sex OWNER TO diaspara_admin;

--
-- Name: species; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.species (
    sp_vernacular_name character varying(30)
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.species OWNER TO diaspara_admin;

--
-- Name: stage; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

```

```

CREATE TABLE nomenclature.stage (
    spe_code character varying(3)
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.stage OWNER TO diaspara_admin;

--
-- Name: turbine_type; Type: TABLE; Schema: nomenclature; Owner: diaspara_admin
--

CREATE TABLE nomenclature.turbine_type (
)
INHERITS (nomenclature.nomenclature);

ALTER TABLE nomenclature.turbine_type OWNER TO diaspara_admin;

--
-- Name: age no_id; Type: DEFAULT; Schema: nomenclature; Owner: postgres
--

ALTER TABLE ONLY nomenclature.age ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.nomen

--
-- Name: biological_characteristic_type no_id; Type: DEFAULT; Schema: nomenclature; Owner: c

--

ALTER TABLE ONLY nomenclature.biological_characteristic_type ALTER COLUMN no_id SET DEFAULT

--
-- Name: downstream_mitigation_measure no_id; Type: DEFAULT; Schema: nomenclature; Owner: d

--

ALTER TABLE ONLY nomenclature.downstream_mitigation_measure ALTER COLUMN no_id SET DEFAULT n

--
-- Name: effort_type no_id; Type: DEFAULT; Schema: nomenclature; Owner: postgres
--

ALTER TABLE ONLY nomenclature.effort_type ALTER COLUMN no_id SET DEFAULT nextval('nomenclatu

```

```

--
-- Name: electrofishing_mean no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.electrofishing_mean ALTER COLUMN no_id SET DEFAULT nextval('nomenclature_electrofishing_mean_no_id_seq'::regclass);

--
-- Name: event_change no_id; Type: DEFAULT; Schema: nomenclature; Owner: postgres
--
ALTER TABLE ONLY nomenclature.event_change ALTER COLUMN no_id SET DEFAULT nextval('nomenclature_event_change_no_id_seq'::regclass);

--
-- Name: fishway_type no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.fishway_type ALTER COLUMN no_id SET DEFAULT nextval('nomenclature_fishway_type_no_id_seq'::regclass);

--
-- Name: gear_type no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.gear_type ALTER COLUMN no_id SET DEFAULT nextval('nomenclature_gear_type_no_id_seq'::regclass);

--
-- Name: maturity no_id; Type: DEFAULT; Schema: nomenclature; Owner: postgres
--
ALTER TABLE ONLY nomenclature.maturity ALTER COLUMN no_id SET DEFAULT nextval('nomenclature_maturity_no_id_seq'::regclass);

--
-- Name: migration_direction no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.migration_direction ALTER COLUMN no_id SET DEFAULT nextval('nomenclature_migration_direction_no_id_seq'::regclass);

--
-- Name: nomenclature no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin

```

```

--
ALTER TABLE ONLY nomenclature.nomenclature ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.no_id'::regclass);
--
-- Name: obstruction_type no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.obstruction_type ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.obstruction_type.no_id'::regclass);
--
-- Name: orient_flow no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.orient_flow ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.orient_flow.no_id'::regclass);
--
-- Name: scientific_observation_method no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.scientific_observation_method ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.scientific_observation_method.no_id'::regclass);
--
-- Name: sex no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.sex ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.sex.no_id'::regclass);
--
-- Name: species no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.species ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.species.no_id'::regclass);
--
-- Name: stage no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.stage ALTER COLUMN no_id SET DEFAULT nextval('nomenclature.stage.no_id'::regclass);

```

```

--
-- Name: turbine_type no_id; Type: DEFAULT; Schema: nomenclature; Owner: diaspara_admin
--

ALTER TABLE ONLY nomenclature.turbine_type ALTER COLUMN no_id SET DEFAULT nextval('nomenclat

--
-- Name: age age_id; Type: CONSTRAINT; Schema: nomenclature; Owner: postgres
--

ALTER TABLE ONLY nomenclature.age
    ADD CONSTRAINT age_id PRIMARY KEY (no_id);

--
-- Name: biological_characteristic_type bc_id; Type: CONSTRAINT; Schema: nomenclature; Owner:
--

ALTER TABLE ONLY nomenclature.biological_characteristic_type
    ADD CONSTRAINT bc_id PRIMARY KEY (no_id);

--
-- Name: downstream_mitigation_measure c_pk_downstream_mitigation_measure; Type: CONSTRAINT;
--

ALTER TABLE ONLY nomenclature.downstream_mitigation_measure
    ADD CONSTRAINT c_pk_downstream_mitigation_measure PRIMARY KEY (no_id);

--
-- Name: electrofishing_mean ef_mean_id; Type: CONSTRAINT; Schema: nomenclature; Owner: dias
--

ALTER TABLE ONLY nomenclature.electrofishing_mean
    ADD CONSTRAINT ef_mean_id PRIMARY KEY (no_id);

--
-- Name: effort_type effort_type_id; Type: CONSTRAINT; Schema: nomenclature; Owner: postgres
--

ALTER TABLE ONLY nomenclature.effort_type

```

```

ADD CONSTRAINT effort_type_id PRIMARY KEY (no_id);

--
-- Name: event_change event_change_id; Type: CONSTRAINT; Schema: nomenclature; Owner: postgres
--

ALTER TABLE ONLY nomenclature.event_change
    ADD CONSTRAINT event_change_id PRIMARY KEY (no_id);

--
-- Name: fishway_type fishway_type_id; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--

ALTER TABLE ONLY nomenclature.fishway_type
    ADD CONSTRAINT fishway_type_id PRIMARY KEY (no_id);

--
-- Name: gear_type gear_type_id; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--

ALTER TABLE ONLY nomenclature.gear_type
    ADD CONSTRAINT gear_type_id PRIMARY KEY (no_id);

--
-- Name: maturity maturity_id; Type: CONSTRAINT; Schema: nomenclature; Owner: postgres
--

ALTER TABLE ONLY nomenclature.maturity
    ADD CONSTRAINT maturity_id PRIMARY KEY (no_id);

--
-- Name: migration_direction migration_direction_id; Type: CONSTRAINT; Schema: nomenclature; Owner: postgres
--

ALTER TABLE ONLY nomenclature.migration_direction
    ADD CONSTRAINT migration_direction_id PRIMARY KEY (no_id);

--
-- Name: nomenclature nomenclature_pkey; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin

```

```

--
ALTER TABLE ONLY nomenclature.nomenclature
    ADD CONSTRAINT nomenclature_pkey PRIMARY KEY (no_id);

--
-- Name: obstruction_type obstruction_type_id; Type: CONSTRAINT; Schema: nomenclature; Owner:
--
ALTER TABLE ONLY nomenclature.obstruction_type
    ADD CONSTRAINT obstruction_type_id PRIMARY KEY (no_id);

--
-- Name: orient_flow orient_flow_id; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara
--
ALTER TABLE ONLY nomenclature.orient_flow
    ADD CONSTRAINT orient_flow_id PRIMARY KEY (no_id);

--
-- Name: scientific_observation_method sc_observ_method_id; Type: CONSTRAINT; Schema: nomen
--
ALTER TABLE ONLY nomenclature.scientific_observation_method
    ADD CONSTRAINT sc_observ_method_id PRIMARY KEY (no_id);

--
-- Name: sex sex_id; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.sex
    ADD CONSTRAINT sex_id PRIMARY KEY (no_id);

--
-- Name: species sp_id; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--
ALTER TABLE ONLY nomenclature.species
    ADD CONSTRAINT sp_id PRIMARY KEY (no_id);

```

```

--
-- Name: stage st_id; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--

ALTER TABLE ONLY nomenclature.stage
    ADD CONSTRAINT st_id PRIMARY KEY (no_id);

--
-- Name: turbine_type turbine_type_id; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--

ALTER TABLE ONLY nomenclature.turbine_type
    ADD CONSTRAINT turbine_type_id PRIMARY KEY (no_id);

--
-- Name: species uk_no_code; Type: CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--

ALTER TABLE ONLY nomenclature.species
    ADD CONSTRAINT uk_no_code UNIQUE (no_code);

--
-- Name: downstream_mitigation_measure tr_downstream_mitigation_measure_insert; Type: TRIGGER
--

CREATE TRIGGER tr_downstream_mitigation_measure_insert
BEFORE INSERT ON nomenclature.downstream_mitigation_measure
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_insert();

--
-- Name: downstream_mitigation_measure tr_downstream_mitigation_measure_update; Type: TRIGGER
--

CREATE TRIGGER tr_downstream_mitigation_measure_update
BEFORE UPDATE ON nomenclature.downstream_mitigation_measure
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_update();

--
-- Name: effort_type tr_effort_insert; Type: TRIGGER; Schema: nomenclature; Owner: postgres
--

```

```

CREATE TRIGGER tr_effort_insert
BEFORE INSERT ON nomenclature.effort_type
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_insert();

--
-- Name: effort_type tr_effort_update; Type: TRIGGER; Schema: nomenclature; Owner: postgres
--

CREATE TRIGGER tr_effort_update
BEFORE UPDATE ON nomenclature.effort_type
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_update();

--
-- Name: event_change tr_event_change_insert; Type: TRIGGER; Schema: nomenclature; Owner: postgres
--

CREATE TRIGGER tr_event_change_insert
BEFORE INSERT ON nomenclature.event_change
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_insert();

--
-- Name: event_change tr_event_change_update; Type: TRIGGER; Schema: nomenclature; Owner: postgres
--

CREATE TRIGGER tr_event_change_update
BEFORE UPDATE ON nomenclature.event_change
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_update();

--
-- Name: fishway_type tr_fishway_type_insert; Type: TRIGGER; Schema: nomenclature; Owner: postgres
--

CREATE TRIGGER tr_fishway_type_insert
BEFORE INSERT ON nomenclature.fishway_type
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_insert();

--
-- Name: fishway_type tr_fishway_type_update; Type: TRIGGER; Schema: nomenclature; Owner: postgres
--

```

```

CREATE TRIGGER tr_fishway_type_update
BEFORE UPDATE ON nomenclature.fishway_type
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_update();

--
-- Name: obstruction_type tr_obstruction_type_insert; Type: TRIGGER; Schema: nomenclature; O
--

CREATE TRIGGER tr_obstruction_type_insert
BEFORE INSERT ON nomenclature.obstruction_type
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_insert();

--
-- Name: obstruction_type tr_obstruction_type_update; Type: TRIGGER; Schema: nomenclature; O
--

CREATE TRIGGER tr_obstruction_type_update
BEFORE UPDATE ON nomenclature.obstruction_type
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_update();

--
-- Name: stage fk_species; Type: FK CONSTRAINT; Schema: nomenclature; Owner: diaspara_admin
--

ALTER TABLE ONLY nomenclature.stage
  ADD CONSTRAINT fk_species FOREIGN KEY (spe_code)
REFERENCES nomenclature.species(no_code) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: SCHEMA nomenclature; Type: ACL; Schema: -; Owner: diaspara_admin
--

GRANT USAGE ON SCHEMA nomenclature TO diaspara_read;

--
-- Name: TABLE nomenclature; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.nomenclature TO diaspara_read;

```

```

--
-- Name: TABLE biological_characteristic_type; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.biological_characteristic_type TO diaspara_read;

--
-- Name: TABLE downstream_mitigation_measure; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.downstream_mitigation_measure TO diaspara_read;

--
-- Name: TABLE electrofishing_mean; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.electrofishing_mean TO diaspara_read;

--
-- Name: TABLE fishway_type; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.fishway_type TO diaspara_read;

--
-- Name: TABLE gear_type; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.gear_type TO diaspara_read;

--
-- Name: TABLE migration_direction; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.migration_direction TO diaspara_read;

--
-- Name: TABLE obstruction_type; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

```

```

GRANT SELECT ON TABLE nomenclature.obstruction_type TO diaspara_read;

--
-- Name: TABLE orient_flow; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.orient_flow TO diaspara_read;

--
-- Name: TABLE scientific_observation_method; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.scientific_observation_method TO diaspara_read;

--
-- Name: TABLE sex; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.sex TO diaspara_read;

--
-- Name: TABLE species; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.species TO diaspara_read;

--
-- Name: TABLE stage; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.stage TO diaspara_read;

--
-- Name: TABLE turbine_type; Type: ACL; Schema: nomenclature; Owner: diaspara_admin
--

GRANT SELECT ON TABLE nomenclature.turbine_type TO diaspara_read;

```

```
--
-- PostgreSQL database dump complete
--
```

#### Data in nomenclature

```
-- *****
-- Fill damdb_nomenclature in damdb_France database
-- la syntaxe COPY FROM stdin ne fonctionne pas dans Dbeaver
-- changement en INSERT INTO.. Values
-- *****

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- TOC entry 8055 (class 0 OID 2558486)
-- Dependencies: 374
-- Data for Name: biological_characteristic_type; Type: TABLE DATA; Schema: damdb_nomenclature
--

INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(38, NULL, 'Biological characteristic type', 'Unknown', 'Unknown', 'Unknown', 'Unknown');
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(42, NULL, 'Biological characteristic type', 'Weight', 'Weight', 'g', 'real');
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(43, NULL, 'Biological characteristic type', 'Age', 'Age', 'year', 'integer');
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(44, NULL, 'Biological characteristic type', 'Sex', 'Sex', 'Dimensionless', 'nomenclature');
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(45, NULL, 'Biological characteristic type', 'Stage', 'stage', 'Dimensionless', 'nomenclature');
INSERT INTO damdb_nomenclature.biological_characteristic_type
```

```

(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(46, NULL, 'Biological characteristic type', 'Rate', 'rate', '%', 'real');
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(48, NULL, 'Biological characteristic type', 'Density', 'Density', 'nb/m²', 'real');
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(39, NULL, 'Biological characteristic type', 'Length', 'Total length', 'mm', 'real');
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(41, NULL, 'Biological characteristic type', 'Upper length', 'Total length - upper bo
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(40, NULL, 'Biological characteristic type', 'Lower length', 'Total length - lower bo
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(47, NULL, 'Biological characteristic type', 'Number', 'Number', 'Dimensionless', 'Int
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(232, NULL, 'Biological characteristic type', 'Number p2', 'Number in the second pass
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(231, NULL, 'Biological characteristic type', 'Number p1', 'Number in the first pass'
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(233, NULL, 'Biological characteristic type', 'Number p3', 'Number in the 3rd pass',
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(234, NULL, 'Biological characteristic type', 'Number p4', 'Number in the 4th pass',
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(235, NULL, 'Biological characteristic type', 'Number p5', 'Number in the 5th pass',
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(262, NULL, 'Biological characteristic type', 'eye_diam_vert', 'Vertical eye diameter
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(263, NULL, 'Biological characteristic type', 'eye_diam_horiz', 'Horizontal eye diamet
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(264, NULL, 'Biological characteristic type', 'length_pect', 'Length of the pectoral f
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(265, NULL, 'Biological characteristic type', 'presence_neuromast', 'Presence of neuro
INSERT INTO damdb_nomenclature.biological_characteristic_type

```

```

(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(266, NULL, 'Biological characteristic type', 'contrast', 'Contrasted colors between o
INSERT INTO damdb_nomenclature.biological_characteristic_type
(no_id, no_code, no_type, no_name, bc_label, bc_unit, bc_data_type)
VALUES(236, NULL, 'Biological characteristic type', 'Number p6', 'Number in the 6th pass', '

--
-- TOC entry 8058 (class 0 OID 2558508)
-- Dependencies: 380
-- Data for Name: control_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.control_type
(no_id, no_code, no_type, no_name)
VALUES(84, NULL, 'Control type', 'Unknown');
INSERT INTO damdb_nomenclature.control_type
(no_id, no_code, no_type, no_name)
VALUES(85, NULL, 'Control type', 'Trapping');
INSERT INTO damdb_nomenclature.control_type
(no_id, no_code, no_type, no_name)
VALUES(86, NULL, 'Control type', 'Resistivity fish counter');
INSERT INTO damdb_nomenclature.control_type
(no_id, no_code, no_type, no_name)
VALUES(87, NULL, 'Control type', 'Visual image analysis');
INSERT INTO damdb_nomenclature.control_type
(no_id, no_code, no_type, no_name)
VALUES(88, NULL, 'Control type', 'Acoustic counter');
INSERT INTO damdb_nomenclature.control_type
(no_id, no_code, no_type, no_name)
VALUES(89, NULL, 'Control type', 'Optoelectronics');

--
-- TOC entry 8059 (class 0 OID 2558511)
-- Dependencies: 381
-- Data for Name: downstream_mitigation_measure; Type: TABLE DATA; Schema: damdb_nomenclature
--

INSERT INTO damdb_nomenclature.downstream_mitigation_measure
(no_id, no_code, no_type, no_name)
VALUES(282, NULL, 'downstream mitigation measure', 'Water intake');
INSERT INTO damdb_nomenclature.downstream_mitigation_measure

```

```

(no_id, no_code, no_type, no_name)
VALUES(283, NULL, 'downstream mitigation measure', 'Fish friendly turbine');
INSERT INTO damdb_nomenclature.downstream_mitigation_measure
(no_id, no_code, no_type, no_name)
VALUES(284, NULL, 'downstream mitigation measure', 'Fish adapted Management');

--
-- TOC entry 8060 (class 0 OID 2558514)
-- Dependencies: 382
-- Data for Name: ecological_productivity; Type: TABLE DATA; Schema: damdb_nomenclature; Owner:
--

INSERT INTO damdb_nomenclature.ecological_productivity
(no_id, no_code, no_type, no_name)
VALUES(215, NULL, 'Ecological productivity', 'Unknown');
INSERT INTO damdb_nomenclature.ecological_productivity
(no_id, no_code, no_type, no_name)
VALUES(216, NULL, 'Ecological productivity', 'Good productivity');
INSERT INTO damdb_nomenclature.ecological_productivity
(no_id, no_code, no_type, no_name)
VALUES(217, NULL, 'Ecological productivity', 'Bad productivity');

--
-- TOC entry 8061 (class 0 OID 2558517)
-- Dependencies: 383
-- Data for Name: ecological_status_class; Type: TABLE DATA; Schema: damdb_nomenclature; Owner:
--

INSERT INTO damdb_nomenclature.ecological_status_class
(no_id, no_code, no_type, no_name)
VALUES(90, NULL, 'Ecological status class', 'High');
INSERT INTO damdb_nomenclature.ecological_status_class
(no_id, no_code, no_type, no_name)
VALUES(91, NULL, 'Ecological status class', 'Good');
INSERT INTO damdb_nomenclature.ecological_status_class
(no_id, no_code, no_type, no_name)
VALUES(92, NULL, 'Ecological status class', 'Moderate');
INSERT INTO damdb_nomenclature.ecological_status_class
(no_id, no_code, no_type, no_name)
VALUES(93, NULL, 'Ecological status class', 'Poor');
INSERT INTO damdb_nomenclature.ecological_status_class

```

```

(no_id, no_code, no_type, no_name)
VALUES(94, NULL, 'Ecological status class', 'Unclassified');

--
-- TOC entry 8062 (class 0 OID 2558520)
-- Dependencies: 384
-- Data for Name: effort_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.effort_type
(no_id, no_code, no_type, no_name)
VALUES(99, NULL, 'Effort type', 'Unknown');
INSERT INTO damdb_nomenclature.effort_type
(no_id, no_code, no_type, no_name)
VALUES(100, NULL, 'Effort type', 'Number of gear');
INSERT INTO damdb_nomenclature.effort_type
(no_id, no_code, no_type, no_name)
VALUES(101, NULL, 'Effort type', 'Duration days');
INSERT INTO damdb_nomenclature.effort_type
(no_id, no_code, no_type, no_name)
VALUES(102, NULL, 'Effort type', 'Area in square meters');
INSERT INTO damdb_nomenclature.effort_type
(no_id, no_code, no_type, no_name)
VALUES(103, NULL, 'Effort type', 'Volume in cubic meters');
INSERT INTO damdb_nomenclature.effort_type
(no_id, no_code, no_type, no_name)
VALUES(104, NULL, 'Effort type', 'Number gear par days');
INSERT INTO damdb_nomenclature.effort_type
(no_id, no_code, no_type, no_name)
VALUES(105, NULL, 'Effort type', 'Number gear per month');

--
-- TOC entry 8063 (class 0 OID 2558523)
-- Dependencies: 385
-- Data for Name: electrofishing_mean; Type: TABLE DATA; Schema: damdb_nomenclature; Owner:
--

INSERT INTO damdb_nomenclature.electrofishing_mean
(no_id, no_code, no_type, no_name, ef_definition)
VALUES(70, NULL, 'Electrofishing mean', 'Unknown', NULL);
INSERT INTO damdb_nomenclature.electrofishing_mean

```

```

(no_id, no_code, no_type, no_name, ef_definition)
VALUES(71, NULL, 'Electrofishing mean', 'By foot', NULL);
INSERT INTO damdb_nomenclature.electrofishing_mean
(no_id, no_code, no_type, no_name, ef_definition)
VALUES(72, NULL, 'Electrofishing mean', 'By boat', NULL);
INSERT INTO damdb_nomenclature.electrofishing_mean
(no_id, no_code, no_type, no_name, ef_definition)
VALUES(73, NULL, 'Electrofishing mean', 'Mix', NULL);

--
-- TOC entry 8064 (class 0 OID 2558528)
-- Dependencies: 386
-- Data for Name: fisher_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.fisher_type
(no_id, no_code, no_type, no_name)
VALUES(95, NULL, 'Fisher type', 'Unknown');
INSERT INTO damdb_nomenclature.fisher_type
(no_id, no_code, no_type, no_name)
VALUES(96, NULL, 'Fisher type', 'Professional fishing');
INSERT INTO damdb_nomenclature.fisher_type
(no_id, no_code, no_type, no_name)
VALUES(97, NULL, 'Fisher type', 'Amateur fishing');
INSERT INTO damdb_nomenclature.fisher_type
(no_id, no_code, no_type, no_name)
VALUES(98, NULL, 'Fisher type', 'Other');

--
-- TOC entry 8065 (class 0 OID 2558531)
-- Dependencies: 387
-- Data for Name: fishway_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(270, 'VS', 'Fishway type', 'Vertical slot fishway');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(271, 'PO', 'Fishway type', 'Pool type fishway');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)

```

```

VALUES(272, 'FL', 'Fishway type', 'Fish lock');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(273, 'D', 'Fishway type', 'Denil pass');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(275, 'RR', 'Fishway type', 'Rock ramp');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(276, 'ER', 'Fishway type', 'Eel ramp');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(277, 'LA', 'Fishway type', 'Lateral canal');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(278, 'AR', 'Fishway type', 'Artificial river');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(279, '?', 'Fishway type', 'Unknown');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(280, 'S', 'Fishway type', 'Sluice');
INSERT INTO damdb_nomenclature.fishway_type
(no_id, no_code, no_type, no_name)
VALUES(274, 'L', 'Fishway type', 'Fish lift');

--
-- TOC entry 8066 (class 0 OID 2558534)
-- Dependencies: 388
-- Data for Name: gear_characteristic_type; Type: TABLE DATA; Schema: damdb_nomenclature; O
--

INSERT INTO damdb_nomenclature.gear_characteristic_type
(no_id, no_code, no_type, no_name)
VALUES(200, NULL, 'Gear characteristic type', 'Unknown');
INSERT INTO damdb_nomenclature.gear_characteristic_type
(no_id, no_code, no_type, no_name)
VALUES(201, NULL, 'Gear characteristic type', 'Stretched mesh size by the trap net');
INSERT INTO damdb_nomenclature.gear_characteristic_type
(no_id, no_code, no_type, no_name)
VALUES(202, NULL, 'Gear characteristic type', 'Sieve area by net');
INSERT INTO damdb_nomenclature.gear_characteristic_type
(no_id, no_code, no_type, no_name)
VALUES(203, NULL, 'Gear characteristic type', 'Net length');

```





```

INSERT INTO damdb_nomenclature.habitat_loss_type (no_id, no_code, no_type, no_name) VALUES (
INSERT INTO damdb_nomenclature.habitat_loss_type (no_id, no_code, no_type, no_name) VALUES (
INSERT INTO damdb_nomenclature.habitat_loss_type (no_id, no_code, no_type, no_name) VALUES (
INSERT INTO damdb_nomenclature.habitat_loss_type (no_id, no_code, no_type, no_name) VALUES (
INSERT INTO damdb_nomenclature.habitat_loss_type (no_id, no_code, no_type, no_name) VALUES (

--
-- TOC entry 8069 (class 0 OID 2558543)
-- Dependencies: 391
-- Data for Name: individual_status; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.individual_status (no_id, no_code, no_type, no_name) VALUES (4
INSERT INTO damdb_nomenclature.individual_status (no_id, no_code, no_type, no_name) VALUES (5
INSERT INTO damdb_nomenclature.individual_status (no_id, no_code, no_type, no_name) VALUES (5
INSERT INTO damdb_nomenclature.individual_status (no_id, no_code, no_type, no_name) VALUES (5
INSERT INTO damdb_nomenclature.individual_status (no_id, no_code, no_type, no_name) VALUES (5

--
-- TOC entry 8070 (class 0 OID 2558546)
-- Dependencies: 392
-- Data for Name: migration_direction; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.migration_direction (no_id, no_code, no_type, no_name) VALUES (
INSERT INTO damdb_nomenclature.migration_direction (no_id, no_code, no_type, no_name) VALUES (
INSERT INTO damdb_nomenclature.migration_direction (no_id, no_code, no_type, no_name) VALUES (

--
-- TOC entry 8071 (class 0 OID 2558549)
-- Dependencies: 393
-- Data for Name: mortality_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.mortality_type (no_id, no_code, no_type, no_name) VALUES (184
INSERT INTO damdb_nomenclature.mortality_type (no_id, no_code, no_type, no_name) VALUES (185
INSERT INTO damdb_nomenclature.mortality_type (no_id, no_code, no_type, no_name) VALUES (186
INSERT INTO damdb_nomenclature.mortality_type (no_id, no_code, no_type, no_name) VALUES (187
INSERT INTO damdb_nomenclature.mortality_type (no_id, no_code, no_type, no_name) VALUES (188
INSERT INTO damdb_nomenclature.mortality_type (no_id, no_code, no_type, no_name) VALUES (189

```

```

INSERT INTO damdb_nomenclature.gear_type (no_id, no_code, no_type, no_name, isscfg_code, ma

--
-- TOC entry 8054 (class 0 OID 2558478)
-- Dependencies: 372
-- Data for Name: nomenclature; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

/* INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(38,
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(42, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(43, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(44, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(45, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(46, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(48, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(39, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(41, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(40, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(47, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(232, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(231,
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(233, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(234, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(235, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(262, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(263, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(264, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(265, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(266, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(236, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(30, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(54, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(55, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(56, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(57, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(58, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(59, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(84, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(85, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(86, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(87, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(88, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(89, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(282, M

```







```

INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(12, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(1, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(2, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(3, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(4, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(5, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(6, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(7, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(8, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(9, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(10, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(281, M
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(13, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(14, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(15, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(16, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(17, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(18, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(19, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(20, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(21, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(22, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(23, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(24, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(25, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(26, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(27, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(28, NU
INSERT INTO damdb_nomenclature.nomenclature (no_id, no_code, no_type, no_name) VALUES(29, NU

--
-- TOC entry 8073 (class 0 OID 2558553)
-- Dependencies: 395
-- Data for Name: observation_origin; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: p
--

INSERT INTO damdb_nomenclature.observation_origin (no_id, no_code, no_type, no_name) VALUES
INSERT INTO damdb_nomenclature.observation_origin (no_id, no_code, no_type, no_name) VALUES

--
-- TOC entry 8074 (class 0 OID 2558556)
-- Dependencies: 396

```





```

-- Data for Name: orient_flow; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.orient_flow (no_id, no_code, no_type, no_name) VALUES(238, '1', '1', '1')
INSERT INTO damdb_nomenclature.orient_flow (no_id, no_code, no_type, no_name) VALUES(239, '2', '1', '1')
INSERT INTO damdb_nomenclature.orient_flow (no_id, no_code, no_type, no_name) VALUES(240, '3', '1', '1')
INSERT INTO damdb_nomenclature.orient_flow (no_id, no_code, no_type, no_name) VALUES(241, '4', '1', '1')

--

-- TOC entry 8079 (class 0 OID 2558571)
-- Dependencies: 401
-- Data for Name: period_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(74, '1', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(75, '2', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(76, '3', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(77, '4', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(78, '5', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(79, '6', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(80, '7', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(81, '8', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(82, '9', '1', '1')
INSERT INTO damdb_nomenclature.period_type (no_id, no_code, no_type, no_name) VALUES(83, '10', '1', '1')

--

-- TOC entry 8080 (class 0 OID 2558574)
-- Dependencies: 402
-- Data for Name: predation_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.predation_type (no_id, no_code, no_type, no_name) VALUES(221, '1', '1', '1')
INSERT INTO damdb_nomenclature.predation_type (no_id, no_code, no_type, no_name) VALUES(222, '2', '1', '1')
INSERT INTO damdb_nomenclature.predation_type (no_id, no_code, no_type, no_name) VALUES(223, '3', '1', '1')

--

-- TOC entry 8081 (class 0 OID 2558577)
-- Dependencies: 403
-- Data for Name: predator_subtype; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

```

```

INSERT INTO damdb_nomenclature.predator_subtype (no_id, no_code, no_type, no_name) VALUES(20

--
-- TOC entry 8082 (class 0 OID 2558580)
-- Dependencies: 404
-- Data for Name: scientific_observation_method; Type: TABLE DATA; Schema: damdb_nomenclatur
--

INSERT INTO damdb_nomenclature.scientific_observation_method (no_id, no_code, no_type, no_na

--
-- TOC entry 8083 (class 0 OID 2558585)
-- Dependencies: 405
-- Data for Name: sex; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.sex (no_id, no_code, no_type, no_name) VALUES(180, NULL, 'Sex
INSERT INTO damdb_nomenclature.sex (no_id, no_code, no_type, no_name) VALUES(181, NULL, 'Sex
INSERT INTO damdb_nomenclature.sex (no_id, no_code, no_type, no_name) VALUES(182, NULL, 'Sex
INSERT INTO damdb_nomenclature.sex (no_id, no_code, no_type, no_name) VALUES(183, NULL, 'Sex

--
-- TOC entry 8056 (class 0 OID 2558489)
-- Dependencies: 375
-- Data for Name: species; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

```

```

INSERT INTO damdb_nomenclature.species (no_id, no_code, no_type, no_name, sp_vernacular_name) VALUES(220, NULL, 'S', 'S', 'S')
--
-- TOC entry 8084 (class 0 OID 2558588)
-- Dependencies: 406
-- Data for Name: stage; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.stage (no_id, no_code, no_type, no_name) VALUES(224, NULL, 'S', 'S')
INSERT INTO damdb_nomenclature.stage (no_id, no_code, no_type, no_name) VALUES(225, NULL, 'S', 'S')
INSERT INTO damdb_nomenclature.stage (no_id, no_code, no_type, no_name) VALUES(226, NULL, 'S', 'S')
INSERT INTO damdb_nomenclature.stage (no_id, no_code, no_type, no_name) VALUES(227, NULL, 'S', 'S')
INSERT INTO damdb_nomenclature.stage (no_id, no_code, no_type, no_name) VALUES(228, NULL, 'S', 'S')
INSERT INTO damdb_nomenclature.stage (no_id, no_code, no_type, no_name) VALUES(229, NULL, 'S', 'S')
INSERT INTO damdb_nomenclature.stage (no_id, no_code, no_type, no_name) VALUES(230, NULL, 'S', 'S')

--
-- TOC entry 8085 (class 0 OID 2558591)
-- Dependencies: 407
-- Data for Name: turbine_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(242, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(243, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(244, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(245, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(246, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(247, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(248, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(249, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(250, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(251, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(252, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(253, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(254, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(255, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(256, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(257, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(258, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(259, 'M', 'S', 'M')
INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(260, 'M', 'S', 'M')

```

```

INSERT INTO damdb_nomenclature.turbine_type (no_id, no_code, no_type, no_name) VALUES(261, N

--
-- TOC entry 8086 (class 0 OID 2558594)
-- Dependencies: 408
-- Data for Name: type_of_unit; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.type_of_unit (no_id, no_code, no_type, no_name, tu_unit) VALU
INSERT INTO damdb_nomenclature.type_of_unit (no_id, no_code, no_type, no_name, tu_unit) VALU
INSERT INTO damdb_nomenclature.type_of_unit (no_id, no_code, no_type, no_name, tu_unit) VALU
INSERT INTO damdb_nomenclature.type_of_unit (no_id, no_code, no_type, no_name, tu_unit) VALU

--
-- TOC entry 8057 (class 0 OID 2558492)
-- Dependencies: 376
-- Data for Name: value_type; Type: TABLE DATA; Schema: damdb_nomenclature; Owner: postgres
--

INSERT INTO damdb_nomenclature.value_type (no_id, no_code, no_type, no_name) VALUES(54, NULL
INSERT INTO damdb_nomenclature.value_type (no_id, no_code, no_type, no_name) VALUES(55, NULL
INSERT INTO damdb_nomenclature.value_type (no_id, no_code, no_type, no_name) VALUES(56, NULL
INSERT INTO damdb_nomenclature.value_type (no_id, no_code, no_type, no_name) VALUES(57, NULL
INSERT INTO damdb_nomenclature.value_type (no_id, no_code, no_type, no_name) VALUES(58, NULL
INSERT INTO damdb_nomenclature.value_type (no_id, no_code, no_type, no_name) VALUES(59, NULL

--
-- TOC entry 8092 (class 0 OID 0)
-- Dependencies: 394
-- Name: nomenclature_no_id_seq; Type: SEQUENCE SET; Schema: damdb_nomenclature; Owner: postgres
--
-- A CONFIRMER (dans la table damdb_nomenclature.nomenclature, le dernier ID est 306
SELECT pg_catalog.setval('damdb_nomenclature.nomenclature_no_id_seq', 307, false);

```

no_id	no_code	no_type	no_name	sp_vernacular_name
333	CIV		Species Anguilla_anguilla (glass eel)	European eel (glass eel stage)
315	127186		Species Salmo salar	Atlantic salmon

no_id	no_code	no_type	no_name	sp_vernacular_name
317	126413	Species	Alosa alosa	Twait shad
318	126415	Species	Alosa fallax	Allis shad
319	101174	Species	Petromyzon marinus	Sea lamprey
320	101172	Species	Lampretra fluviatilis	European river lamprey
30	126281	Species	Anguilla anguilla	European eel
316	127187	Species	Salmo trutta	Sea trout

no_id	no_code	no_type	no_name	spe_code
225	G	Stage	Glass eel	126281
226	Y	Stage	Yellow eel	126281
227	S	Stage	Silver eel	126281
228	GY	Stage	Glass & yellow eel mixed	126281
229	YS	Stage	Yellow & silver eel mixed	126281
230	AL	Stage	G, Y & S eel mixed	126281
426	E	stage	Egg	127187
427	EE	stage	Eyed egg	127187
428	ALV	stage	Alevin with the yolk sac	127187
429	FR	stage	Fry	127187
430	P	stage	Parr	127187
431	SM	stage	Smolt	127187
432	PS	stage	Post Smolt	127187
433	A	stage	Adult	127187
434	AL	stage	All stages	127187
435	_	stage	No life stage	127187
416	E	stage	Egg	127186
417	EE	stage	Eyed egg	127186
418	ALV	stage	Alevin with the yolk sac	127186
419	FR	stage	Fry	127186
420	P	stage	Parr	127186

no_id	no_code	no_type	no_name	spe_code
421	SM	stage	Smolt	127186
422	PS	stage	Post Smolt	127186
423	A	stage	Adult	127186
424	AL	stage	All stages	127186
425	_	stage	No life stage	127186

no_id	no_code	no_type	no_name	bc_label
38		Biological characteristic type	Unknown	Unknown
44		Biological characteristic type	Sex	Sex
45		Biological characteristic type	Stage	stage
48		Biological characteristic type	Density	Density
41		Biological characteristic type	Upper length	Total length - upper bound
40		Biological characteristic type	Lower length	Total length - lower bound
47		Biological characteristic type	Number	Number
232		Biological characteristic type	Number p2	Number in the second pass
231		Biological characteristic type	Number p1	Number in the first pass
233		Biological characteristic type	Number p3	Number in the 3rd pass
234		Biological characteristic type	Number p4	Number in the 4th pass
235		Biological characteristic type	Number p5	Number in the 5th pass
236		Biological characteristic type	Number p6	Number in the 6th pass
265		Biological characteristic type	Presence_neuromast	Presence of neuromasts along
266		Biological characteristic type	Contrast	Contrasted colors between dor
307		Biological characteristic type	Eye_diam_mean_mm	Mean eye diameter
42		Biological characteristic type	Weightg	Weight in g
43		Biological characteristic type	Ageyear	Age in year
262		Biological characteristic type	Eye_diam_vert_mm	Vertical eye diameter
263		Biological characteristic type	Eye_diam_horiz_mm	Horizontal eye diameter
264		Biological characteristic type	Pectoral_lengthmm	Length of the pectoral fin
46		Biological characteristic type	Rate	rate

no_id	no_code	no_type	no_name	bc_label
39		Biological characteristic type	Lengthmm	Total length

no_id	no_code	no_type	no_name
282		downstream mitigation measure	Water intake
283		downstream mitigation measure	Fish friendly turbine
284		downstream mitigation measure	Fish adapted Management

no_id	no_code	no_type	no_name
270	VS	Fishway type	Vertical slot fishway
271	PO	Fishway type	Pool type fishway
272	FL	Fishway type	Fish lock
273	D	Fishway type	Denil pass
275	RR	Fishway type	Rock ramp
276	ER	Fishway type	Eel ramp
277	LA	Fishway type	Lateral canal
278	AR	Fishway type	Artificial river
279	?	Fishway type	Unknown
280	S	Fishway type	Sluice
274	L	Fishway type	Fish lift

no_id	no_code	no_type	no_name	isscfg_code	main_g
106		Gear type	Unknown	01.0.0	SURRO
107		Gear type	Surrounding Nets with Purse Lines (Purse Seines)	01.1.0	SURRO
108		Gear type	Surrounding Nets with Purse Lines - one boat	01.1.1	SURRO
109		Gear type	Surrounding Nets with Purse Lines - two boats	01.1.2	SURRO
110		Gear type	Surrounding Nets without Purse Lines	01.2.0	SURRO
111		Gear type	Unknown	02.0.0	SEINE
112		Gear type	Beach Seines	02.1.0	SEINE

no_id	no_code	no_type	no_name	isscfg_code	main_g
113		Gear type	Boat Seines	02.2.0	SEINE F
114		Gear type	Boat Seines	02.2.1	SEINE F
115		Gear type	Boat Seines	02.2.2	SEINE F
116		Gear type	Boat Seines	02.2.3	SEINE F
117		Gear type	Boat Seines	02.9.0	SEINE F
118		Gear type	Unknown	03.0.0	TRAWL
119		Gear type	Unknown	03.1.0	TRAWL
120		Gear type	Beam trawls	03.1.1	TRAWL
121		Gear type	Bottom otter trawls	03.1.2	TRAWL
122		Gear type	Bottom pair trawls	03.1.3	TRAWL
123		Gear type	Unknown	03.1.4	TRAWL
124		Gear type	Unknown	03.1.5	TRAWL
125		Gear type	Unknown	03.1.9	TRAWL
126		Gear type	Unknown	03.2.0	TRAWL
127		Gear type	Midwater otter trawls	03.2.1	TRAWL
128		Gear type	Midwater pair trawls	03.2.2	TRAWL
129		Gear type	Unknown	03.2.3	TRAWL
130		Gear type	Unknown	03.2.9	TRAWL
131		Gear type	Otter Twin Trawls	03.3.0	TRAWL
132		Gear type	Otter Twin Trawls	03.4.9	TRAWL
133		Gear type	Otter Twin Trawls	03.5.9	TRAWL
134		Gear type	Otter Twin Trawls	03.9.0	TRAWL
135		Gear type	Boat Dredges	04.0.0	DREDG
136		Gear type	Boat Dredges	04.1.0	DREDG
137		Gear type	Hand Dredges	04.2.0	DREDG
138		Gear type	Portable Lift Nets	05.0.0	LIFT NE
139		Gear type	Portable Lift Nets	05.1.0	LIFT NE
140		Gear type	Boat Operated Lift Nets	05.2.0	LIFT NE
141		Gear type	Shore Operated Lift Nets	05.3.0	LIFT NE

no_id	no_code	no_type	no_name	isscfg_code	main_g
142		Gear type	Shore Operated Lift Nets	05.9.0	LIFT NE
143		Gear type	Cast Nets	06.0.0	FALLIN
144		Gear type	Cast Nets	06.1.0	FALLIN
145		Gear type	Falling Gear (Not Specified)	06.9.0	FALLIN
146		Gear type	Set Gillnets (Anchored)	07.0.0	GILLNE
147		Gear type	Set Gillnets (Anchored)	07.1.0	GILLNE
148		Gear type	Drifting Gillnets (Driftnets)	07.2.0	GILLNE
149		Gear type	Encircling gillnets	07.3.0	GILLNE
150		Gear type	Fixed Gillnets (on Stakes)	07.4.0	GILLNE
151		Gear type	Trammel Nets	07.5.0	GILLNE
152		Gear type	Trammel Nets	07.6.0	GILLNE
153		Gear type	Trammel Nets	07.9.0	GILLNE
154		Gear type	Trammel Nets	07.9.1	GILLNE
155		Gear type	Unknown	08.0.0	TRAPS
156		Gear type	Stationary Uncovered Pound Nets	08.1.0	TRAPS
157		Gear type	Pots	08.2.0	TRAPS
158		Gear type	Fyke Nets	08.3.0	TRAPS
159		Gear type	Stow Nets	08.4.0	TRAPS
160		Gear type	Barriers, Fences, Weirs, Corrals, etc.	08.5.0	TRAPS
161		Gear type	Aerial Traps	08.6.0	TRAPS
162		Gear type	Aerial Traps	08.9.0	TRAPS
163		Gear type	Handlines and Pole-Lines (Hand Operated)	09.0.0	HOOKS
164		Gear type	Handlines and Pole-Lines (Hand Operated)	09.1.0	HOOKS
165		Gear type	Handlines and Pole-Lines (Mechanized)	09.2.0	HOOKS
166		Gear type	Set Longlines	09.3.0	HOOKS
167		Gear type	Drifting Longlines	09.4.0	HOOKS
168		Gear type	Longlines (Not Specified)	09.5.0	HOOKS
169		Gear type	Trolling Lines	09.6.0	HOOKS
170		Gear type	Trolling Lines	09.9.0	HOOKS

no_id	no_code	no_type	no_name	isscfg_code	main_g
171		Gear type	Unknown	10.0.0	GRAPP
172		Gear type	Harpoons	10.1.0	GRAPP
173		Gear type	Pumps	11.0.0	HARVE
174		Gear type	Pumps	11.1.0	HARVE
175		Gear type	Mechanized Dredges	11.2.0	HARVE
176		Gear type	Pumps	11.9.0	HARVE
177		Gear type	Miscellaneous	20.0.0	MISCEL
178		Gear type	Recreational fishing gear	25.0.0	RECRE
179		Gear type	Unknown	99.0.0	UNKNO

no_id	no_code	no_type	no_name	ef_definition
70		electrofishing_mean	Unknown	
71		electrofishing_mean	By foot	
72		electrofishing_mean	By boat	
73		electrofishing_mean	Mix	

no\_id no\_code no\_type no\_name

no_id	no_code	no_type	no_name
218	UN	Obstruction_type	Unknown
291	DA	Obstruction_type	Dam
292	WE	Obstruction_type	Weir
293	RR	Obstruction_type	Rock ramp
294	CU	Obstruction_type	Culvert
295	FO	Obstruction_type	Ford
296	BR	Obstruction_type	Bridge
297	OT	Obstruction_type	Other
298	DI	Obstruction_type	Dike

no_id	no_code	no_type	no_name
299	GR	Obstruction_type	Grid
300	PP	Obstruction_type	Penstock Pipe
314	WA	Obstruction type	Waterfall, Natural obstruction

no_id	no_code	no_type	no_name
238	1	orient_flow	[70-90°]
239	2	orient_flow	[50-70°]
240	3	orient_flow	[30-50°]
241	4	orient_flow	<30°

no_id	no_code	no_type	no_name	sc_observation_category	sc_de
68		scientific_observation_method	Unknown	Gear fishing	
61	UN	scientific_observation_method	Unknown	Electro-fishing	Electr
63	P1	scientific_observation_method	Partial1bank	Electro-fishing	Electr
64	P2	scientific_observation_method	Partial2banks	Electro-fishing	Electr
65	PR	scientific_observation_method	Partialrandom	Electro-fishing	Electr
66	PR	scientific_observation_method	Partialprop	Electro-fishing	Electr
67	OT	scientific_observation_method	Other	Electro-fishing	Electr
301	DH	scientific_observation_method	Deep habitat	Electro-fishing	Norma
303	PE	scientific_observation_method	Point sampling eel	Electro-fishing	Electr
305	BB	scientific_observation_method	Boom boat	Electro-fishing	Boom
304	TE	scientific_observation_method	TEF	Electro-fishing	Timed
306	EP	scientific_observation_method	EPA	Electro-fishing	Partia
62	ST	scientific_observation_method	Standard by foot	Electro-fishing	Electr
302	EE	scientific_observation_method	Standard eel	Electro-fishing	Electr

no_id	no_code	no_type	no_name
182	F	sex	Female

no_id	no_code	no_type	no_name
181	M	sex	Male
360	H	sex	Hermaphrodite
361	I	sex	Immature - attempt made but sex could not be destinguished
362	T	sex	Transitional
363	U	sex	Undetermined - no attempt made
364	X	sex	Mixed

no_id	no_code	no_type	no_name
242		Turbine type	Horizontal axis Kaplan (bulb)
243		Turbine type	Other (please specify)
244		Turbine type	Double Francis (spiral case)
245		Turbine type	Francis unspecified
246		Turbine type	Turbine with fixed blade propeller and vertical axis
247		Turbine type	Kaplan not specified
248		Turbine type	Pelton
249		Turbine type	Reversible
250		Turbine type	Kaplan (model of S-turbine)
251		Turbine type	Turbine with fixed blade propeller and horizontal axis
252		Turbine type	Unknown
253		Turbine type	Vertical axis Kaplan
254		Turbine type	Francis without volute
255		Turbine type	Francis (spiral case)
256		Turbine type	Banki-Michell (cross-flow)
257		Turbine type	VLH
258		Turbine type	Archimedean screw turbine
259		Turbine type	Water wheel turbine (aqualienne)
260		Turbine type	Water wheel turbine (others)
261		Turbine type	Propeller unspecified

no_id	no_code	no_type	no_name
307	ST	Event change	Start (Default)
308	CO	Event change	Construction
309	SU	Event change	Start usage
310	ER	Event change	Erasurement
311	EU	Event change	End usage
312	HC	Event change	Height change
313	PE	Event change	Partial erasurement
332	FW	Event change	Fishway built or changed

no_id	no_code	no_type	no_name
1	0FW	age	0 year in freshwater
2	1FW	age	1 year in freshwater
3	2FW	age	2 years in freshwater
4	3FW	age	3 years in freshwater
5	4FW	age	4 years in freshwater
6	5FW	age	5 years in freshwater
7	6FW	age	6 years in freshwater
8	1SW	age	1 year in seawater
9	2SW	age	2 years in seawater
10	MSW	age	Two years or more in seawater
11	1+	age	Older than one year in freshwater
335	0FW	age	0 year in freshwater
336	1FW	age	1 year in freshwater
337	2FW	age	2 years in freshwater
338	3FW	age	3 years in freshwater
339	4FW	age	4 years in freshwater
340	5FW	age	5 years in freshwater
341	6FW	age	6 years in freshwater
342	1SW	age	1 year in seawater

no_id	no_code	no_type	no_name
343	2SW	age	2 years in seawater
344	MSW	age	Two years or more in seawater
345	1+	age	Older than one year in freshwater

no_id	no_code	no_type	no_name
347	A	maturity	Immature
348	B	maturity	Developing
349	Ba	maturity	Developing, but functionally immature (first-time developer)
350	Bb	maturity	Developing and functionally mature
351	C	maturity	Spawning
352	Ca	maturity	Actively spawning
353	Cb	maturity	Spawning capable
354	D	maturity	Regressing / Regenerating
355	Da	maturity	Regressing
356	Db	maturity	Regenerating
357	E	maturity	Omitted spawning
358	F	maturity	Abnormal

no_id	no_code	no_type	no_name
99		Effort type	Unknown
103		Effort type	Volume in cubic meters
101	nrd	Effort type	Number of day (nr day)
104	nd	Effort type	Net-days (nd)
100	gd	Effort type	Gear-days (Fyke net, traps)
102	m2	Effort type	Area in square meters

## Chapter 9

# Annex 3 : Dam tables

The dam db was created from the dbeel using the following script.

SQL code to transfered from the DBEEL

```
-- see migdb exchange_with_server for initial import.

-- we start with a structure collected from SLIME and SUDOANG with empty tables
-- We will populate those tables and modify them

-- we will separate the dam and the electrofishing db even
--if they use the same nomenclature as the source.
ALTER TABLE nomenclature.nomenclature
    ALTER COLUMN no_code TYPE TEXT;

ALTER SCHEMA dbmig RENAME TO electrofishing;
--ALTER SCHEMA dbmig RENAME TO dam; (with manual dump)
DROP TABLE dam.electrofishing;
DROP TABLE dam.fishery;
DROP TABLE dam.gear_fishing;
DROP TABLE dam.predation;
DROP TABLE dam.differentiation;
DROP TABLE dam.biological_characteristic;
DROP TABLE dam.batch;

DROP TABLE electrofishing.bypass;
DROP TABLE electrofishing.hpp;
DROP TABLE electrofishing.physical_obstruction;
DROP TABLE electrofishing.obstruction;
```

```

DROP TABLE electrofishing.turbine;

GRANT ALL PRIVILEGES ON SCHEMA dam TO diaspara_admin;
GRANT ALL PRIVILEGES ON SCHEMA electrofishing TO diaspara_admin;
GRANT USAGE ON SCHEMA dam TO diaspara_read;
GRANT USAGE ON SCHEMA electrofishing TO diaspara_read;

-- Note the foreign data wrappers are also in WP3_mig_db/0_foreign_data_wrapper.sql

CREATE SERVER eda_data_wrapper
  FOREIGN DATA WRAPPER postgres_fdw
  OPTIONS (host 'localhost', port '5432', dbname 'eda2.3');

CREATE USER MAPPING FOR USER
  SERVER wgeel_data_wrapper
  OPTIONS (user 'postgres', password 'postgres', updatable 'false');

CREATE SERVER wgnas_data_wrapper
  FOREIGN DATA WRAPPER postgres_fdw
  OPTIONS (host 'localhost', port '5432', dbname 'salmoglob');

CREATE USER MAPPING FOR USER
  SERVER wgeel_data_wrapper
  OPTIONS (user 'postgres', password 'postgres', updatable 'false');

CREATE USER MAPPING FOR USER
  SERVER wgnas_data_wrapper
  OPTIONS (user 'postgres', password 'postgres');

CREATE SCHEMA montepomi;

IMPORT FOREIGN SCHEMA montepomi
  FROM SERVER eda_data_wrapper
  INTO montepomi;

-- TODO put back these functions when done
DROP FUNCTION nomenclature.nomenclature_id_update() CASCADE;
DROP FUNCTION nomenclature.nomenclature_id_insert() CASCADE;

```

```

ALTER SEQUENCE nomenclature.nomenclature_no_id_seq
  START 1
  RESTART 307;

-- biological_characteristic_type => Align with other vocab for name and code

INSERT INTO nomenclature.biological_characteristic_type
SELECT * FROM nomenclature_eda.biological_characteristic_type; --22

INSERT INTO nomenclature.biological_characteristic_type
(no_id,no_type,no_name,bc_label,bc_unit,bc_data_type)
VALUES (307,'Biological characteristic type',
'eye_diam_mean_mm','Mean eye diameter','mm','real');
UPDATE nomenclature.biological_characteristic_type
SET no_name='Weightg',bc_label='Weight in g'
WHERE no_id=42;
UPDATE nomenclature.biological_characteristic_type
SET no_name='Lengthmm'
WHERE no_id=39;
UPDATE nomenclature.biological_characteristic_type
SET no_name='Ageyear',bc_label='Age in year'
WHERE no_id=43;
UPDATE nomenclature.biological_characteristic_type
SET no_name='Eye_diam_vert_mm'
WHERE no_id=262;
UPDATE nomenclature.biological_characteristic_type
SET no_name='Eye_diam_horiz_mm'
WHERE no_id=263;
UPDATE nomenclature.biological_characteristic_type
SET no_name='Pectoral_lengthmm'
WHERE no_id=264;
UPDATE nomenclature.biological_characteristic_type
SET bc_unit='Dimensionless'
WHERE no_id=46; -- percentage was there an it was wrong

--control_type REMOVE
DROP TABLE IF EXISTS nomenclature.control_type;
--downstream_mitigation_measure OK
--ecological_productivity REMOVE
DROP TABLE IF EXISTS nomenclature.ecological_productivity;
--ecological_status_clas REMOVE
DROP TABLE IF EXISTS nomenclature.ecological_status_class;

```

```

--effort_type REMOVE
DROP TABLE IF EXISTS nomenclature.effort_type;

--fisher_type REMOVE
DROP TABLE IF EXISTS nomenclature.fisher_type;
--fishway_type TODO
--Adapt with descriptions and definitions
--gear_characteristic_type REMOVE
DROP TABLE IF EXISTS nomenclature.gear_characteristic_type;
--gear_type OK
--habitat_loss_type REMOVE
DROP TABLE IF EXISTS nomenclature.habitat_loss_type;
--individual_status REMOVE
DROP TABLE IF EXISTS nomenclature.individual_status;
--migration_direction TODO
--Add to bypass
--Add to fishway
--mortality_type REMOVE
DROP TABLE IF EXISTS nomenclature.mortality_type;
--nomenclature OK
--observation_origin REMOVE
DROP TABLE IF EXISTS nomenclature.observation_origin;
--observation_place_type REMOVE
DROP TABLE IF EXISTS nomenclature.observation_place_type;
--observation_type REMOVE
DROP TABLE IF EXISTS nomenclature.observation_type;
--keep obstruction for dam
--remove rest
--obstruction_impact REMOVE
DROP TABLE IF EXISTS nomenclature.obstruction_impact;
--obstruction_type TODO
--remove chemical obstruction
--obstruction_place TODO
--remove op_gis_systemname
--remove op_gis_layername
--remove op_gis_location
--remove op_no_observationplacetype
DROP TABLE IF EXISTS dam.obstruction_place;
CREATE TABLE dam.obstruction_place (
    op_id uuid DEFAULT uuid_generate_v4() NOT NULL,
    op_placename TEXT NULL,
    op_op_id uuid NULL,
    op_dp_id int4 NULL,
    op_id_original TEXT NOT NULL,
    op_country varchar(2) NULL,

```

```

        geom geometry NULL,
        CONSTRAINT uk_op_id_original UNIQUE (op_id_original),
        CONSTRAINT obstruction_place_pkey PRIMARY KEY (op_id),
        CONSTRAINT fk_op_op_id FOREIGN KEY (op_op_id)
        REFERENCES dam.obstruction_place(op_id) ON DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT fk_op_dp_id FOREIGN KEY (op_dp_id)
        REFERENCES dam.data_provider(dp_id) ON DELETE CASCADE ON UPDATE CASCADE
    );
CREATE INDEX obstruction_place_geom_gist ON dam.obstruction_place USING gist (geom);
ALTER TABLE dam.obstruction_place OWNER TO diaspara_admin;
GRANT SELECT ON dam.obstruction_place TO diaspara_read;

--physical_obstruction TODO
--remove ob_no_origin
--remove ob_no_type
--remove ob_no_period
--remove ot_obstruction_number
--remove ot_no_mortality_type
--remove ot_no_mortality
--remove po_no_obstruction_passability (because it's a score)
--remove po_presence_eel_pass
--remove po_method_perm_ev

-- dbmig.observations definition
CREATE OR REPLACE FUNCTION nomenclature.nomenclature_id_insert()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
BEGIN
    PERFORM no_id FROM "nomenclature".nomenclature WHERE no_id = NEW.no_id;
    IF FOUND THEN
        RAISE EXCEPTION '1- Invalid no_id (%)', NEW.no_id;
    ELSE
        RETURN NEW;
    END IF;
END
$function$
;

CREATE OR REPLACE FUNCTION nomenclature.nomenclature_id_update()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
BEGIN

```

```

PERFORM no_id FROM "nomenclature".nomenclature WHERE no_id = NEW.no_id;
IF FOUND THEN
    RETURN NEW;
ELSE
    RAISE EXCEPTION '1- Invalid no_id (%)', NEW.no_id;
END IF;
END
$function$
;
-- Drop table

-- DROP TABLE dbmig.observations CASCADE;
INSERT INTO nomenclature.fishway_type SELECT * FROM nomenclature_eda.fishway_type;
CREATE TRIGGER tr_fishway_type_insert BEFORE
INSERT
    ON
    nomenclature.fishway_type FOR EACH ROW EXECUTE
FUNCTION nomenclature.nomenclature_id_insert();
CREATE TRIGGER tr_fishway_type_update BEFORE
UPDATE
    ON
    nomenclature.fishway_type FOR EACH ROW EXECUTE
FUNCTION nomenclature.nomenclature_id_update();

ALTER TABLE nomenclature.fishway_type OWNER TO diaspara_admin;
GRANT SELECT ON nomenclature.fishway_type TO diaspara_read;

INSERT INTO nomenclature.downstream_mitigation_measure
SELECT * FROM nomenclature_eda.downstream_mitigation_measure;
CREATE TRIGGER tr_downstream_mitigation_measure_insert
BEFORE INSERT
    ON
    nomenclature.downstream_mitigation_measure
FOR EACH ROW EXECUTE FUNCTION nomenclature.nomenclature_id_insert();
CREATE TRIGGER tr_downstream_mitigation_measure_update
BEFORE UPDATE
    ON
    nomenclature.downstream_mitigation_measure FOR EACH ROW
EXECUTE FUNCTION nomenclature.nomenclature_id_update();

ALTER TABLE nomenclature.downstream_mitigation_measure OWNER TO diaspara_admin;
GRANT SELECT ON nomenclature.downstream_mitigation_measure TO diaspara_read;

INSERT INTO nomenclature.obstruction_type SELECT * FROM nomenclature_eda.obstruction_type;
INSERT INTO nomenclature.obstruction_type (no_code, no_type, no_name)

```

```

VALUES ('WA',
        'Obstruction type',
        'Waterfall, Natural obstruction'
       );

-- Remove physical and chemical obstruction
DELETE FROM nomenclature.obstruction_type
        WHERE no_id=220;
DELETE FROM nomenclature.obstruction_type
        WHERE no_id=219;

CREATE TRIGGER tr_obstruction_type_insert BEFORE
INSERT
        ON
        nomenclature.obstruction_type FOR EACH ROW
EXECUTE FUNCTION nomenclature.nomenclature_id_insert();
CREATE TRIGGER tr_obstruction_type_update BEFORE
UPDATE
        ON
        nomenclature.obstruction_type FOR EACH ROW
EXECUTE FUNCTION nomenclature.nomenclature_id_update();

ALTER TABLE nomenclature.obstruction_type OWNER TO diaspara_admin;
GRANT SELECT ON nomenclature.obstruction_type TO diaspara_read;

CREATE TABLE nomenclature.event_change (
        CONSTRAINT event_change_id PRIMARY KEY (no_id)
)
INHERITS (nomenclature.nomenclature);

INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
        'ST',
        'Event change',
        'Start (Default)');
INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
        'CO',
        'Event change',
        'Construction');
INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
        'SU',
        'Event change',
        'Start usage');

```

```

INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
    'ER',
    'Event change',
    'Erasurement');
INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
    'EU',
    'Event change',
    'End usage');
INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
    'HC',
    'Event change',
    'Height change');
INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
    'PE',
    'Event change',
    'Partial erasurement');
INSERT INTO nomenclature.event_change (no_code, no_type, no_name)
VALUES (
    'FW',
    'Event change',
    'Fishway built or changed');
-- Table Triggers

CREATE TRIGGER tr_event_change_insert BEFORE
INSERT
ON
    nomenclature.event_change FOR EACH ROW
EXECUTE FUNCTION nomenclature.nomenclature_id_insert();
CREATE TRIGGER tr_event_change_update BEFORE
UPDATE
ON
    nomenclature.event_change FOR EACH ROW
EXECUTE FUNCTION nomenclature.nomenclature_id_update();

ALTER TABLE nomenclature.event_change OWNER TO diaspara_admin;
GRANT SELECT ON nomenclature.event_change TO diaspara_read;

DROP TABLE dam.establishment CASCADE;

```

```

-- data provider can reference EDMO if exists otherwise not
DROP TABLE IF EXISTS dam.data_provider CASCADE;
CREATE TABLE dam.data_provider (
  dp_id serial4 NOT NULL,
  dp_name varchar(60) NULL,
  dp_edmo_key int4 NULL,
  CONSTRAINT fk_dp_edmo_key FOREIGN KEY (dp_edmo_key)
  REFERENCES REF."EDMO"("Key") ON UPDATE CASCADE ON DELETE RESTRICT,
  dp_establishment_name TEXT,
  CONSTRAINT data_provider_pkey PRIMARY KEY (dp_id)
);

CREATE TABLE dam.obstruction (
  ob_id uuid DEFAULT uuid_generate_v4() NOT NULL,
  ob_starting_date date NULL,
  ob_ending_date date NULL,
  ob_op_id uuid NOT NULL,
  ob_obstruction_type_no_id int4 NULL,
  ob_fishway_type_no_id int4 NULL,
  ob_mitigation_measure_no_id int4 NULL,
  ob_height float4 NULL,
  ob_height_date date NULL,
  ob_downs_water_depth numeric NULL,
  ob_date_last_update date NULL,
  ob_event_change_no_id int4 NULL,
  CONSTRAINT obstruction_pkey PRIMARY KEY (ob_id),
  CONSTRAINT fk_ob_fishway_type_no_id FOREIGN KEY (ob_fishway_type_no_id)
  REFERENCES nomenclature.fishway_type(no_id) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT fk_ob_mitigation_measure_no_id FOREIGN KEY (ob_mitigation_measure_no_id)
  REFERENCES nomenclature.downstream_mitigation_measure(no_id) ON DELETE RESTRICT ON U
  CONSTRAINT fk_ob_op_id FOREIGN KEY (ob_op_id)
  REFERENCES dam.obstruction_place(op_id) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT fk_ob_obstruction_type_no_id FOREIGN KEY (ob_obstruction_type_no_id)
  REFERENCES nomenclature.obstruction_type(no_id) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT fk_event_change_no_id FOREIGN KEY (ob_event_change_no_id)
  REFERENCES nomenclature.event_change(no_id) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT uk_obstruction UNIQUE (ob_op_id, ob_starting_date)
);

-- Table Triggers

```

```

CREATE TRIGGER tr_obstruction_insert BEFORE
INSERT
  ON
    dam.obstruction FOR EACH ROW EXECUTE FUNCTION dam.observations_id_insert();
CREATE TRIGGER tr_obstruction_update BEFORE
UPDATE
  ON
    dam.obstruction FOR EACH ROW EXECUTE FUNCTION dam.observations_id_update();

ALTER TABLE dam.obstruction OWNER TO diaspara_admin;
GRANT SELECT ON dam.obstruction TO diaspara_read;

--decide what to do with po_date_presence_eel_pass
--move ob_dp_id to obstruction_place
--add natural obstruction to ot_no_obstruction_type

--species
--Use tr_species_spe

INSERT INTO nomenclature.species
  SELECT * FROM nomenclature_eda.species;

INSERT INTO nomenclature.species (no_code,no_type,no_name,sp_vernacular_name)
  VALUES ('127186','Species','Salmo salar','Atlantic salmon');
INSERT INTO nomenclature.species (no_code,no_type,no_name,sp_vernacular_name)
  VALUES ('127187','Species','Salmo trutta','Sea trout');
INSERT INTO nomenclature.species (no_code,no_type,no_name,sp_vernacular_name)
  VALUES ('126413','Species','Alosa alosa','Twait shad');
INSERT INTO nomenclature.species (no_code,no_type,no_name,sp_vernacular_name)
  VALUES ('126415','Species','Alosa fallax','Allis shad');
INSERT INTO nomenclature.species (no_code,no_type,no_name,sp_vernacular_name)
  VALUES ('101174','Species','Petromyzon marinus','Sea lamprey');
INSERT INTO nomenclature.species (no_code,no_type,no_name,sp_vernacular_name)
  VALUES ('101172','Species','Lampretra fluviatilis','European river lamprey');
UPDATE nomenclature.species
  SET sp_vernacular_name='European eel',
    no_code='126281'
  WHERE no_id=30;
INSERT INTO nomenclature.species (no_code,no_type,no_name,sp_vernacular_name)
  VALUES ('CIV','Species','Anguilla anguilla (glass eel)','European eel (glass eel stage)')

UPDATE nomenclature.species
  SET no_code='127186'
  WHERE no_id=315;
UPDATE nomenclature.species

```

```

SET no_code='126413'
WHERE no_id=317;
UPDATE nomenclature.species
SET no_code='126415'
WHERE no_id=318;
UPDATE nomenclature.species
SET no_code='101174'
WHERE no_id=319;
UPDATE nomenclature.species
SET no_code='101172'
WHERE no_id=320;
UPDATE nomenclature.species
SET no_code='126281'
WHERE no_id=30;
UPDATE nomenclature.species
SET no_code='127187'
WHERE no_id=316;

ALTER TABLE nomenclature.species OWNER TO diaspara_admin;
GRANT SELECT ON nomenclature.species TO diaspara_read;

--id
--expertise on whether species can cross the dam
--when was it built
--DELETE FROM nomenclature.fishway_type;

INSERT INTO nomenclature.fishway_type (no_id,no_code,no_type,no_name)
SELECT no_id,no_code,no_type,no_name FROM nomenclature_eda.fishway_type;
ALTER TABLE nomenclature.fishway_type OWNER TO diaspara_admin;
GRANT SELECT ON nomenclature.fishway_type TO diaspara_read;

-- DROP TABLE montepomi.dbeel_physical_obstruction_pass_species;

CREATE TABLE dam.fishway (
    fi_ob_id uuid NULL,
    fi_op_id uuid NULL,
    fi_id_original varchar(10) NULL,
    fi_species_id int4 NULL,
    fi_fishway_id int4 NULL,
    fi_presence_pass bool NULL,
    fi_date date NULL,
    CONSTRAINT fishway_pkey PRIMARY KEY (fi_ob_id, fi_species_id),
    CONSTRAINT f_ob_id FOREIGN KEY (fi_ob_id)

```

```

REFERENCES dam.obstruction (ob_id) ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT f_op_id FOREIGN KEY (fi_op_id)
REFERENCES dam.obstruction_place (op_id) ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT fk_fi_species_id FOREIGN KEY (fi_species_id)
REFERENCES nomenclature.species(no_id) ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_fi_fishway_id FOREIGN KEY (fi_fishway_id)
REFERENCES nomenclature.fishway_type(no_id) ON UPDATE CASCADE ON DELETE RESTRICT
);

ALTER TABLE dam.fishway OWNER TO diaspara_admin;
GRANT SELECT ON dam.fishway TO diaspara_read;

--physical_obstruction_score_species OK
--orient_flow OK
--period_type REMOVE
DROP TABLE IF EXISTS nomenclature.period_type;
--predation_type REMOVE
DROP TABLE IF EXISTS nomenclature.predation_type;
--predator_subtype REMOVE
DROP TABLE IF EXISTS nomenclature.predator_subtype;

--sex OK

--turbine_type OK
--type_of_unit REMOVE
DROP TABLE IF EXISTS nomenclature.type_of_unit;
--value_type REMOVE
DROP TABLE IF EXISTS nomenclature.value_type;

INSERT INTO nomenclature.orient_flow
SELECT * FROM nomenclature_eda.orient_flow;--4

--dbeel_hpp TODO
--remove_hpp_main_grid_or_production
--see if we keep_hpp_presence_of_bar_rack or go for more specific
--add_turbine_type
DROP TABLE IF EXISTS dam.hpp CASCADE;
CREATE TABLE dam.hpp (
   _hpp_id uuid NOT NULL,
   _hpp_ob_id uuid NULL,
   _hpp_name text NULL,
   _hpp_presence_bypass bool NULL,
   _hpp_total_flow_bypass numeric NULL,
   _hpp_orient_flow_no_id int4 NULL,

```

```

hpp_presence_of_bar_rack bool NULL,
hpp_bar_rack_space numeric NULL,
hpp_surface_bar_rack numeric NULL,
hpp_inclination_bar_rack numeric NULL,
hpp_presence_bypass_trashrack bool NULL,
hpp_nb_trashrack_bypass int4 NULL,
hpp_turb_max_flow numeric NULL,
hpp_reserved_flow numeric NULL,
hpp_flow_trashrack_bypass numeric NULL,
hpp_max_power numeric NULL,
hpp_nb_turbines int4 NULL,
hpp_turbine_type int4 NULL,
hpp_orientation_bar_rack numeric NULL,
hpp_id_original text NULL,
hpp_source text NULL,
CONSTRAINT hpp_pkey PRIMARY KEY (hpp_id),
CONSTRAINT fk_hpp_ob_id FOREIGN KEY (hpp_ob_id)
REFERENCES dam.obstruction(ob_id) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_hpp_orient_flow_no_id FOREIGN KEY (hpp_orient_flow_no_id)
REFERENCES nomenclature.orient_flow(no_id) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT fk_hpp_turbine_type FOREIGN KEY (hpp_turbine_type)
REFERENCES nomenclature.turbine_type(no_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

ALTER TABLE dam.hpp OWNER TO diaspara_admin;
GRANT SELECT ON dam.hpp TO diaspara_read;

-- DROP TABLE dam.bypass;

CREATE TABLE dam.bypass (
  bypass_id uuid NOT NULL,
  bypass_hpp_id uuid NULL,
  bypass_water_depth numeric NULL,
  bypass_width numeric NULL,
  bypass_is_flowng bool NULL,
  CONSTRAINT bypass_pkey PRIMARY KEY (bypass_id)
  CONSTRAINT fk_bypass_hpp_id FOREIGN KEY (bypass_hpp_id)
  REFERENCES dam.hpp ON DELETE CASCADE ON UPDATE CASCADE
);

ALTER TABLE dam.bypass OWNER TO diaspara_admin;
GRANT SELECT ON dam.bypass TO diaspara_read;

--dbeel_turbine OK

```

```

-- DROP TABLE IF EXISTS dam.turbine;

CREATE TABLE dam.turbine (
    turb_id uuid NOT NULL,
    turb_hpp_id uuid NULL,
    turb_turbine_type_no_id int4 NULL,
    turb_in_service bool NULL,
    turb_max_power numeric NULL,
    turb_min_working_flow numeric NULL,
    turb_hpp_height numeric NULL,
    turb_diameter numeric NULL,
    turb_rotation_speed numeric NULL,
    turb_nb_blades int4 NULL,
    turb_max_turbine_flow numeric NULL,
    turb_description text NULL,
    CONSTRAINT turbine_pkey PRIMARY KEY (turb_id),
    CONSTRAINT fk_turb_turbine_type_no_id FOREIGN KEY (turb_turbine_type_no_id)
    REFERENCES nomenclature.turbine_type(no_id) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT fk_turb_hpp_id FOREIGN KEY (turb_hpp_id) REFERENCES dam.hpp(hpp_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO nomenclature.turbine_type SELECT * FROM nomenclature_eda.turbine_type; --20

```

The postgresSQL code for creating the table is as following

SQL code to create dam structure

```

--
-- PostgreSQL database dump
--

-- Dumped from database version 14.20 (Ubuntu 14.20-0ubuntu0.22.04.1)
-- Dumped by pg_dump version 17.5

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET transaction_timeout = 0;
SET client_encoding = 'UTF8';

```

```

SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: dam; Type: SCHEMA; Schema: -; Owner: diaspara_admin
--

CREATE SCHEMA dam;

ALTER SCHEMA dam OWNER TO diaspara_admin;

--
-- Name: fk_id_batch_insert(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.fk_id_batch_insert() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM ob_id FROM "dam".observations WHERE ob_id = NEW.ba_ob_id;
    IF FOUND THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION '6- Invalid ba_ob_id (%)', NEW.ba_ob_id;
    END IF;
END
$$;

ALTER FUNCTION dam.fk_id_batch_insert() OWNER TO diaspara_admin;

--
-- Name: fk_id_environmental_characteristic_insert(); Type: FUNCTION; Schema: dam; Owner: d
--

CREATE FUNCTION dam.fk_id_environmental_characteristic_insert() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM op_id FROM "dam".observation_places WHERE op_id = NEW.ec_op_id;

```

```

IF FOUND THEN
    RETURN NEW;
ELSE
    RAISE EXCEPTION '5- Invalid ec_op_id (%)', NEW.ec_op_id;
END IF;
END
$$;

ALTER FUNCTION dam.fk_id_environmental_characteristic_insert() OWNER TO diaspara_admin;

--
-- Name: fk_id_observations_insert(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.fk_id_observations_insert() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM op_id FROM "dam".observation_places WHERE op_id = NEW.ob_op_id;
    IF FOUND THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION '5- Invalid ob_op_id (%)', NEW.ob_op_id;
    END IF;
END
$$;

ALTER FUNCTION dam.fk_id_observations_insert() OWNER TO diaspara_admin;

--
-- Name: integrity_observation_places_verify(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.integrity_observation_places_verify() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM ob_op_id FROM "dam".observations WHERE ob_op_id = OLD.op_id;
    IF FOUND THEN
        RAISE EXCEPTION
            '6- You try to delete an observation places record while there is some records in ob
/*
    ELSE

```

```

        PERFORM ec_op_id FROM "dam".environmental_characteristic WHERE ec_op_id = OLD.op_id;
        IF FOUND THEN
            RAISE EXCEPTION '6- You try to delete an observation places record while the
            ELSE
                RETURN OLD;
            END IF;
    */
    END IF;
END
$$;

ALTER FUNCTION dam.integrity_observation_places_verify() OWNER TO diaspara_admin;

--
-- Name: integrity_observations_verify(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.integrity_observations_verify() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM ba_ob_id FROM "dam".batch WHERE ba_ob_id = OLD.ob_id;
    IF FOUND THEN
        RAISE EXCEPTION
            '6- You try to delete an observation record while there is some records in batch re
    ELSE
        RETURN OLD;
    END IF;
END
$$;

ALTER FUNCTION dam.integrity_observations_verify() OWNER TO diaspara_admin;

--
-- Name: observation_id_place_insert(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.observation_id_place_insert() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM op_id FROM "dam".observation_places WHERE op_id = NEW.op_id;
    IF FOUND THEN

```

```

        RAISE EXCEPTION '2- Invalid op_id (%)', NEW.op_id;
    ELSE
        RETURN NEW;
    END IF;
END
$$;

ALTER FUNCTION dam.observation_id_place_insert() OWNER TO diaspara_admin;

--
-- Name: observation_id_place_update(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.observation_id_place_update() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    IF NEW.op_id = OLD.op_id THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION 'Do not update op_id (%)', OLD.op_id;
    END IF;
END
$$;

ALTER FUNCTION dam.observation_id_place_update() OWNER TO diaspara_admin;

--
-- Name: observations_id_insert(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.observations_id_insert() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    PERFORM ob_id FROM "dam".observations WHERE ob_id = NEW.ob_id;
    IF FOUND THEN
        RAISE EXCEPTION '3- Invalid ob_id (%)', NEW.ob_id;
    ELSE
        RETURN NEW;
    END IF;
END
$$;

```

```

ALTER FUNCTION dam.observations_id_insert() OWNER TO diaspara_admin;

--
-- Name: observations_id_update(); Type: FUNCTION; Schema: dam; Owner: diaspara_admin
--

CREATE FUNCTION dam.observations_id_update() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    IF NEW.ob_id = OLD.ob_id THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION 'Do not update ob_id (%)', OLD.ob_id;
    END IF;
END
$$;

ALTER FUNCTION dam.observations_id_update() OWNER TO diaspara_admin;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: bypass; Type: TABLE; Schema: dam; Owner: diaspara_admin
--

CREATE TABLE dam.bypass (
    bypass_id uuid NOT NULL,
    bypass_hpp_id uuid,
    bypass_water_depth numeric,
    bypass_width numeric,
    bypass_is_flowng boolean
);

ALTER TABLE dam.bypass OWNER TO diaspara_admin;

--
-- Name: data_provider; Type: TABLE; Schema: dam; Owner: postgres
--

```

```

CREATE TABLE dam.data_provider (
    dp_id integer NOT NULL,
    dp_name character varying(60),
    dp_edmo_key integer,
    dp_establishment_name text
);

ALTER TABLE dam.data_provider OWNER TO postgres;

--
-- Name: data_provider_dp_id_seq; Type: SEQUENCE; Schema: dam; Owner: postgres
--

CREATE SEQUENCE dam.data_provider_dp_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER SEQUENCE dam.data_provider_dp_id_seq OWNER TO postgres;

--
-- Name: data_provider_dp_id_seq; Type: SEQUENCE OWNED BY; Schema: dam; Owner: postgres
--

ALTER SEQUENCE dam.data_provider_dp_id_seq OWNED BY dam.data_provider.dp_id;

--
-- Name: fishway; Type: TABLE; Schema: dam; Owner: diaspara_admin
--

CREATE TABLE dam.fishway (
    fi_ob_id uuid NOT NULL,
    fi_op_id uuid,
    fi_id_original character varying(10),
    fi_species_id integer NOT NULL,
    fi_fishway_id integer,
    fi_presence_pass boolean,
    fi_date date
);

```

```

ALTER TABLE dam.fishway OWNER TO diaspara_admin;

--
-- Name: hpp; Type: TABLE; Schema: dam; Owner: postgres
--

CREATE TABLE dam.hpp (
    hpp_id uuid NOT NULL,
    hpp_ob_id uuid,
    hpp_name text,
    hpp_presence_bypass boolean,
    hpp_total_flow_bypass numeric,
    hpp_orient_flow_no_id integer,
    hpp_presence_of_bar_rack boolean,
    hpp_bar_rack_space numeric,
    hpp_surface_bar_rack numeric,
    hpp_inclination_bar_rack numeric,
    hpp_presence_bypass_trashrack boolean,
    hpp_nb_trashrack_bypass integer,
    hpp_turb_max_flow numeric,
    hpp_reserved_flow numeric,
    hpp_flow_trashrack_bypass numeric,
    hpp_max_power numeric,
    hpp_nb_turbines integer,
    hpp_turbine_type integer,
    hpp_orientation_bar_rack numeric,
    hpp_id_original text,
    hpp_source text
);

ALTER TABLE dam.hpp OWNER TO postgres;

--
-- Name: obstruction; Type: TABLE; Schema: dam; Owner: diaspara_admin
--

CREATE TABLE dam.obstruction (
    ob_id uuid DEFAULT public.uuid_generate_v4() NOT NULL,
    ob_starting_date date,
    ob_ending_date date,
    ob_op_id uuid NOT NULL,
    ob_obstruction_type_no_id integer,
    ob_fishway_type_no_id integer,

```

```

ob_mitigation_measure_no_id integer,
ob_height real,
ob_height_date date,
ob_downs_water_depth numeric,
ob_date_last_update date,
ob_event_change_no_id integer
);

ALTER TABLE dam.obstruction OWNER TO diaspara_admin;

--
-- Name: obstruction_place; Type: TABLE; Schema: dam; Owner: diaspara_admin
--

CREATE TABLE dam.obstruction_place (
  op_id uuid DEFAULT public.uuid_generate_v4() NOT NULL,
  op_placename text,
  op_op_id uuid,
  op_dp_id integer,
  op_id_original text NOT NULL,
  op_country character varying(2),
  geom public.geometry
);

ALTER TABLE dam.obstruction_place OWNER TO diaspara_admin;

--
-- Name: turbine; Type: TABLE; Schema: dam; Owner: postgres
--

CREATE TABLE dam.turbine (
  turb_id uuid NOT NULL,
  turb_hpp_id uuid,
  turb_turbine_type_no_id integer,
  turb_in_service boolean,
  turb_max_power numeric,
  turb_min_working_flow numeric,
  turb_hpp_height numeric,
  turb_diameter numeric,
  turb_rotation_speed numeric,
  turb_nb_blades integer,
  turb_max_turbine_flow numeric,
  turb_description text

```

```

);

ALTER TABLE dam.turbine OWNER TO postgres;

--
-- Name: data_provider dp_id; Type: DEFAULT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.data_provider ALTER COLUMN dp_id SET DEFAULT nextval('dam.data_provider');

--
-- Name: bypass bypass_pkey; Type: CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.bypass
    ADD CONSTRAINT bypass_pkey PRIMARY KEY (bypass_id);

--
-- Name: data_provider data_provider_pkey; Type: CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.data_provider
    ADD CONSTRAINT data_provider_pkey PRIMARY KEY (dp_id);

--
-- Name: fishway fishway_pkey; Type: CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.fishway
    ADD CONSTRAINT fishway_pkey PRIMARY KEY (fi_ob_id, fi_species_id);

--
-- Name: hpp hpp_pkey; Type: CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.hpp
    ADD CONSTRAINT hpp_pkey PRIMARY KEY (hpp_id);

--

```

```

-- Name: obstruction obstruction_pkey; Type: CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.obstruction
    ADD CONSTRAINT obstruction_pkey PRIMARY KEY (ob_id);

--

-- Name: obstruction_place obstruction_place_pkey; Type: CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.obstruction_place
    ADD CONSTRAINT obstruction_place_pkey PRIMARY KEY (op_id);

--

-- Name: turbine turbine_pkey; Type: CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.turbine
    ADD CONSTRAINT turbine_pkey PRIMARY KEY (turb_id);

--

-- Name: obstruction uk_obstruction; Type: CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.obstruction
    ADD CONSTRAINT uk_obstruction UNIQUE (ob_op_id, ob_starting_date);

--

-- Name: obstruction_place uk_op_id_original; Type: CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.obstruction_place
    ADD CONSTRAINT uk_op_id_original UNIQUE (op_id_original);

--

-- Name: obstruction_place_geom_gist; Type: INDEX; Schema: dam; Owner: diaspara_admin
--

CREATE INDEX obstruction_place_geom_gist ON dam.obstruction_place USING gist (geom);

```

```

--
-- Name: obstruction tr_obstruction_insert; Type: TRIGGER; Schema: dam; Owner: diaspara_admin
--

CREATE TRIGGER tr_obstruction_insert BEFORE INSERT ON dam.obstruction
FOR EACH ROW EXECUTE FUNCTION dam.observations_id_insert();

--
-- Name: obstruction tr_obstruction_update; Type: TRIGGER; Schema: dam; Owner: diaspara_admin
--

CREATE TRIGGER tr_obstruction_update BEFORE UPDATE ON dam.obstruction
FOR EACH ROW EXECUTE FUNCTION dam.observations_id_update();

--
-- Name: data_provider fk_dp_edmo_key; Type: FK CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.data_provider
    ADD CONSTRAINT fk_dp_edmo_key FOREIGN KEY (dp_edmo_key)
REFERENCES ref."EDMO"("Key") ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: obstruction fk_event_change_no_id; Type: FK CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.obstruction
    ADD CONSTRAINT fk_event_change_no_id FOREIGN KEY (ob_event_change_no_id)
REFERENCES nomenclature.event_change(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: fishway fk_fi_fishway_id; Type: FK CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.fishway
    ADD CONSTRAINT fk_fi_fishway_id FOREIGN KEY (fi_fishway_id)
REFERENCES nomenclature.fishway_type(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: fishway fk_fi_species_id; Type: FK CONSTRAINT; Schema: dam; Owner: diaspara_admin

```

```

--
ALTER TABLE ONLY dam.fishway
    ADD CONSTRAINT fk_fi_species_id FOREIGN KEY (fi_species_id)
REFERENCES nomenclature.species(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: hpp fk_hpp_ob_id; Type: FK CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.hpp
    ADD CONSTRAINT fk_hpp_ob_id FOREIGN KEY (hpp_ob_id)
REFERENCES dam.obstruction(ob_id) ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: hpp fk_hpp_orient_flow_no_id; Type: FK CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.hpp
    ADD CONSTRAINT fk_hpp_orient_flow_no_id FOREIGN KEY (hpp_orient_flow_no_id)
REFERENCES nomenclature.orient_flow(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: hpp fk_hpp_turbine_type; Type: FK CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.hpp
    ADD CONSTRAINT fk_hpp_turbine_type FOREIGN KEY (hpp_turbine_type)
REFERENCES nomenclature.turbine_type(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: obstruction fk_ob_fishway_type_no_id; Type: FK CONSTRAINT; Schema: dam; Owner: dias
--

ALTER TABLE ONLY dam.obstruction
    ADD CONSTRAINT fk_ob_fishway_type_no_id FOREIGN KEY (ob_fishway_type_no_id)
REFERENCES nomenclature.fishway_type(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: obstruction fk_ob_mitigation_measure_no_id; Type: FK CONSTRAINT; Schema: dam; Owner:

```

```

--
ALTER TABLE ONLY dam.obstruction
    ADD CONSTRAINT fk_ob_mitigation_measure_no_id FOREIGN KEY (ob_mitigation_measure_no_id)
REFERENCES nomenclature.downstream_mitigation_measure(no_id) ON UPDATE CASCADE ON DELETE RES

--
-- Name: obstruction fk_ob_obstruction_type_no_id; Type: FK CONSTRAINT; Schema: dam; Owner:
--

ALTER TABLE ONLY dam.obstruction
    ADD CONSTRAINT fk_ob_obstruction_type_no_id FOREIGN KEY (ob_obstruction_type_no_id)
REFERENCES nomenclature.obstruction_type(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: obstruction fk_ob_op_id; Type: FK CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.obstruction
    ADD CONSTRAINT fk_ob_op_id FOREIGN KEY (ob_op_id)
REFERENCES dam.obstruction_place(op_id) ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: obstruction_place fk_op_op_id; Type: FK CONSTRAINT; Schema: dam; Owner: diaspara_admin
--

ALTER TABLE ONLY dam.obstruction_place
    ADD CONSTRAINT fk_op_op_id FOREIGN KEY (op_op_id)
REFERENCES dam.obstruction_place(op_id) ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: turbine fk_turb_hpp_id; Type: FK CONSTRAINT; Schema: dam; Owner: postgres
--

ALTER TABLE ONLY dam.turbine
    ADD CONSTRAINT fk_turb_hpp_id FOREIGN KEY (turb_hpp_id)
REFERENCES dam.hpp(hpp_id) ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: turbine fk_turb_turbine_type_no_id; Type: FK CONSTRAINT; Schema: dam; Owner: postgres

```

```

--
ALTER TABLE ONLY dam.turbine
    ADD CONSTRAINT fk_turb_turbine_type_no_id FOREIGN KEY (turb_turbine_type_no_id)
REFERENCES nomenclature.turbine_type(no_id) ON UPDATE CASCADE ON DELETE RESTRICT;

--
-- Name: SCHEMA dam; Type: ACL; Schema: -; Owner: diaspara_admin
--

GRANT USAGE ON SCHEMA dam TO diaspara_read;

--
-- Name: FUNCTION fk_id_batch_insert(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.fk_id_batch_insert() TO diaspara_read;

--
-- Name: FUNCTION fk_id_environmental_characteristic_insert(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.fk_id_environmental_characteristic_insert() TO diaspara_read;

--
-- Name: FUNCTION fk_id_observations_insert(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.fk_id_observations_insert() TO diaspara_read;

--
-- Name: FUNCTION integrity_observation_places_verify(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.integrity_observation_places_verify() TO diaspara_read;

--
-- Name: FUNCTION integrity_observations_verify(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

```

```

GRANT ALL ON FUNCTION dam.integrity_observations_verify() TO diaspara_read;

--
-- Name: FUNCTION observation_id_place_insert(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.observation_id_place_insert() TO diaspara_read;

--
-- Name: FUNCTION observation_id_place_update(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.observation_id_place_update() TO diaspara_read;

--
-- Name: FUNCTION observations_id_insert(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.observations_id_insert() TO diaspara_read;

--
-- Name: FUNCTION observations_id_update(); Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT ALL ON FUNCTION dam.observations_id_update() TO diaspara_read;

--
-- Name: TABLE bypass; Type: ACL; Schema: dam; Owner: diaspara_admin
--

GRANT SELECT ON TABLE dam.bypass TO diaspara_read;

--
-- Name: TABLE data_provider; Type: ACL; Schema: dam; Owner: postgres
--

GRANT SELECT ON TABLE dam.data_provider TO diaspara_read;

```

```
--  
-- Name: TABLE fishway; Type: ACL; Schema: dam; Owner: diaspara_admin  
--  
GRANT SELECT ON TABLE dam.fishway TO diaspara_read;  
  
--  
-- Name: TABLE hpp; Type: ACL; Schema: dam; Owner: postgres  
--  
GRANT SELECT ON TABLE dam.hpp TO diaspara_read;  
  
--  
-- Name: TABLE obstruction; Type: ACL; Schema: dam; Owner: diaspara_admin  
--  
GRANT SELECT ON TABLE dam.obstruction TO diaspara_read;  
  
--  
-- Name: TABLE obstruction_place; Type: ACL; Schema: dam; Owner: diaspara_admin  
--  
GRANT SELECT ON TABLE dam.obstruction_place TO diaspara_read;  
  
--  
-- Name: TABLE turbine; Type: ACL; Schema: dam; Owner: postgres  
--  
GRANT SELECT ON TABLE dam.turbine TO diaspara_read;  
  
--  
-- PostgreSQL database dump complete  
--
```

## Chapter 10

# Annex 4 : Country dam table example

SQL code to create france schema

```
-- *****
-- Creating schema France (with diaspara_admin right)
-- *****

CREATE SCHEMA France;
ALTER SCHEMA France OWNER TO damdb_FranceManager;
GRANT USAGE ON SCHEMA France TO diaspara_read;
COMMENT ON SCHEMA France is 'Data from France';
```

SQL code to import france

```
CREATE SCHEMA dam_france;

-- dam_france.physical_obstruction definition

GRANT SELECT ON SCHEMA dam_france TO diaspara_read;
ALTER SCHEMA dam_france OWNER TO diaspara_admin;

INSERT INTO dam.data_provider
(dp_id, dp_name, dp_edmo_key, dp_establishment_name)
VALUES (1, 'Pierre Sagnes', NULL, 'Office français pour la biodiversité'); -- not sure th

-- dam_france.obstruction_place definition

-- Drop table
```

```

-- DROP TABLE dam_france.obstruction_place;

CREATE TABLE dam_france.obstruction_place(
  CONSTRAINT uk_op_id_original UNIQUE (op_id_original),
  CONSTRAINT obstruction_place_pkey PRIMARY KEY (op_id),
  --CONSTRAINT fk_op_op_id FOREIGN KEY(op_op_id)
  --REFERENCES dam_france.obstruction_place(op_id) ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT fk_op_country FOREIGN KEY (op_country)
  REFERENCES ref.tr_country_cou(cou_code) ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_op_dp_id FOREIGN KEY (op_dp_id)
  REFERENCES dam.data_provider(dp_id) ON UPDATE CASCADE ON DELETE RESTRICT
)
INHERITS (dam.obstruction_place);
CREATE INDEX obstruction_place_the_geom_idx ON dam_france.obstruction_place USING gist (geom)

INSERT INTO dam_france.obstruction_place
(op_id,
op_placename,
op_op_id,
op_dp_id,
op_id_original,
op_country,
geom)
SELECT
op_id,
op_placename,
op_op_id,
1 AS op_dp_id,
id_original AS op_id_original,
country,
the_geom AS geom
FROM
montepomi.dbeel_obstruction_place; --93359

GRANT SELECT ON TABLE dam_france.obstruction_place TO diaspara_read;
ALTER TABLE dam_france.obstruction_place OWNER TO diaspara_admin;

-- DROP TABLE dam_france.physical_obstruction;

CREATE TABLE dam_france.obstruction(
  CONSTRAINT nn_ob_ending_date CHECK ((ob_ending_date IS NOT NULL)),
  CONSTRAINT nn_ob_starting_date CHECK ((ob_starting_date IS NOT NULL)),

```

```

CONSTRAINT uk_ob_id_date UNIQUE (ob_id, ob_starting_date, ob_ending_date),
CONSTRAINT physical_obstruction_pkey PRIMARY KEY (ob_id),
CONSTRAINT fk_ob_op_id FOREIGN KEY (ob_op_id)
REFERENCES dam_france.obstruction_place(op_id) ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ob_fishway_type_no_id FOREIGN KEY (ob_fishway_type_no_id)
REFERENCES nomenclature.fishway_type(no_id) ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ob_mitigation_measure_no_id FOREIGN KEY (ob_mitigation_measure_no_id)
REFERENCES nomenclature.downstream_mitigation_measure(no_id) ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ob_event_change_no_id FOREIGN KEY (ob_event_change_no_id)
REFERENCES nomenclature.event_change(no_id) ON UPDATE CASCADE ON DELETE RESTRICT
)
INHERITS (dam.obstruction);

GRANT SELECT ON TABLE dam_france.obstruction TO diaspara_read;
ALTER TABLE dam_france.obstruction OWNER TO diaspara_admin;
--SELECT DISTINCT ouv_type_evt FROM montepomi.dbeel_physical_obstruction

INSERT INTO dam_france.obstruction
(ob_id,
ob_starting_date,
ob_ending_date,
ob_op_id,
ob_obstruction_type_no_id,
ob_fishway_type_no_id,
ob_mitigation_measure_no_id,
ob_height,
ob_height_date,
ob_downs_water_depth,
ob_date_last_update,
ob_event_change_no_id)
SELECT
ob_id,
ob_starting_date,
ob_ending_date,
ob_op_id,
ot_no_obstruction_type AS ob_obstruction_type_no_id,
NULL AS ob_fishway_type_no_id,
mitigation_measure_no_id AS ob_mitigation_measure_no_id,
po_obstruction_height AS ob_height,
h_chute_date_mesure AS ob_height_date,
po_downs_water_depth AS ob_downs_water_depth,
CURRENT_DATE AS ob_date_last_update,
CASE WHEN ouv_type_evt = 'date construction ouvrage' THEN 308
      WHEN ouv_type_evt = 'fin activité usage' THEN 311

```

```

        WHEN ouv_type_evt = 'début activité usage' THEN 309
        WHEN ouv_type_evt = 'derasement' THEN 313
        WHEN ouv_type_evt = 'arasement' THEN 310
        WHEN ouv_type_evt = 'dispositif de franchissement' THEN 332
        WHEN ouv_type_evt = 'date debut (default)' THEN 307
        ELSE NULL END AS ob_event_change_no_id
FROM montepomi.dbeel_physical_obstruction ; -- 99224

-- dam_france.hpp
-- DROP TABLE dam_france.hpp;

CREATE TABLE dam_france.hpp (
    CONSTRAINT c_pk_hpp_id PRIMARY KEY (hpp_id),
    CONSTRAINT c_fk_hpp_ob_id FOREIGN KEY (hpp_ob_id)
    REFERENCES dam_france.obstruction(ob_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT c_fk_hpp_orient_flow_no_id FOREIGN KEY (hpp_orient_flow_no_id)
    REFERENCES nomenclature.orient_flow(no_id)
    ON DELETE CASCADE ON UPDATE CASCADE
)
INHERITS (dam.hpp);

INSERT INTO dam_france.hpp
(hpp_id,
hpp_ob_id,
hpp_name,
hpp_presence_bypass,
hpp_total_flow_bypass,
hpp_orient_flow_no_id,
hpp_presence_of_bar_rack,
hpp_bar_rack_space,
hpp_surface_bar_rack,
hpp_inclination_bar_rack,
hpp_presence_bypass_trashrack,
hpp_nb_trashrack_bypass,
hpp_turb_max_flow,
hpp_reserved_flow,
hpp_flow_trashrack_bypass,
hpp_max_power,
hpp_nb_turbines,
hpp_turbine_type,
hpp_orientation_bar_rack,

```

```

hpp_id_original,
hpp_source)
SELECT
hpp_id,
hpp_ob_id,
hpp_name,
hpp_presence_bypass,
hpp_total_flow_bypass,
hpp_orient_flow_no_id,
hpp_presence_of_bar_rack,
hpp_bar_rack_space,
hpp_surface_bar_rack,
hpp_inclination_bar_rack,
hpp_presence_bypass_trashrack,
hpp_nb_trashrack_bypass,
hpp_turb_max_flow,
hpp_reserved_flow,
hpp_flow_trashrack_bypass,
hpp_max_power,
NULL AS hpp_nb_turbines,
NULL AS hpp_turbine_type,
hpp_orientation_bar_rack,
hpp_id_original,
hpp_source
FROM montepomi.dbeel_hpp ; -- 6909

GRANT SELECT ON TABLE dam_france.hpp TO diaspara_read;
ALTER TABLE dam_france.hpp OWNER TO diaspara_admin;
-- DROP TABLE dam_france.physical_obstruction_pass_species;

CREATE TABLE dam_france.fishway (
  CONSTRAINT fishway_pkey PRIMARY KEY (fi_ob_id, fi_species_id),
  CONSTRAINT fk_fi_fishway_id FOREIGN KEY (fi_fishway_id)
  REFERENCES nomenclature.fishway_type(no_id)
  ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT fk_fi_species_id FOREIGN KEY (fi_species_id)
  REFERENCES nomenclature.species(no_id)
  ON DELETE RESTRICT ON UPDATE CASCADE
) INHERITS (dam.fishway);

-- there are duplicates in the original table (two different fishways), this is wrong we ev
INSERT INTO dam_france.fishway
(fi_ob_id,
fi_op_id,

```

```

fi_id_original,
fi_species_id,
fi_fishway_id,
fi_presence_pass,
fi_date)
SELECT
DISTINCT ON (ob_id,species)
ob_id AS fi_ob_id,
op_id AS fi_op_id,
id_original AS fi_id_original,
CASE WHEN species = 'alose' THEN 317
      WHEN species = 'anguille' THEN 30
      WHEN species = 'saumon' THEN 315
      WHEN species = 'civelle' THEN 333
      WHEN species = 'lamproie' THEN 319
      END AS fi_fishway_id,
NULL AS fi_fishway_id,
po_presence_pass AS fi_presence_pass,
date_passe AS fi_date
FROM montepomi.dbeel_physical_obstruction_pass_species; --286255

GRANT SELECT ON TABLE dam_france.fishway TO diaspara_read;
ALTER TABLE dam_france.fishway OWNER TO diaspara_admin;

-- dam_france.bypass definition

-- Drop table

-- DROP TABLE dam_france.bypass;

CREATE TABLE dam_france.bypass (
  CONSTRAINT c_pk_bypass_id PRIMARY KEY (bypass_id),
  CONSTRAINT c_fk_bypass_hpp_id FOREIGN KEY (bypass_hpp_id)
  REFERENCES dam_france.hpp(hpp_id)
  ON DELETE CASCADE ON UPDATE CASCADE
)
INHERITS (dam.bypass);

GRANT SELECT ON TABLE dam_france.bypass TO diaspara_read;
ALTER TABLE dam_france.bypass OWNER TO diaspara_admin;

-- dam_france.turbine definition

-- Drop table

-- DROP TABLE dam_france.turbine;

```

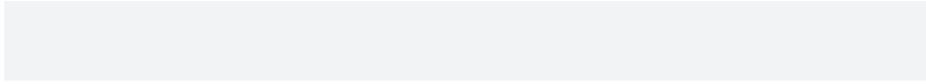
```

CREATE TABLE dam_france.turbine (
  CONSTRAINT c_pk_turb_id PRIMARY KEY (turb_id),
  CONSTRAINT c_fk_turb_hpp_id FOREIGN KEY (turb_hpp_id)
  REFERENCES dam_france.hpp(hpp_id) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT c_fk_turb_turbine_type_no_id FOREIGN KEY (turb_turbine_type_no_id)
  REFERENCES nomenclature.turbine_type(no_id) ON DELETE CASCADE ON UPDATE CASCADE
)
INHERITS (dam.turbine);

INSERT INTO dam_france.turbine(
turb_id,
turb_hpp_id,
turb_turbine_type_no_id,
turb_in_service,
turb_max_power,
turb_min_working_flow,
turb_hpp_height,
turb_diameter,
turb_rotation_speed,
turb_nb_blades,
turb_max_turbine_flow,
turb_description)
SELECT
turb_id,
turb_hpp_id,
turb_turbine_type_no_id,
turb_in_service,
turb_max_power,
turb_min_working_flow,
turb_hpp_height,
turb_diameter,
turb_rotation_speed,
turb_nb_blades,
turb_max_turbine_flow,
turb_description
FROM montepomi.dbeel_turbine ; --1139

GRANT SELECT ON TABLE dam_france.turbine TO diaspara_read;
ALTER TABLE dam_france.turbine OWNER TO diaspara_admin;

```



# Chapter 11

## Annex 5 : Database ER diagram

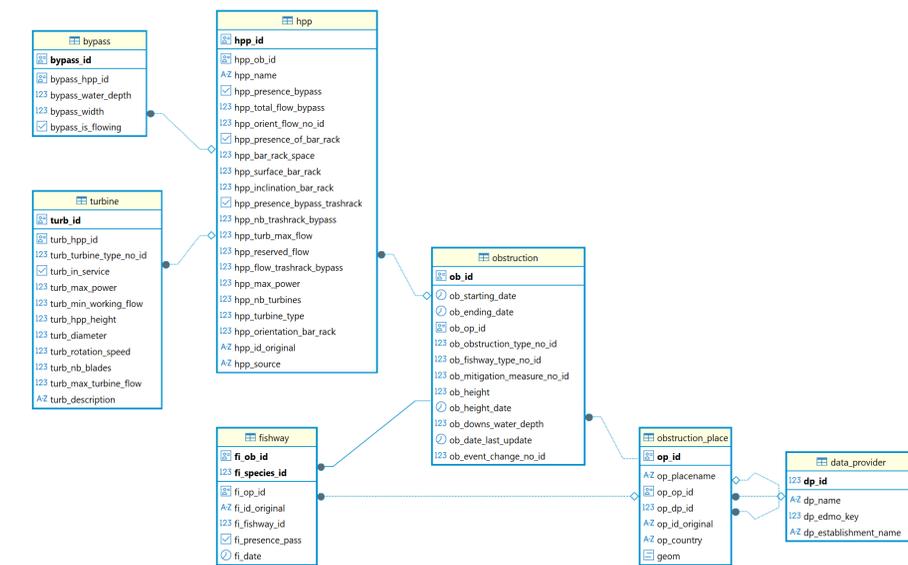


Figure 11.1: Entity Relationship database dam

Acknowledgements : The Ecohydraulique pole in Toulouse, Pierre Sagnes for support with the data and developing things during SUDOANG. Karl Kreuztenberger, for providing the various dumps to the french tables.

Walker, A. M., Eider Andonegi, P. Apolostolaki, M. Aprahamian, L. Beaulaton, D. Bevacqua, Cédric Briand, et al. 2011. "Pilot Projects to Estimate

Potential and Actual Escapement of Silver Eel.” DEFRA.