



DIASPARA WP3.4 FISH STOCK DATABASE : METRIC DB final report

version = pre-release, for
review. Still have to integrate
salmon from WP2

Briand Cédric, Oliviero Jules,
Helminen Jani

January 16, 2026

Contents

1	Building database structure from WGEEL	5
1.1	Station	5
1.1.1	SemanticRelation	5
1.1.2	PRGOV	6
1.1.3	EDMO	7
1.1.4	PURPM	7
1.1.5	DTYPE	8
1.1.6	WLTYP	9
1.1.7	MSTAT	9
1.1.8	Deprecated	10
1.2	Habitat	11
1.3	Fishway type ref.tr_fishway_fiw	14
1.4	Fish migration monitoring ref.tr_monitoring_mon	16
1.5	Fish Traits ref.tr_trait_tra	18
1.6	Numeric fish trait (tr_traitnumeric_trn)	20
1.7	Qualitative fish trait (tr_traitqualitative_trq)	25
1.8	Values of Qualitative fish trait (tr_traitvaluequal_trv)	28
1.9	Fish trait measurement method	32
1.10	Creating the station table	35
1.11	Creating the version table refeel.tr_version_ver	40
1.12	series table t_series_ser	42
1.12.1	Creating series main table for series	42
1.12.2	Creating inherited series	46
1.12.3	Import t_series_ser from wgeel	49
1.13	Annual series table	53
1.14	Creating inherited table for annual series	54
1.15	Import data for annual series	56
1.16	Creating table joining series and station	58
1.17	Creating inherited joining series and station	58
2	Creating additional table for stock annex	59
2.1	Creating table t_recruitmentmetadata	59

3	Creating group metrics	62
3.1	Create group table	62
3.2	Create group table for dateel	64
3.3	Import data for group	65
4	Grouptrait table	72
4.1	Create table	72
4.2	Import data	75
5	Creating ind metrics	89
5.1	Fish table	89
5.2	Table t_indtrait_int	93
6	Import data from WP2 (TODO)	107
7	WGBAST	108
7.1	series table t_series_ser	108
7.1.1	Creating inherited series	108
7.1.2	Import t_series_ser from trutta database	111
7.2	Creating table joining series and station	113
7.3	Creating inherited table for annual series	114
7.4	Import data for annual series	115
8	Acknowledgements and references	118

Note : If you are reading this file as a pdf you should try to open it on the diaspara website, as some code does not render well as a pdf : [metric database](#)

Aside the main database report, which describes all the vocabularies used in this document, we have to build a separate database for metrics. In the project these correspond to Life History Traits (LHT), but additionally, the data should also correspond to the time series and the sampling db developed by WGEEL. The two latter data structures (series and sampling) are very similar, and they both hold very similar group metrics and individual metrics.

The first was developed initially to store data about the series used in recruitment data. In practice, it consists of three tables, the **t_series_ser** (Figure Figure 1 - top in blue) table contains series id and description, with columns describing the sampling details, the stage used, the method... This is the main identifier of the series which will be used as a reference in all dependent tables. The second **t_dataserie_das** table (Figure Figure 1 - on the right) holds data about annual values in series. These are typically annual counts for recruitment, along with additional effort data. Linked to these are group metric series used to describe the series, mean age of eel, mean size, proportion of glass eel among the yellow eels, proportion of females ... (Figure Figure 1 - in orange) Finally, we can link individual metrics. The individual metrics are all detailed for one fish. And they concern metrics like size, weight, sex, but also can hold data about quality, contamination. So these are in essence the Life History traits analysed

[illegible]

The second type of data was developed to hold the data collected for DCF. These can be metrics collected from sampling by the fishermen, data coming from the analysis of electrofishing data, or other experimental sampling that are not reported as series. Currently the two structures for series and sampling are very close, the only difference is that there is no annual number linked to the sampling data, and that they are not linked to a stage in the first table, so the stage is added in the fish table. The difference in table structure is illustrated below in tables highlighted in yellow (Figure Figure 2).

- The first objective is to join the two database to simplify the database development and handling of data.
- The second objective is to use the new referentials created for the migddb database.
- The third objective is to import data from Diaspara WP2, the excel sheets have been created in february 2025 and will already (in March) require some adaptation as the database evolves, for instance the referential of stages is no longer in line with the templates.
- The fourth objective is to hand over this database, along with the migddb, to ICES, for integration in ICES database ecosystem, and use of DATSU in datacall.
- The fifth objective is to adapt the shiny scripts of data integration.

Chapter 1

Building database structure from WGEEL

[git issue #23 Write simplified structure from WGEEL](#)

The main issue will require to merge the two table structures (sampling and series) and adapt to migdb vocabulary.

Once done a beta version probably not completely adapted will be released.

[milestone metric DB beta version](#)

1.1 Station

To include the station, we need two tables: the first one is the Station dictionary and the second is the relation Dictionary (which relates two stations together). Now these tables in turn rely on some vocabs from ICES which we are loading straight as they are in our template database.

1.1.1 SemanticRelation

```
SemanticRelation <- getCodeList("SemanticRelation")
knitr::kable(SemanticRelation)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Table 1.1: Semantic relation

Id	Guid	Key	Description	LongDescription	Modified
169652	83921b19-6958-4619-b941-55fb3f207b8d	1	Broader		2017-02-27T12
169651	3420f8cf-1090-4810-80ee-7a144c26d3bb	2	Narrower		2017-02-27T12

169653	1d8976e7-7fb6-4dac-9b70-b282788e57c6	3	Related	2017-02-27T12
196324	4c8a38b4-2609-4a0a-aa49-b05148b1f315	4	SameAs	2018-05-30T13
196325	8c7c8ae7-1942-432e-9646-db0aef26498b	5	IsReplacedBy	2018-07-11T11
196764	d6d94c4f-d888-4a72-8152-16986a72d86d	6	Replaces	2018-07-11T11

```
dbWriteTable(con_diaspara, "SemanticRelation", PRGOV)
dbExecute(con_diaspara, 'ALTER TABLE "SemanticRelation" SET SCHEMA ref;')
dbExecute(con_diaspara,
  'ALTER TABLE ref."SemanticRelation" ADD CONSTRAINT semanticrelation_pkey PRIMARY KEY ("Key")')
```

1.1.2 PRGOV

```
PRGOV <- getCodeList("PRGOV")
knitr::kable(PRGOV)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Id	Guid	Key	Description
156691	47a15656-9a4c-4edc-9013-aeb76fc4405f	A	AMAP
196223	28a85b12-c702-40b9-a79c-6010184a7ba7	ASMT-MIME	Assessment Grouping for OSPAR MIM
156746	34f8c0f8-c096-45ed-a2c6-dfad117b44aa	BO	BONUS
156745	1cb8b3d2-6a52-4d6c-9e71-59170afb035	BO1	BONUS Beast
156722	bcdd09de-b915-4f10-97a1-d094f9544d44	FP7J	FP7 - Jericho
156689	83052fb7-d4e1-482f-a947-801e2273f478	H	HELCOM
156742	24f1673d-d7e1-4501-8104-665d4bf2fd93	H1	HELCOM BMP
156743	d29af7d1-4cc1-42ac-aea4-4c253ac0f139	H2	HELCOM COMBINE
202517	6ef57389-8fbe-4cd3-8bc7-660c77d84199	H3	HELCOM Continuous Underwater Noi
156690	e14c46d5-b1b0-48c7-89ef-ac8da49f4e0f	ICES	ICES
156860	6e18b63f-3105-4c21-b741-fa44b2a9be84	MP	MEDPOL Mediterranean Pollution and
261196	4af03f7e-aa21-41d0-99fe-a71a3839f621	MSFD	Marine Strategy Framework Directive
169793	e2f717e3-c74f-4d84-b6bc-237c9bed7cfa	MSFD-10	Marine Strategy Framework Directive
232239	bf985b9c-e5f9-4d38-ac5e-84781a0d9e77	MSFD-8	Marine Strategy Framework Directive
194868	26a82124-3a54-46e4-8637-b9671ee936c4	MSFD-9	Marine Strategy Framework Directive
156687	745bad18-04de-4053-a752-48003d53dd38	NTL	National
156688	35b9d6b2-f4b5-4cac-9e5f-df038848c45d	O	OSPAR
156740	84f3e3f0-a432-4264-b27a-8963b0f79c2f	O1	OSPAR JMP
156741	6bf2bb6d-5e90-4ab2-9b57-4d573ad6d491	O2	OSPAR CEMP
156744	2f5b26a2-1a19-47e9-a56a-8f7e57cf6b3a	SDN	SeaDataNet
156692	533f3f8f-df60-47fb-a2a8-74ebc0afb1e0	WFD	Water Framework Directive

Table 1.3: European Directory of Marine Organisations (EDMO)

```
EDMO <- getCodeList("EDMO")|>slice_head(n=10) |>
knitr::kable() |>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

```
dbWriteTable(con_diaspara, "PRGOV", PRGOV)
dbExecute(con_diaspara, 'ALTER TABLE "PRGOV" SET SCHEMA ref;')
dbExecute(con_diaspara,
'ALTER TABLE ref."PRGOV" ADD CONSTRAINT prgov_pkey PRIMARY KEY ("Key");')
```

1.1.3 EDMO

💡 Addings institutions to the EDMO

To request a new EDMO an organisation needs to send an email to info@maris.nl

```
dbWriteTable(con_diaspara, "EDMO", EDMO)
dbExecute(con_diaspara, 'ALTER TABLE "EDMO" SET SCHEMA ref;')
dbExecute(con_diaspara,
'ALTER TABLE ref."EDMO" ADD CONSTRAINT edmo_pkey PRIMARY KEY ("Key");')
```

1.1.4 PURPM

```
PURPM <- getCodeList("PURPM")
knitr::kable(PURPM)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Id	Guid	Key	Description
261197	21c70853-68e1-4c05-94fc-02c04c431011	8	MSFD Descriptior 8 - Contaminants
261198	b5758296-3a14-452f-aaa3-4d9367377b67	9	MSFD Descriptior 9 - Contaminants in seafood d
26936	0eed054b-b478-4c6e-b448-51a58c6ec967	B	Biological effects monitoring
26937	b677e830-34f8-4eaf-88e3-3377d726d209	E	Eutrophication effects monitoring
140653	18ad665b-1597-4cab-90ba-84445678ef14	F	Fishery trawl surveys
26938	3f0310df-3a9c-48fb-a1e3-8a47a918f1aa	H	Human health risk assessment
140654	89cd9460-1d43-4fd0-b658-7619a97bb864	L	Litter
26939	2451faa3-831b-43c6-9aff-939c701d864c	N	No specific purpose
137083	e01f97eb-6e49-4741-bc75-4ff29407584a	O	Ocean acidification
51445	f02b4faa-d081-4061-b8c8-81b339b6da2e	R	Research

26940	49df3df6-f880-4330-9707-a276eb92b4f2	S	Spatial (geographical) distribution monitoring
26941	562ae8f8-e175-4f07-ae92-a5af7b32539d	T	Temporal trend monitoring
198772	9efdc23c-153b-44bf-a978-55a406384eb2	U	Underwater noise monitoring

```
dbWriteTable(con_diaspara, "PURPM", PURPM)
dbExecute(con_diaspara, 'ALTER TABLE "PURPM" SET SCHEMA ref;')
dbExecute(con_diaspara,
'ALTER TABLE ref."PURPM" ADD CONSTRAINT purm_pkey PRIMARY KEY ("Key");')
```

1.1.5 DTYPE

```
Station_DTYPE <- getCodeList("Station_DTYPE")
knitr::kable(Station_DTYPE)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Table 1.5: Sta

Id	Guid	Key	Description
156767	9ae94eaa-67f5-4499-9fec-9c753d454d8b	BP	Bacterioplankton
148249	0a1d0292-367b-48f8-bfe4-d53c460c59b1	CF	Contaminants/hazardous substances in biota
148250	da00a888-4d78-40f3-bc37-3b2575f75f7b	CS	Contaminants/hazardous substances in sediment
202489	e03228f9-9f70-46b7-899f-226b02054f2b	CUWN	Continuous underwater noise
148251	8655c1bd-330a-4f7a-812c-f6cc0bd8f5a2	CW	Contaminants/hazardous substances in water
148256	85103227-9697-4487-a874-053e9d88de5d	DF	Disease in biota
148252	185f5a57-72e1-419b-b4ab-2a0b2aa0f899	EF	Biological effects in biota
148253	8bdccd4f-f0b7-401f-be43-9533a3fd72bf	ES	Biological effects in sediment
156766	6583e8b5-5d79-4ba2-94cb-099dd745e8eb	EU	Eutrophication effects
148254	8a758bca-6647-4b47-94a6-7d85ea48a29b	EW	Biological effects in water
148263	3edfcdff-974f-47fe-be3a-8f742fb5b0a7	HY	Physico-chemical parameters in water
148258	fad0afaa-28e9-40b5-ac9c-1b1a06f82bc5	LT	Litter data
148255	ba545513-a5dc-4087-bd05-71c28da65f54	NU	Nutrients in water
148257	3768b56d-d6c7-48cf-9056-4c03a33e233d	OA	Ocean acidification
148259	3ba04e2e-025d-45d1-abd9-71fdd3508793	PB	Phytobenthos
148260	5fb5a8fb-cfce-4bda-9335-825071df513d	PP	Phytoplankton
156768	fc02fa31-489b-4205-9dcf-bef7a1c9aded	PR	Primary production
148262	8f667c67-b741-4ee8-a363-51327e9b4799	ZB	Zoobenthos
148261	e5278079-7d6a-46b4-b844-4be4e4c0d344	ZP	Zooplankton

```
dbWriteTable(con_diaspara, "Station_DTYPE", Station_DTYPE)
dbExecute(con_diaspara, 'ALTER TABLE "Station_DTYPE" SET SCHEMA ref;')
dbExecute(con_diaspara,
'ALTER TABLE ref."Station_DTYPE" ADD CONSTRAINT station_dtype_pkey PRIMARY KEY ("Key");')
```

1.1.6 WLTP

```
WLTP <- getCodeList("WLTP")
knitr::kable(WLTP)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Table 1.0

Id	Guid	Key	Description
156363	9388a49e-4a5f-46b3-9b7b-8dcfb03e7b6f	BP	Beach - peri-urban
156362	a6e1119b-3bcc-4b0a-8053-f8f7a86f0053	BR	Beach - rural
156361	53708e1d-d3c6-47a7-a0ee-1c70e6fa7f9b	BU	Beach - urban
53327	86d9897d-6adc-430e-a689-f321b75cadcc	C	WFD Coastal water
53329	0a28de1c-018a-4841-b3a3-a68e01c81cd5	CE	Coastal water (Estuary)
54066	85a228c9-9033-4f97-accf-4968db78a90a	CF	Coastal water (Fjord)
53330	e57989c4-a3f9-4560-9460-60dce3a26185	CR	Coastal water (River)
252557	eb979616-8b95-4c1b-92e8-b5f8d8bf1b96	FW	Fresh water
53334	c557dc19-27b9-46b6-a164-d7d8d4f55738	L	Land station
134991	5b3da387-3b2b-47c4-9967-c3161f533207	LK	Lake
252556	b51355b0-b905-4b4e-ab12-98d9b47d7752	MC	Marine water (coast)
53333	a75522ef-5e4a-4e2d-8550-38091cb6c994	MO	Marine water (open sea)
252558	913ae617-a160-4687-a62c-8923c6762c4f	NA	Not applicable
53331	1c792737-6f55-422a-b709-88af2d78c4ea	T	WFD Transitional water - implies reduced salinity
53332	4f85f72f-c2f0-41c7-b966-c80c897b80d7	TT	Transitional water (Tidal) - significant tide and range

```
dbExecute(con_diaspara, 'DROP TABLE IF EXISTS ref."WLTP";')
WLTP$Key[is.na(WLTP$Key)] <- "NA"
dbWriteTable(con_diaspara, "WLTP", WLTP)
dbExecute(con_diaspara,
'ALTER TABLE "WLTP" SET SCHEMA ref;')
dbExecute(con_diaspara,
'ALTER TABLE ref."WLTP" ADD CONSTRAINT wltpe_pkey PRIMARY KEY ("Key");')
```

1.1.7 MSTAT

```
MSTAT <- getCodeList("MSTAT")
knitr::kable(MSTAT)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Id	Guid	Key	Description
----	------	-----	-------------

51864	35cd7508-3e7b-47a9-9b5d-ba043280e22e	B	WFD B - Baseline/Reference station
53326	00b2cc8b-aefa-4823-bbfc-6e6710f2a5c8	ID	Impacted directly from disruptions such as dredging
53325	57914639-50a3-4d7c-8fbd-cfd662ad76e9	IH	WFD I(HZ) - Impacted directly by discharges coming from
53810	fb7f3e57-3392-4b55-8077-5b3da96d7900	IH-A	where impact is aluminum industry (primary) (non-ferrous)
53811	7931ff8a-4257-4d19-a8e5-8bdf2060db93	IH-C	where impact is chemical/pharmaceutical industry
54097	1bdcff7c-10b3-4c71-b6fd-87cd51ac9e26	IH-D	dock where impact is antifoulants, fuel oil/petroleum
53814	7774858e-2b82-4457-b5a3-cb07a613fba9	IH-E	where impact is surface treatment/electroplating
53815	c14296ec-08ea-443d-8128-077f53bde913	IH-F	where impact is phosphogypsum-fertilizers
54098	28ccc88f-0f1e-4557-989e-21a097203f6d	IH-H	harbour where impact is antifoulants, fuel oil/petroleum
53816	a6f885b4-87b3-4bee-b4c0-4822fd459699	IH-I	where impact is iron and steel industry (primary)
53817	27eca6da-1b91-47b3-8758-3622b40dcf38	IH-M	where impact is metal industry (non-ferrous)
53818	98d3b8cd-0ea4-407d-bee4-28938d432c51	IH-O	where impact is oil/gas exploration/production platform
53819	6bd5f9c1-46cc-47b7-b151-f9590a1c9862	IH-P	where impact is paper and pulp industry
53820	9a3e08c1-ecee-48ad-97bd-d94af2fa6603	IH-S	where impact is iron and steel industry (secondary)
54099	6544bdd0-a2c6-45ae-bf9f-dbe1ab63319d	IH-W	where impact is from diffuse emissions from waste
51865	74356f36-785b-454b-b83a-7aa6a102ae3a	IP	WFD I(PHY) - Impacted directly by discharges coming from
53822	3ab88ae3-e7f4-424b-89bb-d1fe8c57753e	IP-B	where impact is fossil fuel burning/non-nuclear power plant
53823	a92b6a9a-d5f2-4d78-b37b-eea996c10f3c	IP-N	where impact is nuclear power plant
53824	75b8b6e4-7a79-4367-9094-648740baa137	IP-T	where impact is municipal waste water treatment
53324	ceb0b857-171a-49cc-84c6-49aa38e09590	RH	WFD R(HZ) - Representative of general conditions
53289	08354400-b079-4773-af80-134243623425	RP	WFD R(PHY) - Representative of general conditions

```
dbWriteTable(con_diaspara, "MSTAT", MSTAT)
dbExecute(con_diaspara,
'ALTER TABLE ref."MSTAT" ADD CONSTRAINT mstat_pkey PRIMARY KEY ("Key");')
```

1.1.8 Deprecated

```
Deprecated <- getCodeList("Deprecated")
knitr::kable(Deprecated) |>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Table 1.8: Deprecated

Id	Guid	Key	Description	LongDescription	Modified
169654	1f2a621c-b427-4aea-b66a-03437d3217bd	False	Not deprecated		2017-02-27T10:00:00Z
169655	0e88dd12-78b0-4ec6-a3e1-2178e0b22f2a	True	Deprecated		2017-02-27T10:00:00Z

1.2 Habitat

The habitat database is managed by [EIONET](#) Habitats directive Art. 17 reporting in year 2018. This is only for Natura2000 sites, a better options would be <https://eunis.eea.europa.eu/habitats-code-browser.jsp>

SQL code to create tables

```
-- this is a new referential

DROP TABLE IF EXISTS ref.tr_habitat_hab;
CREATE TABLE ref.tr_habitat_hab (
  hab_code TEXT PRIMARY KEY,
  hab_description TEXT,
  hab_definition TEXT,
  hab_icesvalue character varying(4),
  hab_icesguid uuid,
  hab_icestablesource text);

ALTER TABLE ref.tr_habitat_hab OWNER TO diaspara_admin;
GRANT SELECT ON ref.tr_habitat_hab TO diaspara_read;

COMMENT ON TABLE ref.tr_habitat_hab IS
  'Table of habitats from EIONET Habitats directive Art. 17 reporting in year 2018.';
COMMENT ON COLUMN ref.tr_habitat_hab.hab_code IS
  'Code for habitat';
COMMENT ON COLUMN ref.tr_habitat_hab.hab_description IS
  'Description of the habitat';
COMMENT ON COLUMN ref.tr_habitat_hab.hab_definition IS
  'Definition of the habitat';
COMMENT ON COLUMN ref.tr_habitat_hab.hab_icesvalue IS
  'Code for the habitat in the ICES database';
COMMENT ON COLUMN ref.tr_habitat_hab.hab_icesguid IS
  'GUID in the ICES database';
COMMENT ON COLUMN ref.tr_habitat_hab.hab_icestablesource IS
  'Source table in ICES vocab';
```

ICES / DIASPORA is this enough ?

The idea was to have estuaries and lagoons, but rivers are associated with vegetation, So I'm not sure if this vocab is enough ?

```
# Load required libraries
library(XML)

# Parse the XML file downloaded from https://dd.eionet.europa.eu/vocabulary/art17_2018/habitats
xml_file <- "R/data/habitats.xml" # Replace with your actual file path
```

```

#file.exists("R/data/habitats.xml")
xml_data <- XML::xmlParse(xml_file)

# Extract all <value> nodes
values <- getNodeSet(xml_data, "//value")

# Extract relevant data into a data frame
extract_info <- function(node) {
  id <- XML::xmlGetAttr(node, "id")
  code <- gsub("http://dd.eionet.europa.eu/vocabulary/art17_2018/habitats/",
x= id,replacement = "")
  label <- XML::xmlValue(node[["label"]])
  #defintion <- XML::xmlValue(node[["defintion"]])
  status_node <- XML::getNodeSet(node, ".//status/label")[[1]]
  status <- XML::xmlValue(status_node)
  return(data.frame(id = id, code = code, label = label,
status = status, stringsAsFactors = FALSE))
}

habitat <- do.call(rbind, lapply(values, extract_info))
habitat <- habitat[c(1:32,77:84,125:142),]

# Save the data frame as an RData file
save(habitat, file = "habitat_art_17_2018.RData")
# load(file = "habitat_art_17_2018.RData")
dbExecute(con_diaspara, 'DELETE FROM ref.tr_habitat_hab;')

dbWriteTable(con_diaspara, "habitat_temp", habitat)

dbExecute(con_diaspara,
'INSERT INTO "ref".tr_habitat_hab SELECT code, label, id, NULL,
NULL, NULL FROM habitat_temp;') #58
dbExecute(con_diaspara, "DROP TABLE habitat_temp")#0

habitat <- dbGetQuery(con_diaspara, "SELECT * FROM ref.tr_habitat_hab;")
knitr::kable(habitat)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))

```

hab_code	hab_description
HabAll	All habitats
HabBenAbyssal	Abyssal
HabBenAll	All benthic habitats
HabBenBathyalLowRock	Lower bathyal rock and biogenic reef

HabBenBathyalLowSed	Lower bathyal sediment
HabBenBathyalUpReef	Upper bathyal rock and biogenic reef
HabBenBathyalUpSed	Upper bathyal sediment
HabBenCircalittCoarSed	Circalittoral coarse sediment
HabBenCircalittMud	Circalittoral mud
HabBenCircalittMxdSed	Circalittoral mixed sediment
HabBenCircalittRock	Circalittoral rock and biogenic reef
HabBenCircalittSand	Circalittoral sand
HabBenInfralittCoarSed	Infralittoral coarse sediment
HabBenInfralittMud	Infralittoral mud
HabBenInfralittMxdSed	Infralittoral mixed sediment
HabBenInfralittRock	Infralittoral rock and biogenic reef
HabBenInfralittSand	Infralittoral sand
HabBenLitRock	Littoral rock and biogenic reef
HabBenLitSed	Littoral sediment
HabBenOffshCoarSed	Offshore circalittoral coarse sediment
HabBenOffshMud	Offshore circalittoral mud
HabBenOffshMxdSed	Offshore circalittoral mixed sediment
HabBenOffshRock	Offshore circalittoral rock and biogenic reef
HabBenOffshSand	Offshore circalittoral sand
HabBenOther	Other benthic habitats
HabOther	Other habitats
HabPelagAll	All pelagic habitats
HabPelagCoastal	Coastal
HabPelagOcean	Oceanic/beyond shelf
HabPelagOther	Other pelagic habitats
HabPelagShelf	Shelf
HabPelagVarSalinity	Variable salinity
1110	Sandbanks which are slightly covered by sea water all the time
1120	Posidonia beds (<i>Posidonia oceanica</i>)
1130	Estuaries
1140	Mudflats and sandflats not covered by seawater at low tide
1150	Coastal lagoons
1160	Large shallow inlets and bays
1170	Reefs
1180	Submarine structures made by leaking gases
3110	Oligotrophic waters containing very few minerals of sandy plains (<i>Littorelletalia</i>)
3120	Oligotrophic waters containing very few minerals generally on sandy soils of the
3130	Oligotrophic to mesotrophic standing waters with vegetation of the <i>Littorelletea</i>
3140	Hard oligo-mesotrophic waters with benthic vegetation of <i>Chara</i> spp.
3150	Natural eutrophic lakes with <i>Magnopotamion</i> or <i>Hydrocharition</i> - type vegetation
3160	Natural dystrophic lakes and ponds

3170	Mediterranean temporary ponds
3180	Turloughs
3190	Lakes of gypsum karst
3210	Fennoscandian natural rivers
3220	Alpine rivers and the herbaceous vegetation along their banks
3230	Alpine rivers and their ligneous vegetation with <i>Myricaria germanica</i>
3240	Alpine rivers and their ligneous vegetation with <i>Salix elaeagnos</i>
3250	Constantly flowing Mediterranean rivers with <i>Glaucium flavum</i>
3260	Water courses of plain to montane levels with the <i>Ranunculion fluitantis</i> and <i>Ca</i>
3270	Rivers with muddy banks with <i>Chenopodion rubri</i> p.p. and <i>Bidention</i> p.p. veget
3280	Constantly flowing Mediterranean rivers with <i>Paspalo-Agrostidion</i> species and h
3290	Intermittently flowing Mediterranean rivers of the <i>Paspalo-Agrostidion</i>

1.3 Fishway type ref.tr_fishway_fiw

SQL code to create table tr_fishway_fiw

```
-- this is a new referential

DROP TABLE IF EXISTS ref.tr_fishway_fiw;
CREATE TABLE ref.tr_fishway_fiw (
fiw_code TEXT PRIMARY KEY,
fiw_description TEXT,
fiw_definition TEXT,
fiw_icesvalue character varying(4),
fiw_icesguid uuid,
fiw_icestablesource text);
DELETE FROM ref.tr_fishway_fiw;
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('VS',
'Vertical slot fishway',
'Vertical Slot Fishways have top-to-bottom opening (slot) in the cross-wall by which water i
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('PO',
'Pool',
'A pool pass consists of a stepped channel divided by cross-walls that form a series of pool
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('FL',
'Fish lock',
'A fish lock, similar in structure to a ship lock, uses a lock chamber with inlet and outlet
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('D',
'Denil pass',
'compact, steeply sloped fish pass featuring angled baffles that create backflows to reduce
```

```

INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('RR',
'Rock ramp',
'Close-to_nature type, gently sloped rough-surfaced sill spanning the river width with a gentle slope');
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('ER',
'Eel ramp',
'Eel ladders/ramps, e.g., pipe-based systems laid through weirs and filled with baffles or boulders');
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('LA',
'Lateral canal',
'Bybass channel that uses lateral canals');
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('AR',
'Artificial river',
'Human-made channel designed to mimic the characteristics of a natural stream, allowing fish passage');
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('UN',
'Unknown',
'Unknown type fish passage');
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('S',
'Sluice',
'Periodically opening sluice gates that creates a strong, directed flow that attracts and allows fish passage');
INSERT INTO ref.tr_fishway_fiw (fiw_code, fiw_description, fiw_definition)
VALUES('HL',
'Hydraulic lift',
'A type of fish elevator that uses hydraulic mechanisms-such as pumps, valves, and water pressure to lift fish');

ALTER TABLE ref.tr_fishway_fiw OWNER TO diaspara_admin;
GRANT SELECT ON ref.tr_fishway_fiw TO diaspara_read;

COMMENT ON TABLE ref.tr_fishway_fiw IS 'Table of fishway type';
COMMENT ON COLUMN ref.tr_fishway_fiw.fiw_code IS 'Code for fishway type';
COMMENT ON COLUMN ref.tr_fishway_fiw.fiw_description IS 'Description of the fishway';
COMMENT ON COLUMN ref.tr_fishway_fiw.fiw_definition IS 'Definition of the fishway';
COMMENT ON COLUMN ref.tr_fishway_fiw.fiw_icesvalue IS 'Code for the fishway in the ICES database';
COMMENT ON COLUMN ref.tr_fishway_fiw.fiw_icesguid IS 'GUID in the ICES database';

habitat <- dbGetQuery(con_diaspara, "SELECT * FROM ref.tr_fishway_fiw;")
knitr::kable(habitat)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))

```

fiw_code	fiw_description	fiw_definition
VS	Vertical slot fishway	Vertical Slot Fishways have top-to-bottom opening (slot) in the cross-wall
PO	Pool	A pool pass consists of a stepped channel divided by cross-walls that form
FL	Fish lock	A fish lock, similar in structure to a ship lock, uses a lock chamber with i
D	Denil pass	compact, steeply sloped fish pass featuring angled baffles that create back
RR	Rock ramp	Close-to_nature type, gently sloped rough-surfaced sill spanning the river
ER	Eel ramp	Eel ladders/ramps, e.g., pipe-based systems laid through weirs and filled
LA	Lateral canal	Bybass channel that uses lateral canals
AR	Artificial river	Human-made channel designed to mimic the characteristics of a natural s
UN	Unknown	Unknown type fish passage
S	Sluice	Periodically opening sluice gates that creates a strong, directed flow that
HL	Hydraulic lift	A type of fish elevator that uses hydraulic mechanisms”such as pumps,

1.4 Fish migration monitoring ref.tr_monitoring_mon

SQL code to create table tr_monitoring_mon

```
-- this is a new referential

DROP TABLE IF EXISTS ref.tr_monitoring_mon;
CREATE TABLE ref.tr_monitoring_mon (
mon_code TEXT PRIMARY KEY,
mon_description TEXT,
mon_definition TEXT,
mon_icesvalue character varying(4),
mon_icesguid uuid,
mon_icestablesource text);
DELETE FROM ref.tr_monitoring_mon;
INSERT INTO ref.tr_monitoring_mon (mon_code, mon_description, mon_definition)
VALUES('SO',
'Sonar HF',
'Multibeam, high frequency sonar used to monitor fish migration, e.g. Didson, Aris, Blueview);
INSERT INTO ref.tr_monitoring_mon (mon_code, mon_description, mon_definition)
VALUES('TR',
'Trap',
'Trap used to catch a part or the whole run. Fish can be directed towards the trap with a sy
or within a fishway. The fishes are counted and measured and then released, most often to c
upstream or downstream from the trap.');
```

```

'Video recording',
'Video recording of the fish used to make specific identification, and measure the length and weight.
The video is often located in a narrow passage within a fishway, often also with light set in the passage.
INSERT INTO ref.tr_monitoring_mon (mon_code, mon_description, mon_definition)
VALUES('RC',
'Resistivity counter',
'Resistivity counters monitor the resistance between electrodes to detect the passage of a fish.
The lower resistance of the fish compared to the water is used to detect the passage.
A series of submerged electrodes are used to detect the direction. Automatic
adjustment of the sensitivity of the counter ensures that the sizes into which fish are
classified remains consistent.');
```

```

INSERT INTO ref.tr_monitoring_mon (mon_code, mon_description, mon_definition)
VALUES('AC',
'Telemetry',
'e.g., Acoustic, PIT, or radiotelemetry receivers are use in fish tracking to detect and de
INSERT INTO ref.tr_monitoring_mon (mon_code, mon_description, mon_definition)
VALUES('IR',
'Infrared counter',
'Counter that uses infrared, e.g., Vaki Riverwatcher');
```

```

ALTER TABLE ref.tr_monitoring_mon OWNER TO diaspara_admin;
GRANT SELECT ON ref.tr_monitoring_mon TO diaspara_read;
```

```

COMMENT ON TABLE ref.tr_monitoring_mon IS 'Table of monitoring devices.
A monitoring device is used to monitor fish passage. It can be attached to a fishway.
A fishway can have several monitoring devices, e.g. a trap and a video recording.
A monitoring device can also be placed without fishway , e.g. a sonar, an accoustic receiver
COMMENT ON COLUMN ref.tr_monitoring_mon.mon_code IS 'Code for monitoring type';
COMMENT ON COLUMN ref.tr_monitoring_mon.mon_description IS 'Description of the monitoring device';
COMMENT ON COLUMN ref.tr_monitoring_mon.mon_definition IS 'Definition of the monitoring device';
COMMENT ON COLUMN ref.tr_monitoring_mon.mon_icesvalue IS 'Code for the fishwat in the ICES database';
COMMENT ON COLUMN ref.tr_monitoring_mon.mon_icesguid IS 'GUID in the ICES database';
```

```

habitat <- dbGetQuery(con_diaspara,
"SELECT * FROM ref.tr_monitoring_mon;")
knitr::kable(habitat)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

mon_code	mon_description	mon_definition
SO	Sonar HF	Multibeam, high frequency sonar used to monitor fish migration, c
TR	Trap	Trap used to catch a part or the whole run. Fish can be directed t
VR	Video recording	Video recording of the fish used to make specific identification, an
RC	Resistivity counter	Resistivity counters monitor the resistance between electrodes to c
AC	Telemetry	e.g., Acoustic, PIT, or radiotelemetry receivers are use in fish trac
IR	Infrared counter	Counter that uses infrared, e.g., Vaki Riverwatcher
OC	Other catching methods	Any other type of fish catching method than a trap. E.g. rod, gill
VC	Visual count	Visual fish counting methods, not including with the use of camer
IQ	Interview or questionnaire	Interview or questionnaire of e.g. recreational fishers or other part
MK	Mark and Recapture	Mark and recapture methods
CD	Catch data	Recreational or commercial fisheries catch data

1.5 Fish Traits ref.tr_trait_tra

This one was formerly called metric, but metric `ref.tr_metric_met` already exist and is better suited to statistics (mean, sd ...). So we are using the word trait. Trait is referring to a species (there are different method, or different measurements for different species). The trait can be, either qualitative or quantitative. The trait is measured by a method, the measure (either categorical eg. sex = 'Male') or quantitative e.g. (length_mm = 150) might be accompanied by a method. For instance a fish is measured for length as fork length or total length, in the individual_trait table, the fish will be related to three columns, the code of the length parameter in the trait table, the quantitative value, and the method used.

SQL code to create tables `ref.tr_trait_tra` and `refeel.tg_trait_tra`

```
-- DROP TABLE ref.tr_trait_tra CASCADE;

CREATE TABLE ref.tr_trait_tra (
  tra_id integer PRIMARY KEY,
  tra_code text NOT NULL,
  tra_description text NULL,
  tra_wkg_code TEXT NOT NULL,
  CONSTRAINT fk_tra_wkg_code FOREIGN KEY (tra_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  tra_spe_code TEXT NOT NULL,
  CONSTRAINT fk_tra_spe_code FOREIGN KEY (tra_spe_code)
  REFERENCES ref.tr_species_spe(spe_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
```

```

tra_typedmetric text NULL,
CONSTRAINT ck_tra_typedmetric CHECK (((tra_typedmetric = 'Individual'::text)
OR (tra_typedmetric = 'Group'::text)
OR (tra_typedmetric = 'Both'::text))),
tra_qualitativeornumeric TEXT,
CONSTRAINT ck_tra_qualitativeornumeric CHECK ((tra_qualitativeornumeric = 'Qualitative'::text)
OR (tra_qualitativeornumeric = 'Numeric'::text)),
CONSTRAINT uk_tra_code UNIQUE (tra_code)
);

COMMENT ON COLUMN ref.tr_trait_tra.tra_id IS
'Integer, id of the trait';
COMMENT ON COLUMN ref.tr_trait_tra.tra_code IS
'Name of the trait';
COMMENT ON COLUMN ref.tr_trait_tra.tra_description IS
'Description of the fish trait';
COMMENT ON COLUMN ref.tr_trait_tra.tra_typedmetric IS
'Is the metric a group metric (group), or individual metric (individual) or can be used in both';
COMMENT ON COLUMN ref.tr_trait_tra.tra_qualitativeornumeric IS
'Indicate variable type, either Qualitative or Numeric';
COMMENT ON COLUMN ref.tr_trait_tra.tra_spe_code IS
'AphiaID of the species code TEXT (e.g.'127186'' Salmo salar )';

GRANT ALL ON ref.tr_trait_tra TO diaspara_admin;
GRANT SELECT ON ref.tr_trait_tra TO diaspara_read;

/*
note The refeel.tg_trait_tra actually contains physically all parms and that's not the case
ref.tr_trait_tra which only gets those by inheritance.
refeel.tg_trait_tra must be created after insertion in tr_traitnumeric_trn
and tr_traitqualitative_trq
*/

CREATE TABLE refeel.tg_trait_tra AS (
SELECT
tra_id,
tra_code,
tra_description,
tra_wkg_code,
tra_spe_code ,
tra_typedmetric,
tra_qualitativeornumeric FROM
refeel.tr_traitnumeric_trn

```

```

UNION
SELECT
    tra_id,
    tra_code,
    tra_description,
    tra_wkg_code,
    tra_spe_code ,
    tra_typedmetric,
    tra_qualitativeornumeric FROM
    refeel.tr_traitqualitative_trq);
ALTER TABLE refeel.tg_trait_tra
ADD CONSTRAINT uk_tra_code UNIQUE (tra_code);

```

⚠ `refeel.tg_trait_tra` and `ref.tr_trait_tra` have different structures.

The first has nothing inside, it's a shell for inheritance. Tables `ref.tr_traitnumeric_trn` and `ref.tr_traitqualitative_trq` are inherited from `ref.tr_trait_tra`. They in turn are inherited from tables `refeel.tr_traitnumeric_trn` and `refeel.tr_traitqualitative_trq`. So the values entered in `refeel.tr_traitqualitative_trq` appear when running a select query from `ref.tr_trait_tra` even if they are not physically in these tables, and foreign key constraint to `ref.tr_trait_tra` would return no value there. Now if I inherit `refeel.tr_trait_tra` the values will get to `ref.tr_trait_tra` from two sources and I will double the lines there. Also, physically data are not there. So I still cannot use it as a reference table. So the table `refeel.tg_trait_tra` is not inherited, it's a grouping table with the values actually physically there.

1.6 Numeric fish trait (`tr_traitnumeric_trn`)

Trait can be either qualitative or quantitative (numeric). Both tables are inherited from `tr_trait_tra`.

SQL code to create tables `ref.tr_traitnumeric_trn` and `refeel.tr_traitnumeric_trn`

```

-- DROP TABLE IF EXISTS ref.tr_traitnumeric_trn;
CREATE TABLE ref.tr_traitnumeric_trn(
trn_uni_code VARCHAR(20) NULL,
trn_minvalue NUMERIC,
trn_maxvalue NUMERIC,
CONSTRAINT fk_tra_wkg_code FOREIGN KEY (tra_wkg_code)
REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
ON UPDATE CASCADE ON DELETE RESTRICT,

```

```

CONSTRAINT fk_tra_spe_code FOREIGN KEY (tra_spe_code)
REFERENCES ref.tr_species_spe(spe_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_trn_uni_code FOREIGN KEY (trn_uni_code)
REFERENCES ref.tr_units_uni(uni_code) ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT uk_trn_code UNIQUE (tra_code)
) INHERITS (ref.tr_trait_tra);

COMMENT ON COLUMN ref.tr_traitnumeric_trn.tra_id IS
'Integer, id of the trait';
COMMENT ON COLUMN ref.tr_traitnumeric_trn.tra_code IS
'Name of the trait';
COMMENT ON COLUMN ref.tr_traitnumeric_trn.tra_description IS
'Description of the fish trait';
COMMENT ON COLUMN ref.tr_traitnumeric_trn.tra_typedmetric IS
'Is the metric a Group metric (group), or Individual metric (individual) or can be used in
COMMENT ON COLUMN ref.tr_traitnumeric_trn.trn_uni_code IS
'Unit used, references tr_unit_uni';
COMMENT ON COLUMN ref.tr_traitnumeric_trn.trn_minvalue IS
'Minimum allowed value';
COMMENT ON COLUMN ref.tr_traitnumeric_trn.trn_maxvalue IS
'Maximum allowed value';

GRANT ALL ON ref.tr_traitnumeric_trn TO diaspara_admin;
GRANT SELECT ON ref.tr_traitnumeric_trn TO diaspara_read;

DROP TABLE IF EXISTS refeel.tr_traitnumeric_trn;
CREATE TABLE refeel.tr_traitnumeric_trn(
CONSTRAINT uk_refeel_tra_id UNIQUE (tra_id),
CONSTRAINT uk_refell_tra_code UNIQUE(tra_code),
CONSTRAINT fk_tra_wkg_code FOREIGN KEY (tra_wkg_code)
REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_tra_spe_code FOREIGN KEY (tra_spe_code)
REFERENCES ref.tr_species_spe(spe_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_trn_uni_code FOREIGN KEY (trn_uni_code)
REFERENCES ref.tr_units_uni(uni_code)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT uk_trn_code UNIQUE (tra_code)
) INHERITS (ref.tr_traitnumeric_trn);

COMMENT ON COLUMN refeel.tr_traitnumeric_trn.tra_id IS

```

```

'Integer, id of the trait';
COMMENT ON COLUMN refeel.tr_traitnumeric_trn.tra_code IS
'Name of the trait';
COMMENT ON COLUMN refeel.tr_traitnumeric_trn.tra_description IS
'Description of the fish trait';
COMMENT ON COLUMN refeel.tr_traitnumeric_trn.tra_typemetric IS
'Is the metric a Group metric (group), or Individual metric (individual) or can be used in
COMMENT ON COLUMN refeel.tr_traitnumeric_trn.trn_uni_code IS
'Unit used, references tr_unit_uni';
COMMENT ON COLUMN refeel.tr_traitnumeric_trn.trn_minvalue IS
'Minimum allowed value';
COMMENT ON COLUMN refeel.tr_traitnumeric_trn.trn_maxvalue IS
'Maximum allowed value';

GRANT ALL ON refeel.tr_traitnumeric_trn TO diaspara_admin;
GRANT SELECT ON refeel.tr_traitnumeric_trn TO diaspara_read;

tra <- dbGetQuery(con_wgeel_local, "SELECT * FROM ref.tr_metrictype_mty ;")
# we will include group metric names later
tra <- tra[(!grepl("mean", tra$mtty_name) |
             tra$mtty_name == 'eye_diam_meanmm' |
             tra$mtty_name == 'teq'),]
# we will also include method later
tra <- tra[!grepl("method", tra$mtty_name),]
res <- dbGetQuery(con_diaspara, "SELECT * FROM ref.tr_trait_tra")
#clipr::write_clip(colnames(res))
tranum <- data.frame(
  "tra_id" = tra$mtty_id,
  "tra_code" = stringi::stri_trans_totitle(tra$mtty_name),
  "tra_description" = tra$mtty_description,
  "tra_wkg_code" = 'WGEEL',
  "Tra_spe_code" = 'ANG',
  "trn_uni_code" = tra$mtty_uni_code,
  "tra_typemetric"=stringi::stri_trans_totitle(tra$mtty_group),
  "tra_qualitativeornumeric"='Numeric',
  "trn_minvalue" = tra$mtty_min,
  "trn_maxvalue" = tra$mtty_max)

# the proportions will be numeric for group and qualitative for individual (
# e.g. evex presence)

tranum[tranum$tra_typemetric=="Both" & grepl("proportion",tranum$tra_code),
c("tra_typemetric")] <- "Group"

```

```

# view(tranum)
# fix names
tranum$tra_description[tranum$tra_code=="Lengthmm"] <-
'Total body length in millimeters (mm) or mean total body length for group'
tranum$tra_description[tranum$tra_code=="Differentiated_proportion"] <-
'Proportion of differentiated eel (between 0 and 1)'

tranum$tra_description[tranum$tra_code=="Female_proportion"] <-
'Female proportion in the population female/(male+female) for group (between 0 and 1)'
tranum$tra_description[tranum$tra_code=="Anguillicola_proportion"] <-
'Prevalence of Anguillicola in proportion in group (between 0 and 1)'
tranum$tra_description[tranum$tra_code=="Evex_proportion"] <-
'EVE and EVEX proportion in the group (between 0 and 1)'
tranum$tra_description[tranum$tra_code=="Hva_proportion"] <-
'HVA proportion in the group (between 0 and 1)'
# duplicated, now two methods
tranum <- tranum[tranum$tra_id!=11,]
tranum$tra_description[tranum$tra_id==10] <-
"Lipid percentage or mean muscle lipid percentage for group"
tranum[tranum$tra_id==10,"tra_code"] <- "Muscle_lipid"

dbWriteTable(con_diaspara_admin, "tr_traitnumeric_trn_temp", tranum, overwrite =TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM refeel.tr_traitnumeric_trn")
dbExecute(con_diaspara_admin, "INSERT INTO refeel.tr_traitnumeric_trn(
tra_id,
tra_code,
tra_wkg_code,
tra_spe_code,
tra_description,
tra_typemetric,
trn_uni_code,
tra_qualitativeornumeric,
trn_minvalue,
trn_maxvalue)
SELECT
tra_id,
tra_code,
'WGEEL',
'ANG',
tra_description,
tra_typemetric,
trn_uni_code,
tra_qualitativeornumeric,
trn_minvalue,

```

```

trn_maxvalue
FROM tr_traitnumeric_trn_temp") #18

dbExecute(con_diaspara_admin,
"INSERT INTO refeel.tr_traitnumeric_trn (tra_id,tra_code,tra_description,
tra_wkg_code,tra_spe_code,tra_typemetric,tra_qualitativeornumeric,
trn_uni_code,trn_minvalue,trn_maxvalue)
VALUES (27,'Eye_diam_vert_mm','Eye diameter (vertical) in mm','WGEEL',
'ANG','Individual','Numeric','mm',1,15);
INSERT INTO refeel.tr_traitnumeric_trn (tra_id,tra_code,tra_description,
tra_wkg_code,tra_spe_code,tra_typemetric,tra_qualitativeornumeric,
trn_uni_code,trn_minvalue,trn_maxvalue)
VALUES (28,'Eye_diam_horiz_mm','Eye diameter (horizontal) in mm',
'WGEEL','ANG','Individual','Numeric','mm',1,15);
") #18

dbExecute(con_diaspara_admin, "UPDATE refeel.tr_traitnumeric_trn
SET tra_code='Eye_diam_mean_mm'
WHERE tra_id=4;")

```

tra_id	tra_code	tra_description
1	Lengthmm	Total body length in millimeters (mm) or mean total body length for
2	Weightg	Weight (g) or mean weight for group
3	Ageyear	Age (year) or mean age for group
5	Pectoral_lengthmm	Pectoral fin length (mm)
7	Differentiated_proportion	Proportion of differentiated eel (between 0 and 1)
9	Anguillicola_intensity	A. crassus intensity or mean A. crassus intensity for group
10	Muscle_lipid	Lipid percentage or mean muscle lipid percentage for group
12	Sum_6_pcb	Sum of six PCBs or mean sum of six PCBs for groups
13	Evex_proportion	EVE and EVEX proportion in the group (between 0 and 1)
14	Hva_proportion	HVA proportion in the group (between 0 and 1)
15	Pb	Lead (Pb) concentration or mean lead (Pb) concentration
16	Hg	Mercury (Hg) concentration or mean lead (Pb) concentration for group
17	Cd	Cadmium (Cd) concentration or mean cadmium concentration for group
24	G_in_gy_proportion	Proportion of glass eel in number during the season when the series is
25	S_in_ys_proportion	Proportion of silver eel in number in the group
26	Teq	Sum TEQ of measured dioxin-like PCBs or mean sum TEQ of measured
6	Female_proportion	Female proportion in the population female/(male+female) for group
8	Anguillicola_proportion	Prevalence of Anguillicola in proportion in group (between 0 and 1)

27	Eye_diam_vert_mm	Eye diameter (vertical) in mm
28	Eye_diam_horiz_mm	Eye diameter (horizontal) in mm
4	Eye_diam_mean_mm	Eye diameter, or average of vertical and horizontal diameter (mm)

1.7 Qualitative fish trait (tr_traitqualitative_trq)

SQL code to create table `ref.tr_traitqualitative_trq` and `refeel.tr_traitqualitative_trq`

```
-- DROP TABLE IF EXISTS ref.tr_traitqualitative_trq CASCADE;
```

```
CREATE TABLE ref.tr_traitqualitative_trq (
  CONSTRAINT fk_tra_wkg_code FOREIGN KEY (tra_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_tra_spe_code FOREIGN KEY (tra_spe_code)
  REFERENCES ref.tr_species_spe(spe_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT uk_trq_code UNIQUE (tra_code)
) INHERITS (ref.tr_trait_tra);
```

```
COMMENT ON TABLE ref.tr_traitqualitative_trq IS
'Table of qualitative trait parameters';
```

```
GRANT ALL ON ref.tr_traitqualitative_trq TO diaspara_admin;
GRANT SELECT ON ref.tr_traitqualitative_trq TO diaspara_read;
```

```
DROP TABLE IF EXISTS refeel.tr_traitqualitative_trq;
CREATE TABLE refeel.tr_traitqualitative_trq (
  CONSTRAINT uk_refeel_tra_id UNIQUE (tra_id),
  CONSTRAINT uk_refeel_tra_code UNIQUE(tra_code),
  CONSTRAINT fk_tra_wkg_code FOREIGN KEY (tra_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_tra_spe_code FOREIGN KEY (tra_spe_code)
  REFERENCES ref.tr_species_spe(spe_code)
  ON UPDATE CASCADE ON DELETE RESTRICT
) INHERITS (ref.tr_traitqualitative_trq);
```

```
COMMENT ON TABLE refeel.tr_traitqualitative_trq IS
'Table of qualitative trait parameters';
```

```
GRANT ALL ON refeel.tr_traitqualitative_trq TO diaspara_admin;
GRANT SELECT ON refeel.tr_traitqualitative_trq TO diaspara_read;
```

```
tra <- dbGetQuery(con_wgeel_local, "SELECT * FROM ref.tr_metrictype_mty ;")
# we will include group metric names later
tra <- tra[grepl("is_", tra$mtty_individual_name) |
  grepl("presence", tra$mtty_individual_name),]
# we will also include method later

#clipr::write_clip(colnames(res))
traqal <- data.frame(
  "tra_id" = tra$mtty_id,
  "tra_code" = stringi::stri_trans_totitle(tra$mtty_individual_name),
  "tra_description" = tra$mtty_description,
  "tra_wkg_code" = 'WGEEL',
  "Tra_spe_code" = 'ANG',
  "tra_uni_code" = tra$mtty_uni_code,
  "tra_typedmetric"='Individual',
  "tra_qualitativeornumeric"='Qualitative'
)

# the proportions will be numeric for group and qualitative for individual (
# e.g. evex presence)

traqal$tra_code <- gsub("\\s*_\\\[^\]\\]+\\\[", "", traqal$tra_code)
traqal$tra_code[traqal$tra_code=="Is_female"] <- "Sex"
traqal$tra_individualname <- traqal$tra_code

# fix names

traqal$tra_description[traqal$tra_code=="Is_differentiated"] <-
'Is the eel differentiated (Y Yes, N No, NA Not applicable,P probable, U Unknown)'
traqal$tra_description[traqal$tra_code=="Sex"] <-
'Sex (F Female,
H Hermaphordite,
I Immature attempt made,M Male,
T Transitional,
U Undetermined no attempt made,
X Mixed)'
traqal$tra_description[traqal$tra_code=="Anguillicola_presence"] <-
'Presence of Anguillicola (Y Yes, N No, NA Not applicable,P probable, U Unknown)'
traqal$tra_description[traqal$tra_code=="Evex_presence"] <-
```

```

'EVE or EVEX presence (Y Yes, N No, NA Not applicable,P probable, U Unknown)'
traqal$tra_description[traqal$tra_code=="Hva_presence"] <-
'HVA presence (Y Yes, N No, NA Not applicable,P probable, U Unknown)'
# view(traqal)

dbWriteTable(con_diaspara_admin, "tr_traitqualitative_trq_temp",
traqal, overwrite =TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM refeel.tr_traitqualitative_trq");
dbExecute(con_diaspara_admin, "INSERT INTO refeel.tr_traitqualitative_trq(
tra_id,
tra_code,
tra_wkg_code,
tra_spe_code,
tra_description,
tra_typemetric,
tra_qualitativeornumeric)
SELECT
tra_id,
tra_code,
'WGEEL',
'ANG',
tra_description,
tra_typemetric,
tra_qualitativeornumeric
FROM tr_traitqualitative_trq_temp") #5
# TODO add group name

dbExecute(con_diaspara_admin, "INSERT INTO refeel.tr_traitqualitative_trq(
tra_id,
tra_code,
tra_wkg_code,
tra_spe_code,
tra_description,
tra_typemetric,
tra_qualitativeornumeric)
SELECT
29,
'Pigment_stage',
'WGEEL',
'ANG',
'Pigmentation stage according to Elie, 1982',
'Individual',
'Qualitative'")
# duplicated for variables which were both group and

```

```

dbExecute(con_diaspara_admin,
  "UPDATE refeel.tr_traitqualitative_trq SET tra_id = 34
  WHERE tra_code='Hva_presence';")
dbExecute(con_diaspara_admin,
  "UPDATE refeel.tr_traitqualitative_trq SET tra_id = 33
  WHERE tra_code='Evex_presence';")
dbExecute(con_diaspara_admin,
  "UPDATE refeel.tr_traitqualitative_trq SET tra_id= 32
  WHERE tra_code='Anguillicola_presence';")
dbExecute(con_diaspara_admin,
  "UPDATE refeel.tr_traitqualitative_trq SET tra_id= 31
  WHERE tra_code='Is_differentiated'")
dbExecute(con_diaspara_admin,
  "UPDATE refeel.tr_traitqualitative_trq SET tra_id= 30
  WHERE tra_code='Sex';")

```

Ta

tra_id	tra_code	tra_description
29	Pigment_stage	Pigmentation stage according to Elie, 1982
34	Hva_presence	HVA presence (Y Yes, N No, NA Not applicable,P probable, U Unknown)
33	Evex_presence	EVE or EVEX presence (Y Yes, N No, NA Not applicable,P probable, U Unknown)
32	Anguillicola_presence	Presence of Anguillicola (Y Yes, N No, NA Not applicable,P probable, U Unknown)
31	Is_differentiated	Is the eel differentiated (Y Yes, N No, NA Not applicable,P probable, U Unknown)
30	Sex	Sex (F Female,H Hermaphrodite, I Immature attempt made, M Male, T Trans)

1.8 Values of Qualitative fish trait (tr_traitvaluequal_trv)

The qualitative traits have values

SQL code to create tables `ref.tr_traitqualvalue_trv` and `refeel.tr_traitqualvalue_trv`

```

-- DROP TABLE IF EXISTS ref.tr_traitvaluequal_trv CASCADE;

CREATE TABLE ref.tr_traitvaluequal_trv(
  trv_id INTEGER,
  trv_trq_code TEXT NOT NULL,
  CONSTRAINT fk_trv_trq_code
  FOREIGN KEY (trv_trq_code)
  REFERENCES ref.tr_traitqualitative_trq(tra_code)
  ON UPDATE CASCADE ON DELETE CASCADE,
  trv_code text NOT NULL ,
  trv_description text NULL,

```

```

    trv_spe_code TEXT NOT NULL,
    CONSTRAINT fk_trv_spe_code FOREIGN KEY (trv_spe_code)
    REFERENCES ref.tr_species_spe(spe_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    trv_wkg_code TEXT NOT NULL,
    CONSTRAINT fk_trv_wkg_code FOREIGN KEY (trv_wkg_code)
    REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT uk_trv_code UNIQUE (trv_code, trv_trq_code, trv_wkg_code)
);

COMMENT ON COLUMN ref.tr_traitvalueequal_trv.trv_id IS
  'Integer, id of the qualitative used';
COMMENT ON COLUMN ref.tr_traitvalueequal_trv.trv_code IS
  'Code of the qualitative trait';
COMMENT ON COLUMN ref.tr_traitvalueequal_trv.trv_description IS
  'Description of the method';

GRANT ALL ON ref.tr_traitvalueequal_trv TO diaspara_admin;
GRANT SELECT ON ref.tr_traitvalueequal_trv TO diaspara_read;

DROP TABLE IF EXISTS refeel.tr_traitvalueequal_trv CASCADE;
CREATE TABLE refeel.tr_traitvalueequal_trv(
  CONSTRAINT uk_trv_id UNIQUE (trv_id),
  CONSTRAINT fk_trv_trq_code
  FOREIGN KEY (trv_trq_code)
  REFERENCES refeel.tr_traitqualitative_trq(tra_code)
  ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT uk_refeel_trv_code UNIQUE (trv_code, trv_trq_code)
) INHERITS (ref.tr_traitvalueequal_trv);

COMMENT ON COLUMN refeel.tr_traitvalueequal_trv.trv_id IS
  'Integer, id of the qualitative used';
COMMENT ON COLUMN refeel.tr_traitvalueequal_trv.trv_code IS
  'Code of the qualitative trait';
COMMENT ON COLUMN refeel.tr_traitvalueequal_trv.trv_description IS
  'Description of the method';

GRANT ALL ON refeel.tr_traitvalueequal_trv TO diaspara_admin;
GRANT SELECT ON refeel.tr_traitvalueequal_trv TO diaspara_read;

dbGetQuery(con_diaspara, "SELECT * FROM refeel.tr_traitvalueequal_trv;") |>
  knitr::kable() |>

```

Table 1.14: Table of possible qualitative trait values

```

trv <- dbGetQuery(con_diaspara, "SELECT * FROM refeel.tr_traitvaluequal_trv;")
trq <- dbGetQuery(con_diaspara, "SELECT * FROM refeel.tr_traitqualitative_trq;")
#clipr::write_clip(colnames(trv))
trq <- data.frame(
  "trv_id"=1:(13+5*4+7),
  "trv_trq_code"=c(
    rep(trq$tra_code[1],13),
    rep(trq$tra_code[2],5),
    rep(trq$tra_code[3],5),
    rep(trq$tra_code[4],5),
    rep(trq$tra_code[5],5),
    rep(trq$tra_code[6],7)),
  "trv_code"=c(c("VA","VB","VIA0","VIA1","VIA2",
    "VIA3","VIA4","VIB","VII", "mix_VIA1_VIA4",
    "mix_VA_VB", "U","NA"),
    rep(c("N","NA","U","Y","P"),4),
    c("F","I","M","T","U","X","H")),
  "trv_description" = c(
    "No pigmentation except a spot on the caudal fin",
    "Early development of the pigmentation on the skull,
    no superficial piment beyond the cerebral spot.",
    "Development of dorsal pigmentation along the base of dorsal fin",
    "The dosal pigmentation is complete from head to tail",
    "Presence of medio lateral pigmentation but
    it does not reach the beginning of the dorsal fin",
    "The medio-lateral pigmentation reaches the beginning of the dorsal fin",
    "Ventro lateral pigmentation distributed along the myosepta.
    Pigments are still distinct.",
    "Pigments are no longer distinct in the ventro lateral region",
    "Loss of transparency, the abdominal cavity takes a silvery color.
    Generalised development of yellow eel pigment cells",
    "Development of surface and branchiostegal pigmentation",
    "No or early pigmentation",
    "Glass eels, but pigmentation stage is unknown",
    "Not available",
    rep(c("No", "Not Applicable", "Unknown","Yes", "Probable"), 4),
    "Female", "Immature - attempt made but sex could not be destinguished",
    "Male", "Transitional", "Undetermined - no attempt made", "Mixed",
    "Hermaphrodite"
  ))
dbWriteTable(con_diaspara_admin, "tr_traitvaluequal_trv_temp", trq,
  overwrite =TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM refeel.tr_traitvaluequal_trv");
dbExecute(con_diaspara_admin, "INSERT INTO refeel.tr_traitvaluequal_trv
SELECT
trv_id,
trv_trq_code,
trv_code,
trv_description,
'WGEEL',
'ANG'
FROM tr_traitvaluequal_trv_temp;") #40

```

```
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Table 1.15: Values for quality

trv_id	trv_trq_code	trv_code	trv_description
1	Pigment_stage	VA	No pigmentation except a spot on the caudal fin
2	Pigment_stage	VB	Early development of the pigmentation on the skull, n
3	Pigment_stage	VIA0	Development of dorsal pigmentation along the base of
4	Pigment_stage	VIA1	The dosal pigmentation is complete from head to tail
5	Pigment_stage	VIA2	Presence of medio lateral pigmentation but it does no
6	Pigment_stage	VIA3	The medio-lateral pigmentation reaches the beginning
7	Pigment_stage	VIA4	Ventro lateral pigmentation distributed along the myo
8	Pigment_stage	VIB	Pigments are no longer distinct in the ventro lateral r
9	Pigment_stage	VII	Loss of transparency, the abdominal cavity takes a sil
10	Pigment_stage	mix_VIA1_VIA4	Development of surface and branchiostegal pigmentat
11	Pigment_stage	mix_VA_VB	No or early pigmentation
12	Pigment_stage	U	Glass eels, but pigmentation stage is unknown
13	Pigment_stage	NA	Not available
15	Hva_presence	NA	Not Applicable
16	Hva_presence	U	Unknown
17	Hva_presence	Y	Yes
18	Hva_presence	P	Probable
20	Evex_presence	NA	Not Applicable
21	Evex_presence	U	Unknown
22	Evex_presence	Y	Yes
23	Evex_presence	P	Probable
25	Anguillicola_presence	NA	Not Applicable
26	Anguillicola_presence	U	Unknown
27	Anguillicola_presence	Y	Yes
28	Anguillicola_presence	P	Probable
30	Is_differentiated	NA	Not Applicable
31	Is_differentiated	U	Unknown
32	Is_differentiated	Y	Yes
33	Is_differentiated	P	Probable
34	Sex	F	Female
35	Sex	I	Immature - attempt made but sex could not be destin
36	Sex	M	Male
37	Sex	T	Transitional
38	Sex	U	Undetermined - no attempt made
39	Sex	X	Mixed
40	Sex	H	Hermaphrodite
14	Hva_presence	N	No

19	Evex__presence	N	No
29	Is__differentiated	N	No
24	Anguillicola__presence	N	No

1.9 Fish trait measurement method

Currently in the wgeel database, the metrics table value contains one column for value which can store numeric or integer. When it is integer, then, the metric id (mty_id) refers either to a category (e.g sex, or is differentiated), or a method (sexed using size, sexed after gonadal inspection). In the new database, the method will be stored in a different column in the group or individual metrics. This table references the methods proposed. Again these methods will be working group specific so we create an inherited table.

SQL code to create tables `ref.tr_traitmethod_trm` and `refeel.tr_traitmethod_trm`

```
-- DROP TABLE ref.tr_traitmethod_trm;

CREATE TABLE ref.tr_traitmethod_trm (
  trm_id integer PRIMARY KEY,
  trm_code text NOT NULL,
  trm_wkg_code TEXT NOT NULL,
  CONSTRAINT fk_trm_wkg_code FOREIGN KEY (trm_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  trm_spe_code TEXT NOT NULL,
  CONSTRAINT fk_trm_spe_code FOREIGN KEY (trm_spe_code)
  REFERENCES ref.tr_species_spe(spe_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  trm_description text NULL,
  CONSTRAINT uk_trm_code UNIQUE (trm_code)
);

COMMENT ON TABLE ref.tr_traitmethod_trm IS
'Table of method used to obtain a trait metric';
COMMENT ON COLUMN ref.tr_traitmethod_trm.trm_id IS
'Integer, id of the method used';
COMMENT ON COLUMN ref.tr_traitmethod_trm.trm_code IS
'Name of the method used';
COMMENT ON COLUMN ref.tr_traitmethod_trm.trm_wkg_code IS
'Working group code';
COMMENT ON COLUMN ref.tr_traitmethod_trm.trm_spe_code IS
'Species code';
COMMENT ON COLUMN ref.tr_traitmethod_trm.trm_description IS
```

```

'Description of the method';

GRANT ALL ON ref.tr_traitmethod_trm TO diaspara_admin;
GRANT SELECT ON ref.tr_traitmethod_trm TO diaspara_read;

DROP TABLE IF EXISTS refeel.tr_traitmethod_trm;
CREATE TABLE refeel.tr_traitmethod_trm (
  CONSTRAINT uk_refeel_tm_id UNIQUE (trm_id),
  CONSTRAINT uk_refeel_tm_code UNIQUE (trm_code),
  CONSTRAINT fk_trm_wkg_code FOREIGN KEY (trm_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_trm_spe_code FOREIGN KEY (trm_spe_code)
  REFERENCES ref.tr_species_spe(spe_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT uk_trm_code UNIQUE (trm_code)
) INHERITS (ref.tr_traitmethod_trm);

COMMENT ON TABLE refeel.tr_traitmethod_trm IS
'Table of method used to obtain a trait metric';
COMMENT ON COLUMN refeel.tr_traitmethod_trm.trm_id IS
'Integer, id of the method used';
COMMENT ON COLUMN refeel.tr_traitmethod_trm.trm_code IS
'Name of the method used';
COMMENT ON COLUMN refeel.tr_traitmethod_trm.trm_wkg_code IS
'Working group code';
COMMENT ON COLUMN refeel.tr_traitmethod_trm.trm_spe_code IS
'Species code';
COMMENT ON COLUMN refeel.tr_traitmethod_trm.trm_description IS
'Description of the method';

GRANT ALL ON refeel.tr_traitmethod_trm TO diaspara_admin;
GRANT SELECT ON refeel.tr_traitmethod_trm TO diaspara_read;

tra <- dbGetQuery(con_wgeel_local, "SELECT * FROM ref.tr_metrictype_mty ;")
tra <- tra[grepl("method", tra$mtty_name), ]
res <- dbGetQuery(con_diaspara, "SELECT * FROM ref.tr_traitmethod_trm")
clipr::write_clip(colnames(res))
traitmethod <- data.frame(
  "trm_id" = 1:6,

```

```

"trm_tra_code" = c(
  "Gonadal_inspection",
  "Length_based_sex",
  "Anguillicola_stereomicroscope_count",
  "Anguillicola_visual_count",
  "Muscle_lipid_fatmeter",
  "Muscle_lipid_gravimeter"
),
"trm_wkg_code" = rep("WGEEL", 6),
"trm_spe_code" = rep("ANG", 6),
"trm_description" = c(
  "The eel is dissected and the gonads are inspected,
  In males, the testes appear as thin, ribbon-like, whitish structures.
  In females, the ovaries are larger, lobed, and more granular,
  often yellowish or pinkish depending on maturity.
  Many eels pass through an intersexual phase
  (Here referred as Mixed to align with ICES Vocab),
  where gonads show both ovarian and testicular tissue.
  This is part of their natural development", # nolint
  "The size at silvering of eels depends on the sex.
  Males are found within the 25-45 cm range, and females are found over 45 cm.
  There is an overlap of sexes around ",# nolint
  "Anguillicola count using a dissecting microscope,
  this allows to detect early-stage infections or small larvae",
  "Anguillicola visual count",
  "Non invasive estimation of the fat content
  by measuring the dielectric properties of tissues,
  this method requires a calibration.",# nolint
  "Gravimeter, muscle tissues are dried and the lipids
  are extracted using solvents, the extracted fat is dried and weighted" # nolint
)
)

dbWriteTable(con_diaspara_admin, "tr_traitmethod_trm_temp", traitmethod, overwrite = TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM refeel.tr_traitmethod_trm")
dbExecute(con_diaspara_admin, "INSERT INTO refeel.tr_traitmethod_trm
SELECT
trm_id,
trm_tra_code,
trm_wkg_code,
trm_spe_code,
trm_description
FROM tr_traitmethod_trm_temp") # 6

```

```
trm <- dbGetQuery(con_diaspara, "SELECT * FROM ref.tr_traitmethod_trm;")
knitr::kable(trm)|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

trm_id	trm_code	trm_wkg_code	trm_spe_code	trm_description
1	Gonadal_inspection	WGEEL	126281	The eel is dissected and
2	Length_based_sex	WGEEL	126281	The size at silvering of
3	Anguillicola_stereomicroscope_count	WGEEL	126281	Anguillicola count using
5	Muscle_lipid_fatmeter	WGEEL	126281	Non invasive estimation
6	Muscle_lipid_gravimeter	WGEEL	126281	Gravimeter, muscle tissue
4	Anguillicola_visual_count	WGEEL	126281	Anguillicola visual count

1.10 Creating the station table

Series will be linked to one or several stations.

The station are loaded into a table that will not be exactly like ICES's since I cannot load the full tables, and don't want to replicate the full complexity of stations. For instance currently station 1000 has several parents for "Station_DTYPE" which I aggregate like EW_{EU}~NU~CW, in my table. But the vocab is here.

SQL code to create tables

```
/*
 *
 * Note since the api retrieves a one to many relationship
 * in the foreign tables I need to drop foreign keys
 */

DROP TABLE IF EXISTS "ref"."StationDictionary" CASCADE;
CREATE TABLE "ref"."StationDictionary"(
  "Definition" TEXT NOT NULL DEFAULT 'Station',
  "HeaderRecord" TEXT NOT NULL DEFAULT 'Record',
  "Station_Code" INTEGER PRIMARY KEY,
  "Station_Country" TEXT NOT NULL,
  --CONSTRAINT fk_station_country
  --FOREIGN KEY ("Station_Country")
  --REFERENCES ref.tr_country_cou(cou_code)
  --ON UPDATE CASCADE ON DELETE CASCADE, -- possible problem here AS I don't have ALL the countries
  "Station_Name" TEXT NOT NULL,
  "Station_LongName" TEXT NULL,
```

```

"Station_ActiveFromDate" CHARACTER VARYING (10) NULL, -- CHANGE I don't have that FROM the A
"Station_ActiveUntilDate" CHARACTER VARYING (10) NULL,
"Station_ProgramGovernance" TEXT NULL,
--CONSTRAINT fk_station_programgovernance
--FOREIGN KEY ("Station_ProgramGovernance")
--REFERENCES "ref"."PRGOV"("Key")
--ON UPDATE CASCADE ON DELETE CASCADE,
"Station_StationGovernance" TEXT NULL,
--CONSTRAINT fk_station_stationgovernance
--FOREIGN KEY ("Station_StationGovernance")
--REFERENCES "ref"."EDMO"("Key")
--ON UPDATE CASCADE ON DELETE CASCADE,
"Station_PURPM" TEXT NULL, -- FAILS WITH NOT NULL
--CONSTRAINT fk_station
--FOREIGN KEY ("Station_PURPM")
--REFERENCES "ref"."PURPM"("Key")
--ON UPDATE CASCADE ON DELETE CASCADE,
"Station_Latitude" NUMERIC NULL,
"Station_LatitudeRange" NUMERIC NULL,
"Station_Longitude" NUMERIC NULL,
"Station_LongitudeRange" NUMERIC NULL,
"Station_Geometry" geometry,
--CONSTRAINT ck_geom_or_latlon
--CHECK ("Station_Geometry" IS NOT NULL
--OR "Station_Latitude" IS NOT NULL
--OR "Station_LatitudeRange" IS NOT NULL),
"Station_DataType" TEXT NULL,
--CONSTRAINT fk_station_datatype
--FOREIGN KEY ("Station_DataType") REFERENCES
--"ref"."Station_DTYPE"("Key")
--ON UPDATE CASCADE ON DELETE CASCADE,
"Station_WLTYP" TEXT,
--CONSTRAINT fk_station_wltype
--FOREIGN KEY ("Station_WLTYP")
--REFERENCES "ref"."WLTYP"("Key")
--ON UPDATE CASCADE ON DELETE CASCADE,
"Station_MSTAT" TEXT,
-- CONSTRAINT fk_station_msat
--FOREIGN KEY ("Station_MSTAT")
--REFERENCES "ref"."MSTAT"("Key")
--ON UPDATE CASCADE ON DELETE CASCADE,
"Station_Notes" TEXT,
"Station_Deprecated" TEXT,
FOREIGN KEY ("Station_Deprecated")
REFERENCES "ref"."Deprecated"("Key")

```

```

ON UPDATE CASCADE ON DELETE CASCADE
);

-- Not sure I'll ever need that one .....
DROP TABLE IF EXISTS "ref"."Relation";
CREATE TABLE "ref"."Relation"(
  "Definiton" TEXT NOT NULL DEFAULT 'Relation',
  "HeaderRecord" TEXT NOT NULL DEFAULT 'Record',
  "Relation_Code" INTEGER NOT NULL,
  CONSTRAINT fk_relation_code FOREIGN KEY ("Relation_Code")
  REFERENCES "ref"."StationDictionary"("Station_Code")
  ON UPDATE CASCADE ON DELETE CASCADE,
  "Relation_Country" TEXT NOT NULL,
  CONSTRAINT fk_relation_country
  FOREIGN KEY ("Relation_Country")
  REFERENCES ref.tr_country_cou(cou_code)
  ON UPDATE CASCADE ON DELETE CASCADE,
  "Relation_Name" CHARACTER VARYING(50) NOT NULL,
  "Relation_ActiveFromDate" CHARACTER VARYING(10) NOT NULL,
  "Relation_RelatedCode" INTEGER NOT NULL,
  CONSTRAINT fk_relation_relatedCode
  FOREIGN KEY ("Relation_RelatedCode")
  REFERENCES "ref"."StationDictionary"("Station_Code")
  ON UPDATE CASCADE ON DELETE CASCADE,
  "Relation_RelatedCountry" TEXT NOT NULL,
  CONSTRAINT fk_relation_related_country
  FOREIGN KEY ("Relation_RelatedCountry")
  REFERENCES ref.tr_country_cou(cou_code)
  ON UPDATE CASCADE ON DELETE CASCADE,
  "Relation_RelatedName" CHARACTER VARYING(50) NOT NULL,
  "Relation_RelatedActiveFromDate" CHARACTER VARYING (10) NOT NULL,
  "Relation_RelationType" TEXT NOT NULL);

COMMENT ON TABLE ref."WLTP" IS 'Water and land station type';
COMMENT ON TABLE ref."PRGOV" IS 'Program Governance referential';
COMMENT ON TABLE ref."PURPM" IS 'Purpose of monitoring';
COMMENT ON TABLE ref."MSTAT" IS 'Purpose of monitoring';
COMMENT ON TABLE ref."EDMO" IS 'European Directory of Marine Organisations (EDMO)';

```

We have to do some fixes see (issue 27)[<https://github.com/ices-tools-prod/icesVocab/issues/27>] and remove nearly all the constraints, including the length of 50

for stationNames. The following chunk creates a vocab with 14086 stations.

```
# tested 28/05/2025

library(icesStation)
system.time(
station <- getListStation()
# user      system      spent
#   268.09      9.06    1402.47
save(station, file = "data/station.Rdata")
# load(file = "data/station.Rdata")
initcap <- function(X) paste0(substring(X,1,1),tolower(substring(X,2, length(X))))

station <- station[station$Station_Name != 'TestBulkUpload1',]
station$Station_Deprecated<- initcap(as.character(station$Station_Deprecated))

dbWriteTable(con_diaspara_admin, "temp_station", station, overwrite = TRUE)
dbExecute(con_diaspara_admin,'DELETE FROM "ref"."StationDictionary"')
dbExecute(con_diaspara_admin,'INSERT INTO "ref"."StationDictionary"
("Definition",
"HeaderRecord",
"Station_Code",
"Station_Country",
"Station_Name",
"Station_LongName",
"Station_ActiveFromDate",
"Station_ActiveUntilDate",
"Station_ProgramGovernance",
"Station_StationGovernance",
"Station_PURPM",
"Station_DataType",
"Station_WLTYP",
"Station_MSTAT",
"Station_Notes",
"Station_Deprecated")
SELECT
"Definition",
"HeaderRecord",
"Station_Code"::INTEGER,
"Station_Country",
"Station_Name",
"Station_LongName",
"Station_ActiveFromDate",
"Station_ActiveUntilDate",
"Station_ProgramGovernance",
```

```

"Station_StationGovernance",
"Station_PURPM",
"Station_DataType",
"Station_WLTYP",
"Station_MSTAT",
"Station_Notes",
"Station_Deprecated"
FROM temp_station') #14086
dbExecute(con_diaspara_admin, "DROP TABLE if exists temp_station")

# tested 28/05/2025

library(icesStation)
system.time(
station <- getListStation())
#      user      system      spent
# 268.09      9.06    1402.47
save(station, file = "data/station.Rdata")
# load(file = "data/station.Rdata")
initcap <- function(X) paste0(substring(X,1,1),tolower(substring(X,2, length(X))))

station <- station[station$Station_Name != 'TestBulkUload1',]
station$Station_Deprecated<- initcap(as.character(station$Station_Deprecated))

dbWriteTable(con_diaspara_admin, "temp_station", station, overwrite = TRUE)
dbExecute(con_diaspara_admin, 'DELETE FROM "ref"."StationDictionary"')
dbExecute(con_diaspara_admin, 'INSERT INTO "ref"."StationDictionary"
("Definition",
"HeaderRecord",
"Station_Code",
"Station_Country",
"Station_Name",
"Station_LongName",
"Station_ActiveFromDate",
"Station_ActiveUntilDate",
"Station_ProgramGovernance",
"Station_StationGovernance",
"Station_PURPM",
"Station_DataType",
"Station_WLTYP",
"Station_MSTAT",
"Station_Notes",
"Station_Deprecated")
SELECT

```

```

"Definition",
"HeaderRecord",
"Station_Code"::INTEGER,
  "Station_Country",
  "Station_Name",
  "Station_LongName",
"Station_ActiveFromDate",
"Station_ActiveUntilDate",
  "Station_ProgramGovernance",
  "Station_StationGovernance",
"Station_PURPM",
  "Station_DataType",
  "Station_WLTYP",
  "Station_MSTAT",
"Station_Notes",
"Station_Deprecated"
FROM temp_station') #14086
dbExecute(con_diaspara_admin, "DROP TABLE if exists temp_station")

```

! Comment on Station import

Even though a table station exists in WGEEL, it really contains no actual code from ICES. To insert data into stations, the data providers will have to check that they have an EDMO code, and then we will have to bulk load stations.

1.11 Creating the version table refeel.tr_version_ver

SQL code to create table `refeel.tr_version_ver`

```

DROP TABLE IF EXISTS refeel.tr_version_ver CASCADE;
CREATE TABLE refeel.tr_version_ver(
  CONSTRAINT ver_code_pkey PRIMARY KEY (ver_code),
  CONSTRAINT fk_ver_spe_code FOREIGN KEY (ver_spe_code)
REFERENCES ref.tr_species_spe(spe_code)
ON UPDATE CASCADE ON DELETE CASCADE,
) inherits (ref.tr_version_ver);

COMMENT ON TABLE refeel.tr_version_ver
IS 'Table of data or variable version, essentially one datacall or advice, inherits ref.tr_v
ALTER TABLE refeel.tr_version_ver ALTER COLUMN ver_wkg_code
SET DEFAULT 'WGEEL';

```

```

COMMENT ON TABLE refeel.tr_version_ver
IS 'Table of data or variable version, essentially one datacall or advice, inherits ref.tr_v

COMMENT ON COLUMN refeel.tr_version_ver.ver_version
IS 'Version code, wkg-year-version.';
COMMENT ON COLUMN refeel.tr_version_ver.ver_year
IS 'Year of assesement.';
COMMENT ON COLUMN refeel.tr_version_ver.ver_spe_code
IS 'Species code e.g. ''127186'' Anguilla anguilla references tr_species_spe.';
COMMENT ON COLUMN refeel.tr_version_ver.ver_wkg_code
IS 'Code of the working group, WGEEL';
COMMENT ON COLUMN refeel.tr_version_ver.ver_stockkeylabel
IS 'Ver_stockkeylabel e.g. ele.2737.nea.';
COMMENT ON COLUMN refeel.tr_version_ver.ver_datacalldoi
IS 'Data call DOI, find a way to retrieve that information
and update this comment';
COMMENT ON COLUMN refeel.tr_version_ver.ver_version
IS 'Version code in original database, eg 2,4 for wgnas, dc_2020 for wgeel.';
COMMENT ON COLUMN refeel.tr_version_ver.ver_description
IS 'Description of the data call / version.';
GRANT ALL ON refeel.tr_version_ver TO diaspara_admin;
GRANT SELECT ON refeel.tr_version_ver TO diaspara_read;

```

This table is the same as in wgnas, it is inherited from ref.

```

# get the latest version from the server
ver <- dbGetQuery(con_wgeel_distant, "select * from ref.tr_datasource_dts")
save(ver, file= "data/tr_datasource_dts.Rdata")
ver <- ver[ver$dts_datasource != 'test',]
dc <- ver[grepl("dc", ver$dts_datasource), "dts_datasource"]
dcyear <- as.integer(lapply(strsplit(dc,"_"), function(X)X[2]))
wgeel <- ver[grepl("wgeel", ver$dts_datasource), "dts_datasource"]
wgeelyear <- as.integer(lapply(strsplit(wgeel,"_"), function(X)X[2]))
tr_version_ver <- data.frame(
  ver_code = paste0(rep("ANG-", 11),
    c(wgeelyear, dcyear, 2025),c("-1","-2",rep("-1", 9))),
  ver_year = c(wgeelyear, dcyear, 2025,2025),
  ver_spe_code = "ANG",
  ver_datacalldoi=c(rep(NA, 10),
    "https://doi.org/10.17895/ices.pub.25816738.v2",
    "https://doi.org/10.17895/ices.pub.25816738.v2",
    "https://doi.org/10.17895/ices.pub.29254589"),
  ver_stockkeylabel =c("ele.2737.nea"),
  ver_version=c(1,2,rep(1,19),2),
  ver_description=ver$dts_description) #

```

```
DBI::dbWriteTable(con_diaspara_admin, "temp_tr_version_ver", tr_version_ver,
overwrite = TRUE)
dbExecute(con_diaspara_admin,
"INSERT INTO refeel.tr_version_ver(ver_code,ver_year,ver_spe_code,
ver_datacalldoi,ver_stockkeylabel,ver_version, ver_description)
SELECT ver_code,ver_year,ver_spe_code, ver_datacalldoi,
ver_stockkeylabel,ver_version, ver_descriptio FROM temp_tr_version_ver;") # 5
DBI::dbExecute(con_diaspara_admin, "DROP TABLE temp_tr_version_ver;")
```

ver_code	ver_year	ver_spe_code	ver_stockkeylabel	ver_datacalldoi
WGEEL-2023-1	2023	Anguilla anguilla	ele	
WGEEL-2024-1	2024	Anguilla anguilla	ele	https://doi.org/10.17895/ices.pub.258
WGEEL-2025-1	2025	Anguilla anguilla	ele	https://doi.org/10.17895/ices.pub.258
WGEEL-2025-2	2025	Anguilla anguilla	ele	https://doi.org/10.17895/ices.pub.292
WGEEL-2016-1	2016	Anguilla anguilla	ele	
WGEEL-2017-2	2017	Anguilla anguilla	ele	
WGEEL-2017-1	2017	Anguilla anguilla	ele	
WGEEL-2018-1	2018	Anguilla anguilla	ele	
WGEEL-2019-1	2019	Anguilla anguilla	ele	
WGEEL-2020-1	2020	Anguilla anguilla	ele	
WGEEL-2021-1	2021	Anguilla anguilla	ele	
WGEEL-2022-1	2022	Anguilla anguilla	ele	

1.12 series table t_series_ser

1.12.1 Creating series main table for series

SQL code to create table dat.t_series_ser

```
--DROP TABLE IF EXISTS dat.t_series_ser CASCADE;

CREATE TABLE dat.t_series_ser (
  ser_id uuid PRIMARY KEY,
  ser_code text NOT NULL,
  CONSTRAINT uk_ser_code UNIQUE (ser_code),
  ser_name TEXT NOT NULL,
  --CONSTRAINT uk_ser_name UNIQUE (ser_name),
  ser_spe_code TEXT NULL,
  CONSTRAINT fk_ser_spe_code FOREIGN KEY (ser_spe_code)
REFERENCES "ref".tr_species_spe(spe_code)
ON UPDATE CASCADE ON DELETE RESTRICT ,
```

```

ser_lfs_code TEXT NULL,
CONSTRAINT fk_ser_lfs_code_ser_spe_code FOREIGN KEY (ser_lfs_code, ser_spe_code)
REFERENCES "ref".tr_lifestage_lfs (lfs_code, lfs_spe_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
ser_are_code TEXT NOT NULL,
CONSTRAINT fk_ser_are_code FOREIGN KEY (ser_are_code)
REFERENCES "ref".tr_area_are (are_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
ser_wkg_code TEXT NOT NULL,
CONSTRAINT fk_ser_wkg_code FOREIGN KEY (ser_wkg_code)
REFERENCES "ref".tr_icworkinggroup_wkg(wkg_code),
ser_ver_code TEXT NOT NULL,
CONSTRAINT fk_ser_ver_code FOREIGN KEY (ser_ver_code)
REFERENCES ref.tr_version_ver(ver_code),
ser_cou_code TEXT NOT NULL,
CONSTRAINT fk_ser_cou_code FOREIGN KEY (ser_cou_code)
REFERENCES ref.tr_country_cou(cou_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_wltyp_code FOREIGN KEY( ser_wltyp_code)
REFERENCES ref."WLTP"("Key")
ON UPDATE CASCADE ON DELETE RESTRICT,
ser_hab_code TEXT NULL,
CONSTRAINT fk_ser_habitat_code FOREIGN KEY(ser_hab_code)
REFERENCES ref.tr_habitat_hab (hab_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
ser_gea_code TEXT NULL,
CONSTRAINT fk_ser_gea_code FOREIGN KEY (ser_gea_code)
REFERENCES ref.tr_gear_gea(gea_code)
ON UPDATE CASCADE ON DELETE CASCADE,
ser_fiw_code TEXT,
CONSTRAINT fk_ser_fiw_code FOREIGN KEY (ser_fiw_code)
REFERENCES ref.tr_fishway_fiw(fiw_code)
ON UPDATE CASCADE ON DELETE CASCADE,
ser_mon_code TEXT,
CONSTRAINT fk_ser_mon_code FOREIGN KEY (ser_mon_code)
REFERENCES ref.tr_monitoring_mon(mon_code)
ON UPDATE CASCADE ON DELETE CASCADE,
ser_uni_code varchar(20),
CONSTRAINT fk_ser_uni_code FOREIGN KEY (ser_uni_code)
REFERENCES ref.tr_units_uni(uni_code)
ON DELETE CASCADE ON UPDATE CASCADE,
ser_effort_uni_code varchar(20),
CONSTRAINT fk_ser_effort_uni_code FOREIGN KEY (ser_effort_uni_code)
REFERENCES ref.tr_units_uni(uni_code)
ON DELETE CASCADE ON UPDATE CASCADE,

```

```

ser_description TEXT NULL,
ser_locationdescription TEXT NULL,
ser_wltyp_code TEXT NULL,
ser_stocking boolean NULL,
ser_stockingcomment TEXT NULL,
ser_protocol TEXT NULL,
ser_samplingstrategy TEXT NULL,
ser_datarightsholder TEXT NULL,
ser_datelastupdate DATE NOT NULL,
geom geometry NULL);

COMMENT ON TABLE dat.t_series_ser IS
'Table of time series, or sampling data identifier. This corresponds to a multi-annual data
It can correspond to time series data or individual metrics collection or both. This table i
the content of the tables in refeel, refnas, refbast... Note this table is joined to Stati
station_activefromdate, stationactiveuntildate, latitude, latituderange, longitude, longituo
COMMENT ON COLUMN dat.t_series_ser.ser_id IS
'UUID, identifier of the series, primary key';
COMMENT ON COLUMN dat.t_series_ser.ser_code IS
'Code of the series';
COMMENT ON COLUMN dat.t_series_ser.ser_name IS
'Name of the series';
COMMENT ON COLUMN dat.t_series_ser.ser_spe_code IS
'Species, e.g. ''127186'' ref.tr_species_spe, the species can be null but
it should correspond to the main species target by the sampling';
COMMENT ON COLUMN dat.t_series_ser.ser_lfs_code IS
'Life stage see tr_lifestage_lfs, Code of the lifestage see tr_lifestage_lfs, the constraint
both lfs_code, and lfs_spe_code (as two species can have the same lifestage code. The lifest
COMMENT ON COLUMN dat.t_series_ser.ser_wkg_code IS
'Code of the working group, one of WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dat.t_series_ser.ser_ver_code IS
'Version code referencing tr_version_ver the data call e.g. NAS_2025dc_2020, wgeel_2016, wk
COMMENT ON COLUMN dat.t_series_ser.ser_cou_code IS
'Code of the country';
COMMENT ON COLUMN dat.t_series_ser.ser_wltyp_code IS
'Code of the habitat type, one of MO (marine open), MC (Marine coastal), T (Transitional wat
COMMENT ON COLUMN dat.t_series_ser.ser_hab_code IS
'Code of the habitat, see tr_habitat_hab';
COMMENT ON COLUMN dat.t_series_ser.ser_are_code IS
'Code of the area, areas are geographical sector most often corresponding to stock units, s
COMMENT ON COLUMN dat.t_series_ser.ser_uni_code IS
'Unit for the value reported in annual table, if the series reports annual data in Kg or Num
COMMENT ON COLUMN dat.t_series_ser.ser_effort_uni_code IS
'Annual data collection effort unit code, used for the effort column in annual data';
COMMENT ON COLUMN dat.t_series_ser.ser_description IS

```

```

'Concise description of the series. Should include species, stage targeted, location and gear
COMMENT ON COLUMN dat.t_series_ser.ser_locationdescription IS
    'This should provide a description of the site, e.g. if ist far inland, in the middle of a
    (e.g. "Bresle river trap 3 km from the sea" or IYFS/IBTS sampling in the Skagerrak-Kattegat
COMMENT ON COLUMN dat.t_series_ser.ser_gea_code IS
    'Code of the gear used, see tr_gear_gea';
COMMENT ON COLUMN dat.t_series_ser.ser_fiw_code IS
    'Code the fishway, eg P0 for pool type fishway';
COMMENT ON COLUMN dat.t_series_ser.ser_mon_code IS
    'Code the Monitoring device, eg S0 for Sonar';
COMMENT ON COLUMN dat.t_series_ser.ser_stocking IS
    'Boolean, Is there restocking (for eel) or artifical reproduction in the river / basin, af
COMMENT ON COLUMN dat.t_series_ser.ser_stockingcomment IS
    'Comment on stocking';
COMMENT ON COLUMN dat.t_series_ser.ser_protocol IS
    'Describe sampling protocol';
COMMENT ON COLUMN dat.t_series_ser.ser_samplingstrategy IS
    'Describe sampling strategy';
COMMENT ON COLUMN dat.t_series_ser.ser_datarightsholder IS
    'Code of the data rights holder of the series, this field will be used in DATSU to acknowl
COMMENT ON COLUMN dat.t_series_ser.ser_datelastupdate IS
    'Last modification in the series, from a trigger';
COMMENT ON COLUMN dat.t_series_ser.geom IS
    'Series geometry column EPSG 4326, can be more detailed than the geometry for station';

GRANT ALL ON dat.t_series_ser TO diaspara_admin;
GRANT SELECT ON dat.t_series_ser TO diaspara_read;

```

This table is inherited so the table created in dat will be empty, and will only receive data by inheritance. It can be considered as a view for ICES (no inheritance in SQL server).

Since this table is inherited the wkg needs to be included. One table will be created per working group. These tables will be collated together. Most details information about metadata will be in the metadata tables, either to describe annual sampling or to describe individual metric collection.

Monitoring stations (including fixed stations) that are used for recurring sampling or data collection are managed via the Station Code Request Application in ICES (ICES 2024). This table will reference monitoring station but it might be NULL, as some sampling designs or data collection, for instance for the DCF, are not related to a station.

QUESTION ICES (Cédric via Teams meeting)

The ser_id might in some cases correspond to a fixed station but not only. For this reason the station is referenced here but another code will be created to reference the sampling collection. Is this OK ?

ANSWER ICES : Maria (20/05/2025)

Yes this makes sense, I would advise to use both the stations and a unique identifier at the top. Using UUID will help unicity there. It's not clear if it needs to be a vocabulary. There is a procedure to integrate stations, but for this we will probably do a bulk import.

i QUESTION to DIASPARA

Is this that many of our series, and many of BAST or NAS would be related to fishways. Shouldn't we need a fishway / fishway monitoring device vocab there ?

i QUESTION to WKEELDATA

Currently everything is in the same table, but it could be split by period following exactly the station work. A description would then spans a limited amount of year from date_begin to date_end. If the series description still hold the date_end is null. If so we need to decide what fields could change over time (if any)

1.12.2 Creating inherited series

SQL code to create table dateel.t_series_ser

```
-- when dumping this is inherited => I have a not null constraint on the server
-- when the table is recreated
```

```
ALTER TABLE dat.t_series_ser ALTER COLUMN ser_are_code DROP NOT NULL;
ALTER TABLE dat.t_series_ser ALTER COLUMN ser_cou_code DROP NOT NULL;
```

```
DROP TABLE IF EXISTS dateel.t_series_ser;
CREATE TABLE dateel.t_series_ser (
    CONSTRAINT pk_ser_id PRIMARY KEY (ser_id),
    CONSTRAINT uk_ser_code UNIQUE (ser_code),
    CONSTRAINT uk_ser_name UNIQUE (ser_name),
    CONSTRAINT fk_ser_spe_code FOREIGN KEY (ser_spe_code)
        REFERENCES ref.tr_species_spe(spe_code)
        ON UPDATE CASCADE ON DELETE RESTRICT ,
    CONSTRAINT fk_ser_lfs_code_ser_spe_code
        FOREIGN KEY (ser_lfs_code, ser_spe_code)
```

```

REFERENCES ref.tr_lifestage_lfs (lfs_code, lfs_spe_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_are_code FOREIGN KEY (ser_are_code)
REFERENCES refeel.tr_area_are (are_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_wkg_code FOREIGN KEY (ser_wkg_code)
REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_ver_code FOREIGN KEY (ser_ver_code)
REFERENCES refeel.tr_version_ver(ver_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_cou_code FOREIGN KEY (ser_cou_code)
REFERENCES ref.tr_country_cou(cou_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_uni_code FOREIGN KEY (ser_uni_code)
REFERENCES ref.tr_units_uni(uni_code)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_ser_effort_uni_code
FOREIGN KEY (ser_effort_uni_code)
REFERENCES ref.tr_units_uni(uni_code)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_ser_wltyp_code FOREIGN KEY( ser_wltyp_code)
REFERENCES ref."WLTP"("Key")
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_gea_code FOREIGN KEY (ser_gea_code)
REFERENCES ref.tr_gear_gea(gea_code)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT fk_ser_fiw_code FOREIGN KEY (ser_fiw_code)
REFERENCES ref.tr_fishway_fiw(fiw_code)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT fk_ser_mon_code FOREIGN KEY (ser_mon_code)
REFERENCES ref.tr_monitoring_mon(mon_code)
ON UPDATE CASCADE ON DELETE CASCADE
) inherits (dat.t_series_ser);

-- In the wgeel schema the default is WGEEL
ALTER TABLE dateel.t_series_ser ALTER COLUMN ser_wkg_code SET DEFAULT 'WGEEL';

COMMENT ON TABLE dateel.t_series_ser IS
'Table of time series, or sampling data identifier. This corresponds to a multi-annual data
It can correspond to time series data or individual metrics collection or both. This table i
COMMENT ON COLUMN dat.t_series_ser.ser_id IS
'UUID, identifier of the series, primary key';
COMMENT ON COLUMN dat.t_series_ser.ser_code IS
'Code of the series, short name of the recruitment series, this must be 4 letters + stage na

```

```

COMMENT ON COLUMN dat.t_series_ser.ser_name IS
    'Name of the series';
COMMENT ON COLUMN dat.t_series_ser.ser_spe_code IS
    'Species, e.g. ''127186''... references ref.tr_species_spe, the species can be null but
    it should correspond to the main species target by the sampling';
COMMENT ON COLUMN dat.t_series_ser.ser_lfs_code IS
    'Life stage see tr_lifestage_lfs, Code of the lifestage see tr_lifestage_lfs, the constraint
    both lfs_code, and lfs_spe_code (as two species can have the same lifestage code. The lifestage
    code is unique for each species);
COMMENT ON COLUMN dat.t_series_ser.ser_wkg_code IS
    'Code of the working group, one of WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dat.t_series_ser.ser_ver_code IS
    'Version code from ref.tr_version_ver the data call e.g. WGNAS_2020_1, WGEEL_2016_1';
COMMENT ON COLUMN dat.t_series_ser.ser_cou_code IS
    'Code of the country';
COMMENT ON COLUMN dat.t_series_ser.ser_wltyp_code IS
    'Code of the habitat type, one of MO (marine open), MC (Marine coastal), T (Transitional water);
COMMENT ON COLUMN dat.t_series_ser.ser_hab_code IS
    'Code of the habitat, see tr_habitat_hab';
COMMENT ON COLUMN dat.t_series_ser.ser_are_code IS
    'Code of the area, areas are geographical sector most often corresponding to stock units, s
    ector;
COMMENT ON COLUMN dat.t_series_ser.ser_uni_code IS
    'Annual value Unit, references table tr_unit_uni.';
COMMENT ON COLUMN dat.t_series_ser.ser_effort_uni_code IS
    'Annual data collection effort unit code, references table tr_unit_uni.';
COMMENT ON COLUMN dat.t_series_ser.ser_description IS
    'Concise description of the series. Should include species, stage targeted, location and g
    eographical sector;
COMMENT ON COLUMN dat.t_series_ser.ser_locationdescription IS
    'This should provide a description of the site, e.g. if ist far inland, in the middle of a
    river (e.g. `Bresle river trap 3 km from the sea` or `IYFS/IBTS sampling in the Skagerrak-Kattegat
    strait');
COMMENT ON COLUMN dat.t_series_ser.ser_gea_code IS
    'Code of the gear used, see tr_gear_gea, if a fishway or scientific monitoring device leave
    the river;
COMMENT ON COLUMN dat.t_series_ser.ser_fiw_code IS
    'Code the fishway, eg P0 for pool type fishway';
COMMENT ON COLUMN dat.t_series_ser.ser_mon_code IS
    'Code the Monitoring device, eg S0 for Sonar';
COMMENT ON COLUMN dat.t_series_ser.ser_stocking IS
    'Boolean, Is there restocking (for eel) or artificial reproduction in the river / basin, af
    ter the first capture;
COMMENT ON COLUMN dat.t_series_ser.ser_stockingcomment IS
    'Comment on stocking';
COMMENT ON COLUMN dat.t_series_ser.ser_protocol IS
    'Describe sampling protocol';
COMMENT ON COLUMN dat.t_series_ser.ser_samplingstrategy IS
    'Describe sampling strategy';
COMMENT ON COLUMN dat.t_series_ser.ser_datarightsholder IS
    'Code of the data rights holder of the series, this field will be used in DATSU to acknowle
    dge the rights holder;

```

```

COMMENT ON COLUMN dat.t_series_ser.ser_datelastupdate IS
  'Last modification in the series, from a trigger';
COMMENT ON COLUMN dat.t_series_ser.geom IS
  'Series geometry column EPSG 4326, can be more detailed than the geometry for station';

GRANT ALL ON dateel.t_series_ser TO diaspara_admin;
GRANT SELECT ON dateel.t_series_ser TO diaspara_read;

-- pb with DUMP
ALTER TABLE dateel.t_series_ser ALTER COLUMN

```

1.12.3 Import t_series_ser from wgeel

```

res <- dbGetQuery(con_diaspara, "SELECT * FROM dateel.t_series_ser")
clipr::write_clip(colnames(res))

# temporarily remove area constraint

dbExecute(con_diaspara_admin, "ALTER TABLE dateel.t_series_ser drop constraint
fk_ser_are_code")
dbExecute(con_diaspara_admin, "ALTER TABLE dateel.t_series_ser ALTER COLUMN
ser_are_code DROP NOT NULL")
dbExecute(con_diaspara_admin, "ALTER TABLE dateel.t_series_ser ALTER COLUMN
ser_cou_code DROP NOT NULL")
dbExecute(con_diaspara_admin,
"ALTER TABLE dateel.t_series_ser DROP CONSTRAINT uk_ser_name")

t_series_ser <-
  data.frame(
    "ser_id" = uuid::UUIDgenerate(n=nrow(res)),
    "ser_code" = res$ser_nameshort,
    "ser_name" = res$ser_namelong,
    "ser_spe_code" = "ANG",
    "ser_lfs_code" = res$ser_lfs_code,
    "ser_are_code" = res$ser_emu_nameshort,
    "ser_wkg_code" = "WGEEL",
    "ser_ver_code" = case_when(res$ser_dts_datasource == "dc_2019" ~ "ANG-2019-1",
                               res$ser_dts_datasource == "dc_2021" ~ "ANG-2021-1",
                               res$ser_dts_datasource == "dc_2022" ~ "ANG-2022-1",
                               res$ser_dts_datasource == "dc_2023" ~ "ANG-2023-1",
                               res$ser_dts_datasource == "dc_2024" ~ "ANG-2024-1",
                               .default = "ANG-2018-1"),
    "ser_cou_code" = res$ser_cou_code,
    "ser_hab_code" = NA,

```

```

"ser_gea_code" = ser$gea_issscfcg_code,
"ser_fiw_code" = NA,
"ser_mon_code" = NA,
"ser_uni_code" = ser$ser_uni_code,
"ser_effort_uni_code" = ser$ser_effort_uni_code,
"ser_description" = ser$ser_comment,
"ser_locationdescription" = ser$ser_locationdescription,
"ser_wltyp_code" = case_when(ser$ser_h ty_code == "T" ~ "T",
                             ser$ser_h ty_code == "C" ~ "MC",
                             ser$ser_h ty_code == "F" ~ "FW",
                             ser$ser_h ty_code == "MO" ~ "MO"),

"ser_stocking" = ser$ser_restocking,
"ser_stockingcomment" = NA,
"ser_protocol" = ser$ser_method,
"ser_samplingstrategy" = NA,
"ser_datarightsholder" = NA,
"ser_datelastupdate" = '2025-06-09',
"geom" = ser$geom
)

res <- dbWriteTable(con_diaspara_admin, "t_series_ser_temp",
                   t_series_ser, overwrite = TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM dateel.t_series_ser;")
dbExecute(con_diaspara_admin, "INSERT INTO dateel.t_series_ser
SELECT
ser_id::uuid,
ser_code,
ser_name,
ser_spe_code,
ser_lfs_code,
ser_are_code,
ser_wkg_code,
ser_ver_code,
ser_cou_code,
ser_hab_code,
ser_gea_code,
ser_fiw_code,
ser_mon_code,
ser_uni_code,
ser_effort_uni_code,
ser_description,
ser_locationdescription,
ser_wltyp_code,
ser_stocking,
ser_stockingcomment,

```

```

ser_protocol,
ser_samplingstrategy,
ser_datarightsholder,
ser_datelastupdate::date,
geom
FROM t_series_ser_temp") # 294

sai <- dbGetQuery(con_wgeel_distant,
"SELECT sai.* FROM datawg.t_samplinginfo_sai sai;
")

t_series_ser2 <-
data.frame(
  "ser_id" = uuid::UUIDgenerate(n=nrow(sai)),
  "ser_code" = sai$sai_id,
  "ser_name" = sai$sai_name,
  "ser_spe_code" = "ANG",
  "ser_lfs_code" = NA,
  "ser_are_code" = NA, # TODO sai_emu_nameshort sai_area_division
  "ser_wkg_code" = "WGEEL",
  "ser_ver_code" = case_when(sai$sai_dts_datasource == "dc_2019" ~ "WGEEL-2019-1",
                             sai$sai_dts_datasource == "dc_2021" ~ "WGEEL-2021-1",
                             sai$sai_dts_datasource == "dc_2022" ~ "WGEEL-2022-1",
                             sai$sai_dts_datasource == "dc_2023" ~ "WGEEL-2023-1",
                             sai$sai_dts_datasource == "dc_2024" ~ "WGEEL-2024-1",
                             .default = "WGEEL-2018-1"),
  "ser_cou_code" = sai$sai_cou_code,
  "ser_hab_code" = NA,
  "ser_gea_code" = NA,
  "ser_fiw_code" = NA,
  "ser_mon_code" = NA,
  "ser_uni_code" = NA,
  "ser_effort_uni_code" = NA,
  "ser_description" = sai$sai_comment,
  "ser_locationdescription" = NA,
  "ser_wltyp_code" = case_when(sai$sai_h ty_code == "T" ~ "T",
                                sai$sai_h ty_code == "C" ~ "MC",
                                sai$sai_h ty_code == "F" ~ "FW",
                                sai$sai_h ty_code == "MO" ~ "MO"),
  "ser_stocking" = NA,
  "ser_stockingcomment" = NA,
  "ser_protocol" = sai$sai_protocol,
  "ser_samplingstrategy" = sai$sai_samplingstrategy,
  "ser_datarightsholder" = NA,
  "ser_datelastupdate" = sai$sai_lastupdate,

```

```

    "geom" = NA # no geom in this table
  )

res <- dbWriteTable(con_diaspara_admin, "t_series_ser_temp2",
                    t_series_ser2, overwrite = TRUE)
dbExecute(con_diaspara_admin, "INSERT INTO dateel.t_series_ser
SELECT
  ser_id::uuid,
  ser_code,
  ser_name,
  ser_spe_code,
  ser_lfs_code,
  ser_are_code,
  ser_wkg_code,
  ser_ver_code,
  ser_cou_code,
  ser_hab_code,
  ser_gea_code,
  ser_fiw_code,
  ser_mon_code,
  ser_uni_code,
  ser_effort_uni_code,
  ser_description,
  ser_locationdescription,
  ser_wltyp_code,
  ser_stocking,
  ser_stockingcomment,
  ser_protocol,
  ser_samplingstrategy,
  ser_datarightsholder,
  ser_datelastupdate::date,
  NULL
FROM t_series_ser_temp2") # 252

```

ser_id	ser_code	ser_name	ser_sp
2445a12b-f5b9-4c8e-aa83-dcecc9e92717	31	ES_Astu_Esva _HIST	126281
95259d08-b9cf-4936-8484-0027c9c2d989	33	GB_Seve_Severn _HIST	126281
379b8d44-0251-4d00-82bd-b2212b6913ad	34	GB_Scot_Girnock_Burn_Scotland _HIST	126281
7e4fd77c-cf99-4eff-9c88-9d6d92b7242e	37	IE_West_Burrishoole _HIST	126281
b0354939-8b73-4b52-8b8c-2b1f95119950	40	NL_Neth_Ijsselmeer _HIST	126281
2e1d59f5-a843-4cc0-85e7-8a6f5b204014	43	NA_Balaton _HIST	126281
b314943a-d752-4a3c-995c-43e514c72a4d	662	ES_Anda_SanPedro _BIOM	126281

ff7d196a-ef19-46c9-a500-2216dde17eee	664	ES__Anda__Tinto__BIOM	126281
72feac60-485a-4740-970c-ed2ab042c40d	659	ES__Anda__Salado__BIOM	126281
8892a7b8-5476-488a-8447-99f358445517	KilY	Killough River	126281
0db2ddf8-d6b9-4894-85af-6f32589f2a26	LevS	Leven Silver	126281
878afb2b-7346-4dca-8ff5-67e7392f5f03	826	SE__Inla__Roxen__HIST	126281
d98b022b-df0e-442b-98e8-a998265eb38c	SuSY	Suffolk Stour Catchment	126281
fcadb13e-2fd3-4392-b622-8ed610cdb980	ImsaGY	Imsa Near Sandnes trapping all	126281
72854801-b3fa-4b81-8827-eb4eda137c43	LeeY	Lee River	126281
31cb9a15-7b2a-46e4-887c-93673d9cd859	FneS	Fane	126281
04a2dfd7-8444-4ba4-afe7-8a520c12df8e	BarS	Barrow	126281
1a6d2324-2d94-42d2-b3e4-4a65aa5ae98d	DeBY	Den Burg fyke net survey	126281
2d8f9e41-f487-4064-b7f1-6a21b084e434	IJsFVY	FYOE-IJM-Veg	126281
d8b66f7a-c834-4b1b-b423-6a43d3b7c06a	IJsFRY	FYOE-IJM-Rock	126281

1.13 Annual series table

This table holds annual data from the series.

SQL code to create table `dat.t_serannual_san`

```
DROP TABLE IF EXISTS dat.t_serannual_san CASCADE;
CREATE TABLE dat.t_serannual_san (
  san_ser_id UUID,
  CONSTRAINT fk_san_ser_id FOREIGN KEY (san_ser_id)
    REFERENCES dat.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  san_id SERIAL NOT NULL,
  CONSTRAINT c_uk_san_id UNIQUE (san_id, san_wkg_code),
  san_value NUMERIC NULL,
  san_year INTEGER NOT NULL,
  CONSTRAINT uk_san_year_svc UNIQUE(san_year, san_ser_id),
  san_comment TEXT NULL,
  san_effort NUMERIC NULL,
  san_datelastupdate DATE NOT NULL,
  san_qal_id INTEGER NOT NULL,
  san_qal_comment TEXT,
  san_wkg_code TEXT NOT NULL,
  CONSTRAINT fk_san_wkg_code FOREIGN KEY (san_wkg_code)
    REFERENCES "ref".tr_icworkinggroup_wkg(wkg_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  san_ver_code TEXT NOT NULL,
  CONSTRAINT fk_san_ver_code FOREIGN KEY (san_ver_code)
    REFERENCES ref.tr_version_ver(ver_code)
    ON UPDATE CASCADE ON DELETE RESTRICT
);
```

```

COMMENT ON TABLE dat.t_serannual_san IS
  'Table of annual abundance data for series in dat.t_series_ser, these are recruitment or survival,
  the content of the tables in datfeel, datnas, datbast ';

COMMENT ON COLUMN dat.t_serannual_san.san_ser_id IS
  'UUID, identifier of the series, primary key, references the table ref.tr_seriesvocab_svc (';
COMMENT ON COLUMN dat.t_serannual_san.san_id IS
  'INTEGER, autoincremented, unique for one working group';
COMMENT ON COLUMN dat.t_serannual_san.san_year IS
  'Year of monitoring, note that for some of the series this corresponds to the main migration year.
  For glass eel, months from september y-1 to august y should be denoted year y / For silver eel, months from
  COMMENT ON COLUMN dat.t_serannual_san.san_comment IS
  'Comment on the annual value of the series';
COMMENT ON COLUMN dat.t_serannual_san.san_effort IS
  'Eventually a measure of effort to collect the series, e.g. number of nr haul, nr fyke.day, nr fyke.
  check the t_metadataannual table for the unit used';
COMMENT ON COLUMN dat.t_serannual_san.san_datelastupdate IS
  'Date of last update on the annual data';
COMMENT ON COLUMN dat.t_serannual_san.san_qal_id IS
  'Quality ID code of the series';
COMMENT ON COLUMN dat.t_serannual_san.san_qal_comment IS
  'Comment related to data quality, e.g. why this year the series should not be used, or used';
COMMENT ON COLUMN dat.t_serannual_san.san_wkg_code IS
  'Code of the working group, one of WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dat.t_serannual_san.san_ver_code IS
  'Version code sourced from ref.tr_version_ver the data call e.g. WGNAS_2020_1, WGEEL_2016_1

ALTER TABLE dateel.t_serannual_san DROP CONSTRAINT fk_san_ver_code

ALTER TABLE dateel.t_serannual_san ADD CONSTRAINT fk_san_ver_code FOREIGN KEY (san_ver_code)
REFERENCES refeel.tr_version_ver(ver_code)
ON UPDATE CASCADE ON DELETE RESTRICT

```

1.14 Creating inherited table for annual series

SQL code to create tables `dateel.t_serannual_san`

```

-- DROP TABLE IF EXISTS dateel.t_serannual_san;

CREATE TABLE dateel.t_serannual_san (

```

```

CONSTRAINT fk_san_ser_id FOREIGN KEY (san_ser_id)
  REFERENCES dateel.t_series_ser (ser_id)
  ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT c_uk_san_id UNIQUE (san_id, san_wkg_code),
CONSTRAINT fk_san_wkg_code FOREIGN KEY (san_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code),
CONSTRAINT fk_san_ver_code FOREIGN KEY (san_ver_code)
  REFERENCES refeel.tr_version_ver(ver_code),
CONSTRAINT uk_san_year_svc UNIQUE(san_year, san_ser_id)
) INHERITS (dat.t_serannual_san);

COMMENT ON TABLE dateel.t_serannual_san IS
  'Table of annual abundance data for series in dat.t_series_ser, these are recruitment or s
the content of the tables in datfeel, datnas, datbast';

COMMENT ON COLUMN dateel.t_serannual_san.san_ser_id IS
  'UUID, identifier of the series, primary key, references the table ref.tr_seriesvocab_svc
COMMENT ON COLUMN dateel.t_serannual_san.san_id IS
  'INTEGER, autoincremented, unique for one working group';
COMMENT ON COLUMN dateel.t_serannual_san.san_year IS
  'Year of monitoring, note that for some of the series this corresponds to the main migrati
For glass eel, months from september y-1 to august y should be denoted year y / For silver e
COMMENT ON COLUMN dateel.t_serannual_san.san_comment IS
  'Comment on the annual value of the series';
COMMENT ON COLUMN dateel.t_serannual_san.san_effort IS
  'Eventually a measure of effort to collect the series, e.g. number of nr haul, nr fyke.day
check the t_metadataannual table for the unit used';
COMMENT ON COLUMN dateel.t_serannual_san.san_datelastupdate IS
  'Date of last update on the annual data';
COMMENT ON COLUMN dateel.t_serannual_san.san_qal_id IS
  'Quality ID code of the series';
COMMENT ON COLUMN dateel.t_serannual_san.san_qal_comment IS
  'Comment related to data quality, e.g. why this year the series should not be used, or used
COMMENT ON COLUMN dateel.t_serannual_san.san_wkg_code IS
  'Code of the working group, one of
WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dateel.t_serannual_san.san_ver_code IS
  'Version code sourced from ref.tr_version_ver the data call e.g. NAS_2025dc_2020, wgeel_20

GRANT ALL ON dateel.t_serannual_san TO diaspara_admin;
GRANT SELECT ON dateel.t_serannual_san TO diaspara_read;

```

1.15 Import data for annual series

Data are imported from WGEEL datawg.t_dataserie_dastable to dateel.t_serannual_ser.

```
das0 <- dbGetQuery(con_wgeel_distant, "SELECT * FROM datawg.t_dataserie_das;")

res <- dbGetQuery(con_diaspara, "SELECT * FROM dateel.t_serannual_san")
clipr::write_clip(colnames(res))

das0$das_dts_datasource[is.na(das0$das_dts_datasource)] <- "WGEEL-2016"
das0$das_dts_datasource <-
  paste0(gsub(pattern = "dc_", replacement = "WGEEL-",
    x= das0$das_dts_datasource), "-1")
das0$das_qal_id[is.na(das0$das_qal_id)] <- 1

nrow(das0) # 6523
das0 <- das0 |> filter(das0$das_qal_id <5)
nrow(das0) # 6402

ser <- dbGetQuery(con_diaspara,
  "SELECT ser_id, ser_code FROM dateel.t_series_ser")
ser0 <- dbGetQuery(con_wgeel_distant,
  "SELECT ser_id, ser_nameshort as ser_code FROM datawg.t_series_ser ser;")

# adding ser_code to the series.
das <- das0 |> rename(ser_id = das_ser_id) |>
  inner_join(ser0) |> select(-ser_id) |>
  inner_join(ser)
nrow(das) #6523

t_serannual_san_temp <-
  data.frame(
    "san_ser_id" = das$ser_id,
    "san_id" = das$das_id,
    "san_value" = das$das_value,
    "san_year" = das$das_year,
    "san_comment" = das$das_comment,
    "san_effort" = das$das_effort,
    "san_datelastupdate" = das$das_last_update,
    "san_qal_id" = das$das_qal_id,
    "san_qal_comment" = das$das_qal_comment,
    "san_wkg_code" = "WGEEL",
    "san_ver_code" = das$das_dts_datasource)
```

```

res <- dbWriteTable(con_diaspara_admin, "t_serannual_san_temp",
                    t_serannual_san_temp, overwrite = TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM dateel.t_serannual_san;")
dbExecute(con_diaspara_admin, "INSERT INTO dateel.t_serannual_san
SELECT
    san_ser_id::uuid,
    san_id,
    san_value,
    san_year,
    san_comment,
    san_effort,
    san_datelastupdate,
    san_qal_id,
    san_qal_comment,
    san_wkg_code,
    san_ver_code
FROM t_serannual_san_temp") # 6402

dbGetQuery(con_diaspara, "SELECT * FROM dateel.t_serannual_san limit 20;")|>
knitr::kable()|>
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))

```

Table 1.19: Table t

san_ser_id	san_id	san_value	san_year	san_comment
9e8e2e9d-8ed6-4a61-ac83-d9249370d9e5	2837	3.820000	2016	Change 2017 3,82 -> 4,279. I
9910419d-ee03-45ab-b4ea-409d72aa505b	2925	120.800000	2017	Additional new traps capture
4e662794-26bc-4da5-90d5-6e47a1455a59	2926	0.454000	2012	NA
4e662794-26bc-4da5-90d5-6e47a1455a59	2927	1.144000	2013	NA
4e662794-26bc-4da5-90d5-6e47a1455a59	2928	0.311000	2014	NA
4e662794-26bc-4da5-90d5-6e47a1455a59	2929	0.159000	2015	NA
4e662794-26bc-4da5-90d5-6e47a1455a59	2930	0.360000	2016	NA
4e662794-26bc-4da5-90d5-6e47a1455a59	2931	0.518000	2017	Trap refurbished, glass eel fel
6846b849-d11a-4eef-903d-39e7e30c503a	2932	0.720000	1987	NA
6846b849-d11a-4eef-903d-39e7e30c503a	2933	14.480000	1988	NA
6846b849-d11a-4eef-903d-39e7e30c503a	2934	0.259000	2007	NA
3ca798cf-9474-48a0-b742-2756aca6df3c	1061	4283.000000	1997	NA
56760cce-14d6-414b-98d0-871bd8a64384	1777	9453.000000	1960	NA
0d83cf08-df7a-4853-95d8-cbdc18cde5f2	632	36.700000	1989	NA
5534b8e4-ced3-4655-b5b7-8252cf01cfe0	680	121.000000	1952	NA
5534b8e4-ced3-4655-b5b7-8252cf01cfe0	690	185.000000	1962	NA
5534b8e4-ced3-4655-b5b7-8252cf01cfe0	695	258.000000	1967	NA
caa6d368-37a5-4093-9585-bc303c769c14	2160	25.200000	1985	number of hauls =6

4ae8d33f-3808-4258-a31f-7bf1f1233a8a	5440	115.000000	2013	Das__gal__id added in 2022.
11e4ae1e-c7a8-4ab1-a080-18545786886d	4537	0.152542	1991	NA

1.16 Creating table joining series and station

The table is created but empty. We need the stations in ICES first.

SQL code to create table `dat.tj_seriesstation_ses`

```
DROP TABLE IF EXISTS dat.tj_seriesstation_ses;
CREATE TABLE dat.tj_seriesstation_ses (
  ses_ser_id uuid PRIMARY KEY,
  CONSTRAINT fk_ses_ser_id FOREIGN KEY (ses_ser_id)
    REFERENCES dat.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT uk_ses_ser_id UNIQUE (ses_ser_id) ,
  ses_station_code INTEGER NULL,
  CONSTRAINT fk_station_code FOREIGN KEY (ses_station_code)
    REFERENCES "ref"."StationDictionary" ("Station_Code")
);
```

```
GRANT ALL ON dat.tj_seriesstation_ses TO diaspara_admin;
GRANT SELECT ON dat.tj_seriesstation_ses TO diaspara_read;
```

1.17 Creating inherited joining series and station

SQL code to create table `dateel.tj_seriesstation_ses`

```
DROP TABLE IF EXISTS dateel.tj_seriesstation_ses;
CREATE TABLE dateel.tj_seriesstation_ses (
  CONSTRAINT fk_ses_ser_id FOREIGN KEY (ses_ser_id)
    REFERENCES dat.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT uk_ses_ser_id UNIQUE (ses_ser_id) ,
  CONSTRAINT fk_station_code FOREIGN KEY (ses_station_code)
    REFERENCES "ref"."StationDictionary" ("Station_Code")
) inherits (dat.tj_seriesstation_ses);
```

```
GRANT ALL ON dateel.tj_seriesstation_ses TO diaspara_admin;
GRANT SELECT ON dateel.tj_seriesstation_ses TO diaspara_read;
```

Chapter 2

Creating additional table for stock annex

2.1 Creating table `t_recruitmentmetadata`

The code was developed during [WKEELDATA6](#) Currently the table is not intended to be held in the db There is still work to be done on the area referential for eel (starting summer 2025). Some referentials are coming from the old db (emu, sampling type), and are currently not transfered.

SQL code to create table `dateel.t_recruitmentmetadata_met`

```
-- notes I will need ref. tr_emu_emu for legacy with this table.
-- but as the table will not be placed in the db
-- I'm not programming foreign keys
-- see
--DROP TABLE IF EXISTS dateel.t_recruitmentmetadata_met ;
CREATE TABLE dateel.t_recruitmentmetadata_met (
  met_ser_id uuid PRIMARY KEY,
  -- the following constraint ensure 1:1 relation to series
  CONSTRAINT fk_met_ser_id FOREIGN KEY (met_ser_id)
    REFERENCES dat.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT uk_mrt_ser_id UNIQUE (met_ser_id) ,
  met_sam_id int4 NULL,
  --CONSTRAINT fk_met_sam_id FOREIGN KEY (met_sam_id)
  -- REFERENCES ref.tr_samplingtype_sam(sam_id)
  -- ON UPDATE CASCADE ON DELETE CASCADE,
  -- this one is a legacy from the old db, for this table
  -- intended only for WGEEL I'll not change it
```

```

met_emu_nameshort varchar(20) NOT NULL,
--CONSTRAINT fk_emu FOREIGN KEY (met_emu_nameshort,met_cou_code)
--REFERENCES ref.tr_emu_emu(emu_nameshort,emu_cou_code)
--ON UPDATE CASCADE ON DELETE CASCADE,
met_qal_id INTEGER,
--FOREIGN KEY (met_qal_id)
--REFERENCES ref.tr_quality_qal(qal_id),
-- currently tr_quality_qal is only in ref (with values from wgeel) not refeel check ??
met_mixturegy BOOLEAN,
met_wetted_above NUMERIC,
met_wetted_total NUMERIC,
met_surfacebasin_upstream NUMERIC,
met_surfacebasin NUMERIC,
met_distanceseakm numeric NULL,
met_marinearea TEXT,
met_start_year INTEGER,
met_end_year INTEGER,
met_duration INTEGER,
met_missing INTEGER,
met_individual_length BOOLEAN,
met_individual_mass BOOLEAN,
met_conversion BOOLEAN,
met_conversion_comment BOOLEAN,
met_raw_to_reported TEXT,
met_internal_issues BOOLEAN,
met_internal_issues_comment TEXT,
met_other BOOLEAN,
met_other_comment TEXT,
met_restocking BOOLEAN,
met_restocking_comment TEXT,
met_barrier BOOLEAN,
met_barrier_comment TEXT,
met_mortality BOOLEAN,
met_mortality_comment TEXT,
met_ext_issues BOOLEAN,
met_ext_issues_comment TEXT,
met_environment BOOLEAN,
met_environment_comment TEXT)

GRANT ALL ON dateel.t_recruitmentmetadata_met TO diaspara_admin;
GRANT SELECT ON dateel.t_recruitmentmetadata_met TO diaspara_read;

```

met_ser_id	met_sam_id	met_emu_nameshort	met_qal_id	met_mixturegy	met_wetted_above
NA	NA	NA	NA	NA	NA
:_____	_____:-	:_____	_____:-	:_____	_____:-

Chapter 3

Creating group metrics

3.1 Create group table

i Note for group metrics

Need a foreign key on both species and lfs code.

SQL code to create table `dat.t_group_gr`

```
-- DROP TABLE dat.t_group_gr CASCADE;

CREATE TABLE dat.t_group_gr (
  gr_id serial4 NOT NULL,
  gr_ser_id UUID NOT NULL,
  CONSTRAINT fk_gr_ser_id FOREIGN KEY (gr_ser_id)
  REFERENCES dat.t_series_ser (ser_id)
  ON UPDATE CASCADE ON DELETE CASCADE,
  gr_gr_id INTEGER NULL,
  CONSTRAINT fk_gr_gr_id FOREIGN KEY (gr_gr_id, gr_wkg_code)
  REFERENCES dat.t_group_gr(gr_id, gr_wkg_code)
  ON UPDATE CASCADE ON DELETE CASCADE,
  gr_wkg_code TEXT NOT NULL,
  CONSTRAINT fk_gr_wkg_code FOREIGN KEY (gr_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT t_group_gr_pkey PRIMARY KEY (gr_id, gr_wkg_code),
  gr_spe_code TEXT,
  CONSTRAINT fk_gr_spe_code FOREIGN KEY (gr_spe_code)
  REFERENCES "ref".tr_species_spe(spe_code)
```

```

ON UPDATE CASCADE ON DELETE RESTRICT ,
gr_lfs_code TEXT,
CONSTRAINT fk_gr_lfs_code_gr_spe_code FOREIGN KEY (gr_lfs_code, gr_spe_code)
REFERENCES "ref".tr_lifestage_lfs (lfs_code, lfs_spe_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
gr_sex_code TEXT,
CONSTRAINT ck_nn_gr_sex_code_gr_gr_id CHECK ((gr_gr_id IS NULL AND gr_sex_code IS NULL) OR
CONSTRAINT ck_gr_sex_code CHECK (gr_sex_code = 'M' OR gr_sex_code = 'F' OR gr_sex_code IS NULL),
gr_year int4 NULL,
gr_number int4 NULL,
gr_comment text NULL,
gr_lastupdate date DEFAULT CURRENT_DATE NOT NULL,
gr_ver_code TEXT NOT NULL,
CONSTRAINT fk_gr_ver_code FOREIGN KEY (gr_ver_code)
REFERENCES ref.tr_version_ver(ver_code)
ON UPDATE CASCADE ON DELETE RESTRICT
);

COMMENT ON TABLE dat.t_group_gr IS
'Table identifying the group metrics, a group metric corresponds to a
number of fish sampled for a given year, mostly to describe the annual series. Comments can
on the sampling with gr_comments. There can be several group metrics for the same year, for
with sampling designs for different stages';

COMMENT ON COLUMN dat.t_group_gr.gr_id IS
'Group ID, serial primary key on gr_id and gr_wkg_code';
COMMENT ON COLUMN dat.t_group_gr.gr_gr_id IS
'Parent group ID, used when giving separate metrics for male and females, in that case the
COMMENT ON COLUMN dat.t_group_gr.gr_wkg_code IS
'Code of the working group, one of
WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dat.t_group_gr.gr_lfs_code IS
'Life stage code';
COMMENT ON COLUMN dat.t_group_gr.gr_spe_code IS
'Species code';
COMMENT ON COLUMN dat.t_group_gr.gr_sex_code IS
'Sex code only for subgroups male or female';
COMMENT ON COLUMN dat.t_group_gr.gr_year IS
'The year';
COMMENT ON COLUMN dat.t_group_gr.gr_number IS
'Number of fish in the group';
COMMENT ON COLUMN dat.t_group_gr.gr_comment IS
'Comment on the group metric, including on the sampling design applied to that particular y
COMMENT ON COLUMN dat.t_group_gr.gr_lastupdate IS
'Last update, inserted automatically';

```

```

COMMENT ON COLUMN dat.t_group_gr.gr_ver_code IS
  'Version code as in tr_version_ver, corresponds to the working group code WGNAS-2024-1 WGE

GRANT ALL ON dat.t_group_gr TO diaspara_admin;
GRANT SELECT ON dat.t_group_gr TO diaspara_read;

```

3.2 Create group table for dateel

SQL code to create table dateel.t_group_gr

```

-- DROP TABLE IF EXISTS dateel.t_group_gr;

CREATE TABLE dateel.t_group_gr (
  CONSTRAINT fk_gr_ser_id FOREIGN KEY (gr_ser_id)
    REFERENCES dateel.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT fk_gr_gr_id FOREIGN KEY (gr_gr_id, gr_wkg_code)
    REFERENCES dateel.t_group_gr(gr_id, gr_wkg_code)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT fk_gr_wkg_code FOREIGN KEY (gr_wkg_code)
    REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT t_group_gr_pkey PRIMARY KEY (gr_id, gr_wkg_code),
  CONSTRAINT fk_gr_ver_code FOREIGN KEY (gr_ver_code)
    REFERENCES refeel.tr_version_ver(ver_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_gr_lfs_code_gr_spe_code FOREIGN KEY (gr_lfs_code, gr_spe_code)
    REFERENCES "ref".tr_lifestage_lfs (lfs_code, lfs_spe_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_gr_spe_code FOREIGN KEY (gr_spe_code)
    REFERENCES "ref".tr_species_spe(spe_code)
    ON UPDATE CASCADE ON DELETE RESTRICT
) INHERITS (dat.t_group_gr);

COMMENT ON TABLE dateel.t_group_gr IS
  'Table identifying the group metrics, a group metric corresponds to a
  number of fish sampled for a given year, mostly to describe the annual series. Comments can
  on the sampling with gr_comments. There can be several group metrics for the same year, for
  with sampling designs for different stages';

COMMENT ON COLUMN dateel.t_group_gr.gr_id IS
  'Group ID, serial primary key on gr_id and gr_wkg_code';

```

```

COMMENT ON COLUMN dat.t_group_gr.gr_gr_id IS
  'Parent group ID, used when giving separate metrics for male and females, in that case the
COMMENT ON COLUMN dateel.t_group_gr.gr_wkg_code IS
  'Code of the working group, one of
WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dateel.t_group_gr.gr_year IS
  'The year';
COMMENT ON COLUMN dateel.t_group_gr.gr_number IS
  'Number of fish in the group';
COMMENT ON COLUMN dateel.t_group_gr.gr_comment IS
  'Comment on the group metric, including on the sampling design applied to that particular y
COMMENT ON COLUMN dateel.t_group_gr.gr_lastupdate IS
  'Last update, inserted automatically';
COMMENT ON COLUMN dateel.t_group_gr.gr_ver_code IS
  'Version code as in tr_version_ver, corresponds to the working group code WGNAS-2024-1 WGE
COMMENT ON COLUMN dat.t_group_gr.gr_lfs_code IS
  'Life stage code';
COMMENT ON COLUMN dat.t_group_gr.gr_spe_code IS
  'Species code';
COMMENT ON COLUMN dat.t_group_gr.gr_sex_code IS
  'Sex code only for subgroups male or female';
GRANT ALL ON dateel.t_group_gr TO diaspara_admin;
GRANT SELECT ON dateel.t_group_gr TO diaspara_read;

```

3.3 Import data for group

Scripts differ a bit for series and sampling. Check for some series with comment so see if need removing.

all related metrics have qal_id=22 following data call 2022

SQL code to Import data

```

/*
SELECT DISTINCT gr_dts_datasource
FROM datwgeel.t_groupseries_grser
*/

DELETE FROM dateel.t_group_gr;
INSERT INTO dateel.t_group_gr
(gr_id, gr_ser_id, gr_gr_id, gr_wkg_code, gr_spe_code, gr_lfs_code, gr_year, gr_number, gr_c
SELECT
gr_id,
tss2.ser_id ,
NULL AS gr_gr_id,
'WGEEL' AS gr_wkg_code,

```

```

'127186' AS gr_spe_code,
tss.ser_lfs_code AS gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
CASE WHEN gr_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN gr_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN gr_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN gr_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN gr_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN gr_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS gr_ver_code
FROM datwgeel.t_groupseries_grser grser
JOIN datwgeel.t_series_ser AS tss ON grser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort; --2681

```

```

INSERT INTO dateel.t_group_gr
(gr_id, gr_ser_id, gr_gr_id, gr_wkg_code, gr_spe_code, gr_lfs_code, gr_year, gr_number, gr_
SELECT
gr_id,
tss2.ser_id AS gr_ser_id,
NULL AS gr_gr_id,
'WGEEL' AS gr_wkg_code,
'127186' AS gr_spe_code,
grsa_lfs_code AS gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
CASE WHEN gr_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN gr_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN gr_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN gr_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN gr_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN gr_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS gr_ver_code
FROM datwgeel.t_groupsamp_grsa
JOIN datwgeel.t_samplinginfo_sai AS tss ON grsa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text; --798

```

```

-- INSERT Males AND Females WITH gr_gr_id

DROP SEQUENCE IF EXISTS seq_group;
CREATE TEMPORARY SEQUENCE seq_group;
SELECT setval('seq_group', (SELECT max(gr_id) FROM dateel.t_group_gr)); -- 13741

-- Males with gr_id from series (subgroupsample)

INSERT INTO dateel.t_group_gr
(gr_id,
gr_ser_id,
gr_gr_id,
gr_wkg_code,
gr_spe_code,
gr_sex_code,
gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
gr_ver_code)
SELECT
nextval('seq_group') AS gr_id,
tss2.ser_id AS gr_ser_id,
grser.gr_id AS gr_gr_id,
'WGEEL' AS gr_wkg_code,
'127186' AS gr_spe_code,
'M' AS gr_sex_code,
tss.ser_lfs_code AS gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
CASE WHEN gr_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN gr_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN gr_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN gr_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN gr_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN gr_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS gr_ver_code
FROM
datwgeel.t_metricgroupseries_megser
JOIN datwgeel.t_groupseries_grser grser ON meg_gr_id = gr_id
JOIN datwgeel.t_series_ser AS tss ON grser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort

```

```

WHERE meg_mty_id IN (18,19,20); --409

# Females WITH gr_gr_id series (subgroupsample)

INSERT INTO dateel.t_group_gr
(gr_id,
gr_ser_id,
gr_gr_id,
gr_wkg_code,
gr_spe_code,
gr_sex_code,
gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
gr_ver_code)
SELECT
nextval('seq_group') AS gr_id,
tss2.ser_id AS gr_ser_id,
grser.gr_id AS gr_gr_id,
'WGEEL' AS gr_wkg_code,
'127186' AS gr_spe_code,
'F' AS gr_sex_code,
tss.ser_lfs_code AS gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
CASE WHEN gr_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN gr_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN gr_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN gr_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN gr_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN gr_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS gr_ver_code
FROM
datwgeel.t_metricgroupseries_megser
JOIN datwgeel.t_groupseries_grser grser ON meg_gr_id = gr_id
JOIN datwgeel.t_series_ser AS tss ON grser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort
WHERE meg_mty_id IN (21,22,23); --553

# males from sampling

```

```

INSERT INTO dateel.t_group_gr
(gr_id,
gr_ser_id,
gr_gr_id,
gr_wkg_code,
gr_spe_code,
gr_sex_code,
gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
gr_ver_code)
SELECT
nextval('seq_group') AS gr_id,
tss2.ser_id AS gr_ser_id,
grsa.gr_id AS gr_gr_id,
'WGEEL' AS gr_wkg_code,
'127186' AS gr_spe_code,
'M' AS gr_sex_code,
grsa_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
CASE WHEN gr_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN gr_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN gr_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN gr_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN gr_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN gr_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS gr_ver_code
FROM
datwgeel.t_metricgroupsamp_megsa
JOIN datwgeel.t_groupsamp_grsa grsa ON meg_gr_id = gr_id
JOIN datwgeel.t_samplinginfo_sai AS tss ON grsa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text
WHERE meg_mty_id IN (18,19,20); --544

# Females from sampling

INSERT INTO dateel.t_group_gr
(gr_id,

```

```

gr_ser_id,
gr_gr_id,
gr_wkg_code,
gr_spe_code,
gr_sex_code,
gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
gr_ver_code)
SELECT
nextval('seq_group') AS gr_id,
tss2.ser_id AS gr_ser_id,
grsa.gr_id AS gr_gr_id,
'WGEEL' AS gr_wkg_code,
'127186' AS gr_spe_code,
'F' AS gr_sex_code,
grsa_lfs_code AS gr_lfs_code,
gr_year,
gr_number,
gr_comment,
gr_lastupdate,
CASE WHEN gr_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN gr_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN gr_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN gr_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN gr_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN gr_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS gr_ver_code
FROM
datwgeel.t_metricgroupsamp_megsa
JOIN datwgeel.t_groupsamp_grsa grsa ON meg_gr_id = gr_id
JOIN datwgeel.t_samplinginfo_sai AS tss ON grsa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text
WHERE meg_mty_id IN (21,22,23); --1183

```

Tabl

gr_id	gr_ser_id	gr_gr_id	gr_wkg_code	gr_spe_code	gr_lfs_code
12932	2d8f9e41-f487-4064-b7f1-6a21b084e434	NA	WGEEL	126281	Y
1049	2d8f9e41-f487-4064-b7f1-6a21b084e434	NA	WGEEL	126281	Y
1048	2d8f9e41-f487-4064-b7f1-6a21b084e434	NA	WGEEL	126281	Y
1047	2d8f9e41-f487-4064-b7f1-6a21b084e434	NA	WGEEL	126281	Y

1046	2d8f9e41-f487-4064-b7f1-6a21b084e434	NA	WGEEL	126281	Y
690	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
689	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
688	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
687	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
686	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
685	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
684	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
683	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
682	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
681	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
680	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
679	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
678	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
677	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y
676	d98b022b-df0e-442b-98e8-a998265eb38c	NA	WGEEL	126281	Y

Chapter 4

Grouptrait table

4.1 Create table

SQL code to create tables `dat.t_grouptrait_grt` and `dateel.t_grouptrait_grt`

```
-- DROP TABLE IF EXISTS dat.t_grouptrait_grt CASCADE;

CREATE TABLE dat.t_grouptrait_grt (
  grt_ser_id uuid,
  CONSTRAINT fk_grt_ser_id FOREIGN KEY (grt_ser_id)
    REFERENCES dat.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  grt_wkg_code TEXT NOT NULL,
  CONSTRAINT fk_grt_wkg_code FOREIGN KEY (grt_wkg_code)
    REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  grt_spe_code TEXT NOT NULL,
  CONSTRAINT fk_grt_spe_code FOREIGN KEY (grt_spe_code)
    REFERENCES ref.tr_species_spe(spe_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  grt_id serial4 NOT NULL,
  grt_gr_id int4 NOT NULL,
  CONSTRAINT fk_grt_gr_id FOREIGN KEY (grt_gr_id, grt_wkg_code)
    REFERENCES dat.t_group_gr(gr_id,gr_wkg_code)
    ON UPDATE CASCADE ON DELETE CASCADE,
  grt_tra_code TEXT NOT NULL,
  CONSTRAINT fk_grt_tra_code FOREIGN KEY (grt_tra_code)
    REFERENCES ref.tr_trait_tra(tra_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
```

```

grt_value numeric NULL,
grt_trv_code TEXT,
CONSTRAINT fk_grt_trv_tra_code FOREIGN KEY (grt_trv_code, grt_tra_code,grt_wkg_code)
REFERENCES ref.tr_traitvaluequal_trv(trv_code, trv_trq_code, trv_wkg_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
grt_trm_code TEXT,
CONSTRAINT fk_grt_trm_code FOREIGN KEY (grt_trm_code)
REFERENCES ref.tr_traitmethod_trm(trm_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
grt_last_update date DEFAULT CURRENT_DATE NOT NULL,
grt_qal_code int4 NULL,
    CONSTRAINT fk_grt_qal_id FOREIGN KEY (grt_qal_code)
REFERENCES ref.tr_quality_qal(qal_code) ON UPDATE CASCADE,
grt_ver_code TEXT NOT NULL,
CONSTRAINT fk_grt_ver_code FOREIGN KEY (grt_ver_code)
REFERENCES ref.tr_version_ver(ver_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT uk_grt_gr UNIQUE (grt_gr_id, grt_trm_code, grt_wkg_code),
CONSTRAINT t_grouptrait_grt_pkey PRIMARY KEY (grt_id, grt_wkg_code),
CONSTRAINT ck_qualitative_or_numeric CHECK
(
    (grt_value IS NULL AND grt_trv_code IS NOT NULL) OR
    (grt_value IS NOT NULL AND grt_trv_code IS NULL)
);
CREATE INDEX dat_t_grouptrait_grt_idx ON dat.t_grouptrait_grt USING btree (grt_gr_id);

GRANT ALL ON dat.t_grouptrait_grt TO diaspara_admin;
GRANT SELECT ON dat.t_grouptrait_grt TO diaspara_read;

DROP TABLE IF EXISTS dateel.t_grouptrait_grt;
CREATE TABLE dateel.t_grouptrait_grt (
    CONSTRAINT fk_grt_ser_id FOREIGN KEY (grt_ser_id)
        REFERENCES dateel.t_series_ser (ser_id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_grt_wkg_code FOREIGN KEY (grt_wkg_code)
        REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT fk_grt_spe_code FOREIGN KEY (grt_spe_code)
        REFERENCES ref.tr_species_spe(spe_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT fk_grt_gr_id FOREIGN KEY (grt_gr_id, grt_wkg_code)
        REFERENCES dateel.t_group_gr(gr_id,gr_wkg_code)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_grt_tra_code FOREIGN KEY (grt_tra_code)

```

```

REFERENCES refeel.tg_trait_tra(tra_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_grt_trv_tra_code FOREIGN KEY (grt_trv_code, grt_tra_code)
-- unlike in dat.t_grouptrait_grt this one does not take ref TO wkgcode, no need AS we are
REFERENCES refeel.tr_traitvaluequal_trv(trv_code, trv_trq_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_grt_trm_code FOREIGN KEY (grt_trm_code)
REFERENCES refeel.tr_traitmethod_trm(trm_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_grt_qal_id FOREIGN KEY (grt_qal_code)
REFERENCES ref.tr_quality_qal(qal_code) ON UPDATE CASCADE,
CONSTRAINT fk_grt_ver_code FOREIGN KEY (grt_ver_code)
REFERENCES refeel.tr_version_ver(ver_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT uk_trv_gr_id_trv_tra_code UNIQUE(grt_gr_id,grt_tra_code)
) INHERITS (dat.t_grouptrait_grt);
CREATE INDEX dateel.t_grouptrait_grt_idx ON dateel.t_grouptrait_grt
USING btree (grt_gr_id);
GRANT ALL ON dateel.t_grouptrait_grt TO diaspara_admin;
GRANT SELECT ON dateel.t_grouptrait_grt TO diaspara_read;

COMMENT ON TABLE dateel.t_grouptrait_grt IS
'Table joining groups and traits';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_ser_id IS
'Series UUID';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_wkg_code IS
'Working group on of WGEEL, WGNAS, WGBAST ...';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_spe_code IS
'Species code here ''127186''';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_id IS
'ID, integer, unique for wkg_code';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_gr_id IS
'ID of the group';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_tra_code IS
'Code of the trait, e.g. Lengthmm';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_value IS
'Value for numeric';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_trv_code IS
'Value for qualitative see refeel.tr_traitvaluequal_trv';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_trm_code IS
'Method see refeel.tr_traitmethod_trm';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_last_update IS
'date last update';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_qal_code IS

```

```
'Quality code references ref.tr_quality_qal';
COMMENT ON COLUMN dateel.t_grouptrait_grt.grt_ver_code IS
'version e.g. WGEEL_2024_1';

-- TODO trigger on date
-- TODO
```

4.2 Import data

[illegible]

Figure 4.1: Mapping previous database for groups :This is how the mapping was done for groups, on the left the new db with in red the fields for individual metric (not in group metrics), on the right, the old db, with in oranges fields that need to be mapped to method. In yellow the groups that now refer to subgroup by sex recreated at previous step

SQL code to import data

```
/*
 * Insert numeric for group - series
 */
DELETE FROM dateel.t_grouptrait_grt;
```

```
INSERT INTO dateel.t_grouptrait_grt
(grt_ser_id,
grt_wkg_code,
grt_spe_code,
grt_id,
grt_gr_id,
grt_tra_code,
grt_value,
grt_trv_code,
grt_trm_code,
grt_last_update,
```

```

grt_qal_code,
grt_ver_code)
-- extract method from table
-- this will extract 3 columns, meg_gr_id, meg_method_anguillicola and meg_method_sex
WITH a1 AS (
SELECT meg_gr_id,
       meg_value AS meg_method_sex
FROM datwgeel.t_metricgroupseries_megser
WHERE meg_mty_id = 27
AND meg_value IS NOT NULL),
a2 AS (
SELECT meg_gr_id,
       meg_value AS meg_method_anguillicola
FROM datwgeel.t_metricgroupseries_megser
WHERE meg_mty_id = 28
AND meg_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.meg_gr_id, a2.meg_gr_id) AS meg_gr_id,
       meg_method_sex,
       meg_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.meg_gr_id = a2.meg_gr_id)
-- Insert select query
SELECT
tss2.ser_id AS grt_ser_id,
'WGEEL' AS grt_wkg_code,
'127186' AS grt_spe_code,
meg_id AS grt_id,
megser.meg_gr_id AS grt_gr_id,
CASE WHEN meg_mty_id = 1 THEN 'Lengthmm'
WHEN meg_mty_id = 2 THEN 'Weightg'
WHEN meg_mty_id = 3 THEN 'Ageyear'
WHEN meg_mty_id = 6 THEN 'Female_proportion'
WHEN meg_mty_id = 7 THEN 'Differentiated_proportion'
WHEN meg_mty_id = 8 THEN 'Anguillicola_proportion'
WHEN meg_mty_id = 9 THEN 'Anguillicola_intensity'
WHEN meg_mty_id = 10 THEN 'Muscle_lipid'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid' -- different method see method
WHEN meg_mty_id = 12 THEN 'Sum_6_pcb'
WHEN meg_mty_id = 13 THEN 'Evex_proportion'
WHEN meg_mty_id = 14 THEN 'Hva_proportion'
WHEN meg_mty_id = 15 THEN 'Pb'
WHEN meg_mty_id = 16 THEN 'Hg'
WHEN meg_mty_id = 17 THEN 'Cd'
WHEN meg_mty_id = 24 THEN 'G_in_gy_proportion'
WHEN meg_mty_id = 25 THEN 'S_in_ys_proportion'

```

```

WHEN meg_mty_id = 26 THEN 'Teq'
ELSE 'problem' END AS grt_tra_code,
meg_value AS grt_value,
NULL AS grt_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN meg_mty_id = 10 THEN 'Muscle_lipid_fatmeter'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see method
WHEN meg_mty_id = 6 AND meg_method_sex = 1 THEN 'Gonadal_inspection'
WHEN meg_mty_id = 6 AND meg_method_sex = 0 THEN 'Length_based_sex'
WHEN meg_mty_id = 8 AND meg_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN meg_mty_id = 8 AND meg_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
WHEN meg_mty_id = 9 AND meg_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN meg_mty_id = 9 AND meg_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
ELSE NULL END AS grt_trm_code,
meg_last_update AS grt_last_update,
meg_qal_id AS grt_qal_code,
CASE WHEN meg_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN meg_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN meg_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN meg_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN meg_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN meg_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS grt_ver_code
FROM datwgeel.t_metricgroupseries_megser megser
JOIN datwgeel.t_groupseries_grser grser ON megser.meg_gr_id = gr_id
JOIN datwgeel.t_series_ser AS tss ON grser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort
LEFT JOIN mm ON megser.meg_gr_id = mm.meg_gr_id -- joining subquery
WHERE meg_mty_id IN (1,2,3,6,7,8,9,10,11,12,13,14,15,16,17,24,25,26); --4194

/*
 * Insert numeric for group - sampling
 * Some values have both gravimeter and fatmeter, gravimeter chosen.
 */

INSERT INTO dateel.t_grouptrait_grt
(grt_ser_id,
grt_wkg_code,
grt_spe_code,
grt_id,
grt_gr_id,
grt_tra_code,

```

```

grt_value,
grt_trv_code,
grt_trm_code,
grt_last_update,
grt_qal_code,
grt_ver_code)
-- extract method from table
-- this will extract 3 columns, meg_gr_id, meg_method_anguillicola and meg_method_sex
WITH a1 AS (
SELECT meg_gr_id,
       meg_value AS meg_method_sex
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 27
AND meg_value IS NOT NULL),
a2 AS (
SELECT meg_gr_id,
       meg_value AS meg_method_anguillicola
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 28
AND meg_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.meg_gr_id, a2.meg_gr_id) AS meg_gr_id,
       meg_method_sex,
       meg_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.meg_gr_id = a2.meg_gr_id)
-- Insert select query
SELECT
tss2.ser_id AS grt_ser_id,
'WGEEL' AS grt_wkg_code,
'127186' AS grt_spe_code,
meg_id AS grt_id,
megsa.meg_gr_id AS grt_gr_id,
CASE WHEN meg_mty_id = 1 THEN 'Lengthmm'
WHEN meg_mty_id = 2 THEN 'Weightg'
WHEN meg_mty_id = 3 THEN 'Ageyear'
WHEN meg_mty_id = 6 THEN 'Female_proportion'
WHEN meg_mty_id = 7 THEN 'Differentiated_proportion'
WHEN meg_mty_id = 8 THEN 'Anguillicola_proportion'
WHEN meg_mty_id = 9 THEN 'Anguillicola_intensity'
WHEN meg_mty_id = 10 THEN 'Muscle_lipid'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid' -- different method see method
WHEN meg_mty_id = 12 THEN 'Sum_6_pcb'
WHEN meg_mty_id = 13 THEN 'Evex_proportion'
WHEN meg_mty_id = 14 THEN 'Hva_proportion'
WHEN meg_mty_id = 15 THEN 'Pb'

```

```

WHEN meg_mty_id = 16 THEN 'Hg'
WHEN meg_mty_id = 17 THEN 'Cd'
WHEN meg_mty_id = 24 THEN 'G_in_gy_proportion'
WHEN meg_mty_id = 25 THEN 'S_in_ys_proportion'
WHEN meg_mty_id = 26 THEN 'Teq'
ELSE 'problem' END AS grt_tra_code,
meg_value AS grt_value,
NULL AS grt_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN meg_mty_id = 10 THEN 'Muscle_lipid_fatmeter'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see method
WHEN meg_mty_id = 6 AND meg_method_sex = 1 THEN 'Gonadal_inspection'
WHEN meg_mty_id = 6 AND meg_method_sex = 0 THEN 'Length_based_sex'
WHEN meg_mty_id = 8 AND meg_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN meg_mty_id = 8 AND meg_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
WHEN meg_mty_id = 9 AND meg_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN meg_mty_id = 9 AND meg_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
ELSE NULL END AS grt_trm_code,
meg_last_update AS grt_last_update,
meg_qal_id AS grt_qal_code,
CASE WHEN meg_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN meg_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN meg_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN meg_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN meg_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN meg_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS grt_ver_code
FROM datwgeel.t_metricgroupsamp_megsa megsa
JOIN datwgeel.t_groupsamp_grsa ON megsa.meg_gr_id = gr_id
JOIN datwgeel.t_samplinginfo_sai AS tss ON grsa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text
LEFT JOIN mm ON megsa.meg_gr_id = mm.meg_gr_id -- joining subquery
WHERE meg_mty_id IN (1,2,3,6,7,8,9,11,12,13,14,15,16,17,24,25,26); --3028

-- Insert only fatmeter where gravimeter does not exist

INSERT INTO dateel.t_grouptrait_grt
(grt_ser_id,
grt_wkg_code,
grt_spe_code,
grt_id,
grt_gr_id,
grt_tra_code,
grt_value,
grt_trv_code,

```

```

grt_trm_code,
grt_last_update,
grt_qal_code,
grt_ver_code)
WITH a1 AS (
SELECT meg_gr_id,
      meg_value AS meg_method_sex
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 27
AND meg_value IS NOT NULL),
a2 AS (
SELECT meg_gr_id,
      meg_value AS meg_method_anguillicola
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 28
AND meg_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.meg_gr_id, a2.meg_gr_id) AS meg_gr_id,
      meg_method_sex,
      meg_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.meg_gr_id = a2.meg_gr_id),
-- Insert select query
fatmeter AS (
SELECT
tss2.ser_id AS grt_ser_id,
'WGEEL' AS grt_wkg_code,
'127186' AS grt_spe_code,
meg_id AS grt_id,
megsa.meg_gr_id AS grt_gr_id,
CASE WHEN meg_mty_id = 1 THEN 'Lengthmm'
WHEN meg_mty_id = 2 THEN 'Weightg'
WHEN meg_mty_id = 3 THEN 'Ageyear'
WHEN meg_mty_id = 6 THEN 'Female_proportion'
WHEN meg_mty_id = 7 THEN 'Differentiated_proportion'
WHEN meg_mty_id = 8 THEN 'Anguillicola_proportion'
WHEN meg_mty_id = 9 THEN 'Anguillicola_intensity'
WHEN meg_mty_id = 10 THEN 'Muscle_lipid'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid' -- different method see method
WHEN meg_mty_id = 12 THEN 'Sum_6_pcb'
WHEN meg_mty_id = 13 THEN 'Evex_proportion'
WHEN meg_mty_id = 14 THEN 'Hva_proportion'
WHEN meg_mty_id = 15 THEN 'Pb'
WHEN meg_mty_id = 16 THEN 'Hg'
WHEN meg_mty_id = 17 THEN 'Cd'
WHEN meg_mty_id = 24 THEN 'G_in_gy_proportion'

```

```

WHEN meg_mty_id = 25 THEN 'S_in_ys_proportion'
WHEN meg_mty_id = 26 THEN 'Teq'
ELSE 'problem' END AS grt_tra_code,
meg_value AS grt_value,
NULL AS grt_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN meg_mty_id = 10 THEN 'Muscle_lipid_fatmeter'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see method
WHEN meg_mty_id = 6 AND meg_method_sex = 1 THEN 'Gonadal_inspection'
WHEN meg_mty_id = 6 AND meg_method_sex = 0 THEN 'Length_based_sex'
WHEN meg_mty_id = 8 AND meg_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN meg_mty_id = 8 AND meg_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
WHEN meg_mty_id = 9 AND meg_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN meg_mty_id = 9 AND meg_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
ELSE NULL END AS grt_trm_code,
meg_last_update AS grt_last_update,
meg_qal_id AS grt_qal_code,
CASE WHEN meg_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN meg_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN meg_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN meg_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN meg_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN meg_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS grt_ver_code
FROM datwgeel.t_metricgroupsamp_megsa megsa
JOIN datwgeel.t_groupsamp_grsa ON megsa.meg_gr_id = gr_id
JOIN datwgeel.t_samplinginfo_sai AS tss ON grsa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text
LEFT JOIN mm ON megsa.meg_gr_id = mm.meg_gr_id -- joining subquery
WHERE meg_mty_id = 10)
SELECT * FROM fatmeter WHERE grt_gr_id NOT IN
(SELECT grt_gr_id FROM dateel.t_grouptrait_grt WHERE grt_tra_code = 'Muscle_lipid')
; --23

-- GROUP metrics FOR males sampling
SELECT count(*) FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id IN (18,19,20);--544

INSERT INTO dateel.t_grouptrait_grt
(grt_ser_id,

```

```

grt_wkg_code,
grt_spe_code,
grt_id,
grt_gr_id,
grt_tra_code,
grt_value,
grt_trv_code,
grt_trm_code,
grt_last_update,
grt_qal_code,
grt_ver_code)
-- extract method from table
-- this will extract 3 columns, meg_gr_id, meg_method_anguillicola and meg_method_sex
WITH a1 AS (
SELECT meg_gr_id,
      meg_value AS meg_method_sex
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 27
AND meg_value IS NOT NULL),
a2 AS (
SELECT meg_gr_id,
      meg_value AS meg_method_anguillicola
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 28
AND meg_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.meg_gr_id, a2.meg_gr_id) AS meg_gr_id,
      meg_method_sex,
      meg_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.meg_gr_id = a2.meg_gr_id)
-- Insert select query
SELECT DISTINCT ON (meg_id)
tss2.ser_id AS grt_ser_id,
'WGEEL' AS grt_wkg_code,
'127186' AS grt_spe_code,
meg_id AS grt_id,
gr.gr_id AS grt_gr_id,
CASE WHEN meg_mty_id IN (18, 21) THEN 'Lengthmm'
WHEN meg_mty_id IN (19,22) THEN 'Weightg'
WHEN meg_mty_id IN (20,23) THEN 'Ageyear'
ELSE 'problem' END AS grt_tra_code,
meg_value AS grt_value,
NULL AS grt_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN meg_mty_id = 10 THEN 'Muscle_lipid_fatmeter'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see method

```

```

WHEN meg_method_sex = 1 THEN 'Gonadal_inspection'
WHEN meg_method_sex = 0 THEN 'Length_based_sex'
ELSE NULL END AS grt_trm_code,
meg_last_update AS grt_last_update,
meg_qal_id AS grt_qal_code,
CASE WHEN meg_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN meg_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN meg_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN meg_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN meg_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN meg_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS grt_ver_code
FROM datwgeel.t_metricgroupsamp_megsa megsa
JOIN datwgeel.t_groupsamp_grsa grsa ON megsa.meg_gr_id = grsa.gr_id
JOIN datwgeel.t_samplinginfo_sai AS tss ON grsa.sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text
LEFT JOIN mm ON megsa.meg_gr_id = mm.meg_gr_id -- joining subquery
JOIN (SELECT gr_id, gr_ser_id, gr_year, gr_lfs_code FROM dateel.t_group_gr WHERE gr_sex_code = 1)
ON (gr.gr_ser_id, COALESCE(gr.gr_year,1), gr_lfs_code) = ( tss2.ser_id, COALESCE(grsa.gr_year,1))
WHERE meg_mty_id IN (18,19,20); --544 Cdric pi fort

-- GROUP metrics FOR females    sampling
SELECT count(*) FROM datwgeel.t_metricgroupseries_megser megsa
WHERE meg_mty_id IN (21,22,23);--553

INSERT INTO dateel.t_grouptrait_grt
(grt_ser_id,
grt_wkg_code,
grt_spe_code,
grt_id,
grt_gr_id,
grt_tra_code,
grt_value,
grt_trv_code,
grt_trm_code,
grt_last_update,
grt_qal_code,
grt_ver_code)
-- extract method from table
-- this will extract 3 columns, meg_gr_id, meg_method_anguillicola and meg_method_sex
WITH a1 AS (
SELECT meg_gr_id,
      meg_value AS meg_method_sex
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 27

```

```

AND meg_value IS NOT NULL),
a2 AS (
  SELECT meg_gr_id,
         meg_value AS meg_method_anguillicola
FROM datwgeel.t_metricgroupsamp_megsa megsa
WHERE meg_mty_id = 28
AND meg_value IS NOT NULL),
mm AS (
  SELECT coalesce(a1.meg_gr_id, a2.meg_gr_id) AS meg_gr_id,
         meg_method_sex,
         meg_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.meg_gr_id = a2.meg_gr_id)
-- Insert select query
SELECT DISTINCT ON (meg_id)
tss2.ser_id AS grt_ser_id,
'WGEEL' AS grt_wkg_code,
'127186' AS grt_spe_code,
meg_id AS grt_id,
gr.gr_id AS grt_gr_id,
CASE WHEN meg_mty_id IN (18, 21) THEN 'Lengthmm'
WHEN meg_mty_id IN (19,22) THEN 'Weightg'
WHEN meg_mty_id IN (20,23) THEN 'Ageyear'
ELSE 'problem' END AS grt_tra_code,
meg_value AS grt_value,
NULL AS grt_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN meg_mty_id = 10 THEN 'Muscle_lipid_fatmeter'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see method
WHEN meg_method_sex = 1 THEN 'Gonadal_inspection'
WHEN meg_method_sex = 0 THEN 'Length_based_sex'
ELSE NULL END AS grt_trm_code,
meg_last_update AS grt_last_update,
meg_qal_id AS grt_qal_code,
CASE WHEN meg_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN meg_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN meg_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN meg_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN meg_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN meg_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS grt_ver_code
FROM datwgeel.t_metricgroupsamp_megsa megsa
JOIN datwgeel.t_groupsamp_grsa grsa ON megsa.meg_gr_id = grsa.gr_id
JOIN datwgeel.t_samplinginfo_sai AS tss ON grsa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text
LEFT JOIN mm ON megsa.meg_gr_id = mm.meg_gr_id -- joining subquery
JOIN (SELECT gr_id, gr_ser_id, gr_year, gr_lfs_code FROM dateel.t_group_gr WHERE gr_sex_code

```

```

ON (gr.gr_ser_id, COALESCE(gr.gr_year,1), gr_lfs_code) = ( tss2.ser_id, COALESCE(grsa.gr_yea
WHERE meg_mty_id IN (21,22,23); --1183

-- GROUP metrics for males series

INSERT INTO dateel.t_grouptrait_grt
(grt_ser_id,
grt_wkg_code,
grt_spe_code,
grt_id,
grt_gr_id,
grt_tra_code,
grt_value,
grt_trv_code,
grt_trm_code,
grt_last_update,
grt_qal_code,
grt_ver_code)
-- extract method from table
-- this will extract 3 columns, meg_gr_id, meg_method_anguillicola and meg_method_sex
WITH a1 AS (
SELECT meg_gr_id,
       meg_value AS meg_method_sex
FROM datwgeel.t_metricgroupseries_megser megser
WHERE meg_mty_id = 27
AND meg_value IS NOT NULL),
a2 AS (
SELECT meg_gr_id,
       meg_value AS meg_method_anguillicola
FROM datwgeel.t_metricgroupseries_megser megser
WHERE meg_mty_id = 28
AND meg_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.meg_gr_id, a2.meg_gr_id) AS meg_gr_id,
meg_method_sex,
meg_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.meg_gr_id = a2.meg_gr_id)
-- Insert select query
SELECT DISTINCT ON (meg_id)
tss2.ser_id AS grt_ser_id,
'WGEEL' AS grt_wkg_code,
'127186' AS grt_spe_code,
meg_id AS grt_id,
gr.gr_id AS grt_gr_id,
CASE WHEN meg_mty_id IN (18, 21) THEN 'Lengthmm'

```

```

WHEN meg_mty_id IN (19,22) THEN 'Weightg'
WHEN meg_mty_id IN (20,23) THEN 'Ageyear'
ELSE 'problem' END AS grt_tra_code,
meg_value AS grt_value,
NULL AS grt_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN meg_mty_id = 10 THEN 'Muscle_lipid_fatmeter'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see method
WHEN meg_method_sex = 1 THEN 'Gonadal_inspection'
WHEN meg_method_sex = 0 THEN 'Length_based_sex'
ELSE NULL END AS grt_trm_code,
meg_last_update AS grt_last_update,
meg_qal_id AS grt_qal_code,
CASE WHEN meg_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN meg_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN meg_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN meg_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN meg_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN meg_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS grt_ver_code
FROM datwgeel.t_metricgroupseries_megser megser
JOIN datwgeel.t_groupseries_grser grser ON megser.meg_gr_id = gr_id
JOIN datwgeel.t_series_ser AS tss ON grser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort
LEFT JOIN mm ON megser.meg_gr_id = mm.meg_gr_id -- joining subquery
JOIN (SELECT gr_id, gr_ser_id, gr_year, gr_lfs_code FROM dateel.t_group_gr WHERE gr_sex_code
ON (gr.gr_ser_id, COALESCE(gr.gr_year,1), gr_lfs_code) = ( tss2.ser_id, COALESCE(grser.gr_y
WHERE meg_mty_id IN (18,19,20); --409

-- GROUP metrics for females series
SELECT count(*) FROM datwgeel.t_metricgroupseries_megser megser
WHERE meg_mty_id IN (21,22,23);--553

INSERT INTO dateel.t_grouptrait_grt
(grt_ser_id,
grt_wkg_code,
grt_spe_code,
grt_id,
grt_gr_id,
grt_tra_code,
grt_value,
grt_trv_code,
grt_trm_code,
grt_last_update,
grt_qal_code,
grt_ver_code)

```

```

-- extract method from table
-- this will extract 3 columns, meg_gr_id, meg_method_anguillicola and meg_method_sex
WITH a1 AS (
SELECT meg_gr_id,
      meg_value AS meg_method_sex
FROM datwgeel.t_metricgroupseries_megser megser
WHERE meg_mty_id = 27
AND meg_value IS NOT NULL),
a2 AS (
SELECT meg_gr_id,
      meg_value AS meg_method_anguillicola
FROM datwgeel.t_metricgroupseries_megser megser
WHERE meg_mty_id = 28
AND meg_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.meg_gr_id, a2.meg_gr_id) AS meg_gr_id,
      meg_method_sex,
      meg_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.meg_gr_id = a2.meg_gr_id)
-- Insert select query
SELECT DISTINCT ON (meg_id)
tss2.ser_id AS grt_ser_id,
'WGEEL' AS grt_wkg_code,
'127186' AS grt_spe_code,
meg_id AS grt_id,
gr.gr_id AS grt_gr_id,
CASE WHEN meg_mty_id IN (18, 21) THEN 'Lengthmm'
WHEN meg_mty_id IN (19,22) THEN 'Weightg'
WHEN meg_mty_id IN (20,23) THEN 'Ageyear'
ELSE 'problem' END AS grt_tra_code,
meg_value AS grt_value,
NULL AS grt_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN meg_mty_id = 10 THEN 'Muscle_lipid_fatmeter'
WHEN meg_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see method
WHEN meg_method_sex = 1 THEN 'Gonadal_inspection'
WHEN meg_method_sex = 0 THEN 'Length_based_sex'
ELSE NULL END AS grt_trm_code,
meg_last_update AS grt_last_update,
meg_qal_id AS grt_qal_code,
CASE WHEN meg_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN meg_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN meg_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN meg_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN meg_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN meg_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'

```

```

ELSE 'WGEEL-2018-1' END AS grt_ver_code
FROM datwgeel.t_metricgroupseries_megser megser
JOIN datwgeel.t_groupseries_grser grser ON megser.meg_gr_id = gr_id
JOIN datwgeel.t_series_ser AS tss ON grser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort
LEFT JOIN mm ON megser.meg_gr_id = mm.meg_gr_id -- joining subquery
JOIN (SELECT gr_id, gr_ser_id, gr_year, gr_lfs_code FROM dateel.t_group_gr WHERE gr_sex_code
ON (gr.gr_ser_id, COALESCE(gr.gr_year,1), gr_lfs_code) = ( tss2.ser_id, COALESCE(grser.gr_ye
WHERE meg_mty_id IN (21,22,23); --553
;

```

Table 4.1: Table t_grouptrait_g

grt_ser_id	grt_wkg_code	grt_spe_code	grt_id	grt_gr_id	grt_tra
cbdf353c-2ce7-4d71-8d52-60548d94a3e8	WGEEL	126281	9581	3725	Weightg
dec9711e-04d6-4641-8162-8aa71b6edade	WGEEL	126281	14006	4248	Lengthm
a76757f8-4f3c-4513-a65e-82cae81c7063	WGEEL	126281	19591	5098	Weightg
5b7cd6eb-bb35-44d4-8487-f799df771078	WGEEL	126281	21684	5618	Ageyear
7eb93711-736d-442c-9f14-1d293e576394	WGEEL	126281	12488	3990	Muscle_l
7eb93711-736d-442c-9f14-1d293e576394	WGEEL	126281	5405	2708	Muscle_l
7eb93711-736d-442c-9f14-1d293e576394	WGEEL	126281	5394	2707	Muscle_l
7eb93711-736d-442c-9f14-1d293e576394	WGEEL	126281	5383	2706	Muscle_l
7eb93711-736d-442c-9f14-1d293e576394	WGEEL	126281	5372	2705	Muscle_l
7eb93711-736d-442c-9f14-1d293e576394	WGEEL	126281	5361	2704	Muscle_l
7eb93711-736d-442c-9f14-1d293e576394	WGEEL	126281	5350	2703	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	12477	3989	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	5324	2701	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	5274	2697	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	5252	2695	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	5226	2693	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	5203	2691	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	5177	2689	Muscle_l
0cd50424-0ba6-4b6c-b9fe-d9572ff9c424	WGEEL	126281	5151	2687	Muscle_l
ae1030ac-b099-41b6-80bf-c3a7c455d17f	WGEEL	126281	12466	3988	Muscle_l

Chapter 5

Creating ind metrics

5.1 Fish table

SQL code to create tables `dat.t_fish_fi` and `dateel.t_fish_fi`

```
-- DROP TABLE IF EXISTS dat.t_fish_fi CASCADE;

CREATE TABLE dat.t_fish_fi (
  fi_id serial4 NOT NULL,
  fi_ser_id UUID NOT NULL,
  CONSTRAINT fk_fi_ser_id FOREIGN KEY (fi_ser_id)
  REFERENCES dat.t_series_ser (ser_id)
  ON UPDATE CASCADE ON DELETE CASCADE,
  fi_wkg_code TEXT NOT NULL,
  CONSTRAINT fk_fi_wkg_code FOREIGN KEY (fi_wkg_code)
  REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT t_fish_fi_pkey PRIMARY KEY (fi_id, fi_wkg_code),
  fi_spe_code TEXT,
  CONSTRAINT fk_fi_spe_code FOREIGN KEY (fi_spe_code)
  REFERENCES ref.tr_species_spe(spe_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  fi_lfs_code TEXT,
  CONSTRAINT fk_fi_lfs_code_fi_spe_code FOREIGN KEY (fi_lfs_code, fi_spe_code)
  REFERENCES ref.tr_lifestage_lfs (lfs_code, lfs_spe_code)
  ON UPDATE CASCADE ON DELETE RESTRICT,
  fi_date date NULL,
  fi_year int4 NULL,
  CONSTRAINT ck_fi_date_fi_year CHECK (((fi_date IS NOT NULL) OR (fi_year IS NOT NULL))),
  fi_comment text NULL,
```

```

fi_lastupdate date DEFAULT CURRENT_DATE NOT NULL,
fi_idsource TEXT NULL UNIQUE,
fi_ver_code TEXT NOT NULL,
CONSTRAINT fk_gr_ver_code FOREIGN KEY (fi_ver_code)
REFERENCES ref.tr_version_ver(ver_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
fi_x_4326 NUMERIC,
fi_y_4326 NUMERIC,
fi_geom GEOMETRY(Point, 4326)
);

COMMENT ON TABLE dat.t_fish_fi IS 'Table identifying the fish metrics, a fish metric corresponds to a
fish sampled at a given date or year.';

COMMENT ON COLUMN dat.t_fish_fi.fi_id IS 'Fi ID, serial primary key on fi_id and fi_wkg_code';
COMMENT ON COLUMN dat.t_fish_fi.fi_wkg_code IS 'Code of the working group, one of
WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dat.t_fish_fi.fi_lfs_code IS 'Life stage code';
COMMENT ON COLUMN dat.t_fish_fi.fi_spe_code IS 'Species code';
COMMENT ON COLUMN dat.t_fish_fi.fi_year IS 'The year';
COMMENT ON COLUMN dat.t_fish_fi.fi_comment IS 'Comment on the fish';
COMMENT ON COLUMN dat.t_fish_fi.fi_lastupdate IS 'Last update, inserted automatically';
COMMENT ON COLUMN dat.t_fish_fi.fi_ver_code IS 'Version code as in tr_version_ver, corresponds to a
specific version of the metric';
COMMENT ON COLUMN dat.t_fish_fi.fi_idsource IS 'Identifier of the fish in the source (country, date, etc)';

GRANT ALL ON dat.t_fish_fi TO diaspara_admin;
GRANT SELECT ON dat.t_fish_fi TO diaspara_read;

DROP TABLE IF EXISTS dateel.t_fish_fi;
CREATE TABLE dateel.t_fish_fi (
CONSTRAINT fk_fi_ser_id FOREIGN KEY (fi_ser_id)
REFERENCES dateel.t_series_ser (ser_id)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT fk_fi_wkg_code FOREIGN KEY (fi_wkg_code)
REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT t_fish_fi_pkey PRIMARY KEY (fi_id, fi_wkg_code),
CONSTRAINT fk_fi_spe_code FOREIGN KEY (fi_spe_code)
REFERENCES ref.tr_species_spe(spe_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_fi_lfs_code_fi_spe_code FOREIGN KEY (fi_lfs_code, fi_spe_code)
REFERENCES ref.tr_lifestage_lfs (lfs_code, lfs_spe_code)
);

```

```

    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT ck_fi_date_fi_year CHECK (((fi_date IS NOT NULL) OR (fi_year IS NOT NULL))),
    CONSTRAINT fk_gr_ver_code FOREIGN KEY (fi_ver_code)
    REFERENCES refeel.tr_version_ver(ver_code)
    ON UPDATE CASCADE ON DELETE RESTRICT
) INHERITS (dat.t_fish_fi);

GRANT ALL ON dateel.t_fish_fi TO diaspara_admin;
GRANT SELECT ON dateel.t_fish_fi TO diaspara_read;

-- Change to adapt to DIASPARA salmon dataset WP2
-- ALTER TABLE dat.t_fish_fish ADD COLUMN fi_tagnumber TEXT DEFAULT NULL;

```

SQL code to import data tables

```

-- insert from series
INSERT INTO dateel.t_fish_fi
(fi_id,
fi_ser_id,
fi_wkg_code,
fi_spe_code,
fi_lfs_code,
fi_date,
fi_year,
fi_comment,
fi_lastupdate,
fi_idsource,
fi_ver_code,
fi_x_4326,
fi_y_4326,
fi_geom)
SELECT
fi_id,
tss2.ser_id AS fi_ser_id,
'WGEEL' AS fi_wkg_code,
'127186' AS fi_spe_code,
fi_lfs_code,
fi_date,
fi_year,
fi_comment,
fi_lastupdate,

```

```

fi_id_cou AS fi_idsource,
CASE WHEN fi_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN fi_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN fi_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN fi_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN fi_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN fi_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS fi_ver_code,
st_x(tss.geom),
st_y(tss.geom),
tss.geom
FROM datwgeel.t_fishseries_fiser
JOIN datwgeel.t_series_ser tss ON ser_id = fiser_ser_id
JOIN dat.t_series_ser tss2 ON ser_code = ser_nameshort ; --757787

-- Insert from sampling

INSERT INTO dateel.t_fish_fi
(fi_id,
fi_ser_id,
fi_wkg_code,
fi_spe_code,
fi_lfs_code,
fi_date,
fi_year,
fi_comment,
fi_lastupdate,
fi_idsource,
fi_ver_code,
fi_x_4326,
fi_y_4326,
fi_geom)
SELECT
fi_id,
tss2.ser_id AS fi_ser_id,
'WGEEL' AS fi_wkg_code,
'127186' AS fi_spe_code,
CASE WHEN fi_lfs_code = 'NA' THEN 'YS' ELSE fi_lfs_code END AS fi_lfs_code,
fi_date,
fi_year,
fi_comment,
fi_lastupdate,
fi_id_cou AS fi_idsource,
CASE WHEN fi_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN fi_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'

```

```

        WHEN fi_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
        WHEN fi_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
        WHEN fi_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
        WHEN fi_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
        ELSE 'WGEEL-2018-1' END AS fi_ver_code,
fisa_x_4326,
fisa_y_4326,
fisa_geom
FROM datwgeel.t_fishsamp_fisa
JOIN datwgeel.t_samplinginfo_sai AS tss ON fisa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::text; --98545

```

Table 5.1: Table

fi_id	fi_ser_id	fi_wkg_code	fi_spe_code	fi_lfs_code	fi_date
897144	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2012-09-2
897143	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2012-09-2
897142	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2012-09-2
897141	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2012-09-2
897140	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2012-09-2
897139	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2012-09-2
897138	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2012-09-2
3419021	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-04-2
3419020	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-04-2
3419024	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-04-2
3419023	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-04-2
3419022	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-04-2
3419279	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2
3419278	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2
3419277	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2
3419276	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2
3419275	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2
3419274	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2
3419273	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2
3419272	32870779-52d8-4326-b473-43095b95625a	WGEEL	126281	Y	2022-10-2

5.2 Table t_indtrait_int

SQL code to create tables `dat.t_indivtrait_int` and `dateel.t_indivtrait_int`

```
-- DROP TABLE IF EXISTS dat.t_indivtrait_int CASCADE;
```

```

CREATE TABLE dat.t_indivtrait_int (
  int_ser_id uuid,
  CONSTRAINT fk_int_ser_id FOREIGN KEY (int_ser_id)
    REFERENCES dat.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  int_wkg_code TEXT NOT NULL,
  CONSTRAINT fk_int_wkg_code FOREIGN KEY (int_wkg_code)
    REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  int_spe_code TEXT NOT NULL,
  CONSTRAINT fk_int_spe_code FOREIGN KEY (int_spe_code)
    REFERENCES ref.tr_species_spe(spe_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  int_id serial4 NOT NULL,
  int_fi_id int4 NOT NULL,
  CONSTRAINT fk_int_fi_id FOREIGN KEY (int_fi_id, int_wkg_code)
    REFERENCES dat.t_fish_fi(fi_id,fi_wkg_code)
    ON UPDATE CASCADE ON DELETE CASCADE,
  int_tra_code TEXT NOT NULL,
  CONSTRAINT fk_int_tra_code FOREIGN KEY (int_tra_code)
    REFERENCES ref.tr_trait_tra(tra_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  int_value numeric NULL,
  int_trv_code TEXT,
  CONSTRAINT fk_int_trv_tra_code FOREIGN KEY (int_trv_code, int_tra_code,int_wkg_code)
    REFERENCES ref.tr_traitvaluequal_trv(trv_code, trv_trq_code, trv_wkg_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  int_trm_code TEXT,
  CONSTRAINT fk_int_trm_code FOREIGN KEY (int_trm_code)
    REFERENCES ref.tr_traitmethod_trm(trm_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  int_last_update date DEFAULT CURRENT_DATE NOT NULL,
  int_qal_code int4 NULL,
  CONSTRAINT fk_int_qal_id FOREIGN KEY (int_qal_code)
    REFERENCES ref.tr_quality_qal(qal_code) ON UPDATE CASCADE,
  int_ver_code TEXT NOT NULL,
  CONSTRAINT fk_int_ver_code FOREIGN KEY (int_ver_code)
    REFERENCES ref.tr_version_ver(ver_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT uk_int_fi UNIQUE (int_fi_id, int_trm_code, int_wkg_code),
  CONSTRAINT t_indivtrait_int_pkey PRIMARY KEY (int_id, int_wkg_code),
  CONSTRAINT ck_qualitative_or_numeric CHECK
  (
    (int_value IS NULL AND int_trv_code IS NOT NULL) OR
    (int_value IS NOT NULL AND int_trv_code IS NULL)
  )
);

```

```

    )
);
CREATE INDEX dat_t_indivtrait_int_idx ON dat.t_indivtrait_int USING btree (int_fi_id);

GRANT ALL ON dat.t_indivtrait_int TO diaspara_admin;
GRANT SELECT ON dat.t_indivtrait_int TO diaspara_read;

DROP TABLE IF EXISTS dateel.t_indivtrait_int;
CREATE TABLE dateel.t_indivtrait_int (
    CONSTRAINT fk_int_ser_id FOREIGN KEY (int_ser_id)
        REFERENCES dateel.t_series_ser (ser_id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_int_wkg_code FOREIGN KEY (int_wkg_code)
        REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT fk_int_spe_code FOREIGN KEY (int_spe_code)
        REFERENCES ref.tr_species_spe(spe_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT fk_int_fi_id FOREIGN KEY (int_fi_id, int_wkg_code)
        REFERENCES dateel.t_fish_fi(fi_id,fi_wkg_code)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_int_tra_code FOREIGN KEY (int_tra_code)
        REFERENCES refeel.tg_trait_tra(tra_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT fk_int_trv_tra_code FOREIGN KEY (int_trv_code, int_tra_code)
        -- unlike in dat.t_indivtrait_int this one does not take ref TO wkgcode, no need AS we are
        REFERENCES refeel.tr_traitvaluequal_trv(trv_code, trv_trq_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT fk_int_trm_code FOREIGN KEY (int_trm_code)
        REFERENCES refeel.tr_traitmethod_trm(trm_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT fk_int_qal_id FOREIGN KEY (int_qal_code)
        REFERENCES ref.tr_quality_qal(qal_code) ON UPDATE CASCADE,
    CONSTRAINT fk_int_ver_code FOREIGN KEY (int_ver_code)
        REFERENCES refeel.tr_version_ver(ver_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT uk_trv_fi_id_trv_tra_code UNIQUE(int_fi_id,int_tra_code)
) INHERITS (dat.t_indivtrait_int);
CREATE INDEX dateel_t_indivtrait_int_idx ON dateel.t_indivtrait_int USING btree (int_fi_id);

GRANT ALL ON dateel.t_indivtrait_int TO diaspara_admin;
GRANT SELECT ON dateel.t_indivtrait_int TO diaspara_read;

COMMENT ON TABLE dateel.t_indivtrait_int IS

```

```

'Table joining fish and traits';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_ser_id IS
'Series UUID';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_wkg_code IS
'Working ind on of WGEEL, WGNAS, WGBAST ...';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_spe_code IS
'Species code here ''127186''';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_id IS
'ID, integer, unique for wkg_code';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_fi_id IS
'ID of the fish';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_tra_code IS
'Code of the trait, e.g. Lengthmm';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_value IS
'Value for numeric';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_trv_code IS
'Value for qualitative see refeel.tr_traitvaluequal_trv';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_trm_code IS
'Method see refeel.tr_traimethod_trm';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_last_update IS 'date last update';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_qal_code IS
'Quality code references ref.tr_quality_qal';
COMMENT ON COLUMN dateel.t_indivtrait_int.int_ver_code IS
'version e.g. WGEEL_2024_1';

-- TODO trigger on date

```

SQL code to import individual metric tables

```

/*
 * Insert numeric for indiv - series
 */
DELETE FROM dateel.t_indivtrait_int;

INSERT INTO dateel.t_indivtrait_int
(int_ser_id,
int_wkg_code,
int_spe_code,
int_id,
int_fi_id,
int_tra_code,
int_value,
int_trv_code,
int_trm_code,
int_last_update,

```

```

int_qal_code,
int_ver_code)
WITH a1 AS (
SELECT mei-fi-id,
      mei_value AS mei_method_sex
FROM datwgeel.t_metricindsamp_meisa meisa
WHERE mei_mty_id = 27
AND mei_value IS NOT NULL),
a2 AS (
SELECT mei-fi-id,
      mei_value AS mei_method_anguillicola
FROM datwgeel.t_metricindsamp_meisa meisa
WHERE mei_mty_id = 28
AND mei_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.mei-fi-id, a2.mei-fi-id) AS mei-fi-id,
      mei_method_sex,
      mei_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.mei-fi-id = a2.mei-fi-id)
SELECT
tss2.ser_id AS int_ser_id,
'WGEEL' AS int_wkg_code,
'127186' AS int_spe_code,
fi-id AS int_id,
meiser.mei-fi-id AS int-fi-id,
CASE WHEN mei_mty_id = 1 THEN 'Lengthmm'
WHEN mei_mty_id = 2 THEN 'Weightg'
WHEN mei_mty_id = 3 THEN 'Ageyear'
WHEN mei_mty_id = 4 THEN 'Eye_diam_meanmm'
WHEN mei_mty_id = 5 THEN 'Pectoral_lengthmm'
WHEN mei_mty_id = 9 THEN 'Anguillicola_intensity'
WHEN mei_mty_id = 11 THEN 'Muscle_lipid' -- IF fatmeter insert later only if gravimeter does
WHEN mei_mty_id = 12 THEN 'Sum_6_pcb'
WHEN mei_mty_id = 15 THEN 'Pb'
WHEN mei_mty_id = 16 THEN 'Hg'
WHEN mei_mty_id = 17 THEN 'Cd'
WHEN mei_mty_id = 26 THEN 'Teq'
ELSE 'problem' END AS int_tra_code,
mei_value AS int_value,
NULL AS int_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN mei_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see METHOD
WHEN mei_mty_id = 9 AND mei_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN mei_mty_id = 9 AND mei_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
ELSE NULL END AS int_trm_code,
mei_last_update AS int_last_update,

```

```

mei_qal_id AS int_qal_code,
CASE WHEN mei_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN mei_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN mei_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN mei_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN mei_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN mei_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS int_ver_code
FROM datwgeel.t_metricindseries_meiser meiser
JOIN datwgeel.t_fishseries_fiser fiser ON meiser.mei-fi_id = fi_id
JOIN datwgeel.t_series_ser AS tss ON fiser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort
LEFT JOIN mm ON meiser.mei-fi_id = mm.mei-fi_id -- joining subquery
WHERE mei_mty_id IN (1,2,3,4,5,9,11,12,15,16,17,26); --1145525

--10 Muscle lipid fatmeter

INSERT INTO dateel.t_indivtrait_int
(int_ser_id,
int_wkg_code,
int_spe_code,
int_id,
int-fi_id,
int_tra_code,
int_value,
int_trv_code,
int_trm_code,
int_last_update,
int_qal_code,
int_ver_code)
WITH fatmeter AS (
SELECT
tss2.ser_id AS int_ser_id,
'WGEEL' AS int_wkg_code,
'127186' AS int_spe_code,
fi_id AS int_id,
meiser.mei-fi_id AS int-fi_id,
CASE WHEN mei_mty_id = 10 THEN 'Muscle_lipid'
ELSE 'problem' END AS int_tra_code,
mei_value AS int_value,
NULL AS int_trv_code,

```

```

CASE WHEN mei_mty_id = 10 THEN 'Muscle_lipid_fatmeter' -- different method see method
ELSE NULL END AS int_trm_code,
mei_last_update AS int_last_update,
mei_qal_id AS int_qal_code,
CASE WHEN mei_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN mei_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN mei_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN mei_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN mei_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN mei_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS int_ver_code
FROM datwgeel.t_metricindseries_meiser meiser
JOIN datwgeel.t_fishseries_fiser fiser ON meiser.mei-fi_id = fi_id
JOIN datwgeel.t_series_ser AS tss ON fiser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort
WHERE mei_mty_id IN (10))
SELECT * FROM fatmeter WHERE int-fi_id NOT IN
(SELECT int-fi_id FROM dateel.t_indivtrait_int WHERE int_tra_code = 'Muscle_lipid')
; --0

/*
 * Insert numeric for - sampling
 * Some values have both gravimeter and fatmeter, gravimeter chosen.
 */

INSERT INTO dateel.t_indivtrait_int
(int_ser_id,
int_wkg_code,
int_spe_code,
int_id,
int-fi_id,
int_tra_code,
int_value,
int_trv_code,
int_trm_code,
int_last_update,
int_qal_code,
int_ver_code)
WITH a1 AS (
SELECT mei-fi_id,
      mei_value AS mei_method_sex
FROM datwgeel.t_metricindsamp_meisa meisa
WHERE mei_mty_id = 27
AND mei_value IS NOT NULL),

```

```

a2 AS (
  SELECT mei-fi_id,
         mei_value AS mei_method_anguillicola
  FROM datwgeel.t_metricindsamp_meisa meisa
  WHERE mei_mty_id = 28
  AND mei_value IS NOT NULL),
mm AS (
  SELECT coalesce(a1.mei-fi_id, a2.mei-fi_id) AS mei-fi_id,
         mei_method_sex,
         mei_method_anguillicola
  FROM a1 FULL OUTER JOIN a2 ON a1.mei-fi_id = a2.mei-fi_id)
SELECT
  tss2.ser_id AS int_ser_id,
  'WGEEL' AS int_wkg_code,
  '127186' AS int_spe_code,
  mei_id AS int_id,
  meisa.mei-fi_id AS int-fi_id,
  CASE WHEN mei_mty_id = 1 THEN 'Lengthmm'
  WHEN mei_mty_id = 2 THEN 'Weightg'
  WHEN mei_mty_id = 3 THEN 'Ageyear'
  WHEN mei_mty_id = 4 THEN 'Eye_diam_meanmm'
  WHEN mei_mty_id = 5 THEN 'Pectoral_lengthmm'
  WHEN mei_mty_id = 9 THEN 'Anguillicola_intensity'
  WHEN mei_mty_id = 11 THEN 'Muscle_lipid' -- IF fatmeter insert later only if gravimeter does
  WHEN mei_mty_id = 12 THEN 'Sum_6_pcb'
  WHEN mei_mty_id = 15 THEN 'Pb'
  WHEN mei_mty_id = 16 THEN 'Hg'
  WHEN mei_mty_id = 17 THEN 'Cd'
  WHEN mei_mty_id = 26 THEN 'Teq'
  ELSE 'problem' END AS int_tra_code,
  mei_value AS int_value,
  NULL AS int_trv_code,
  CASE WHEN mei_mty_id = 11 THEN 'Muscle_lipid_gravimeter' -- different method see METHOD
  WHEN mei_mty_id = 9 AND mei_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
  WHEN mei_mty_id = 9 AND mei_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
  ELSE NULL END AS int_trm_code,
  mei_last_update AS int_last_update,
  mei_qal_id AS int_qal_code,
  CASE WHEN mei_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
  WHEN mei_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
  WHEN mei_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
  WHEN mei_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
  WHEN mei_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
  WHEN mei_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
  ELSE 'WGEEL-2018-1' END AS int_ver_code

```

```

FROM datwgeel.t_metricindsamp_meisa meisa
JOIN datwgeel.t_fishsamp_fisa ON meisa.mei-fi-id = fi-id
JOIN datwgeel.t_samplinginfo_sai AS tss ON fisa-sai-id = sai-id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai-id::TEXT
LEFT JOIN mm ON meisa.mei-fi-id = mm.mei-fi-id -- joining subquery
WHERE mei_mty-id IN (1,2,3,4,5,9,11,12,15,16,17,26); --298005

-- Insert only fatmeter where gravimeter does not exist

INSERT INTO dateel.t_indivtrait_int
(int_ser_id,
int_wkg_code,
int_spe_code,
int_id,
int-fi-id,
int_tra_code,
int_value,
int_trv_code,
int_trm_code,
int_last_update,
int_qal_code,
int_ver_code)
WITH fatmeter AS (
SELECT
tss2.ser_id AS int_ser_id,
'WGEEL' AS int_wkg_code,
'127186' AS int_spe_code,
mei-id AS int_id,
meisa.mei-fi-id AS int-fi-id,
CASE WHEN mei_mty-id = 10 THEN 'Muscle_lipid'
ELSE 'problem' END AS int_tra_code,
mei_value AS int_value,
NULL AS int_trv_code, -- there ARE NO qualitative VALUES FOR GROUP metrics
CASE WHEN mei_mty-id = 10 THEN 'Muscle_lipid_fatmeter' -- different method see method
ELSE NULL END AS int_trm_code,
mei_last_update AS int_last_update,
mei_qal-id AS int_qal_code,
CASE WHEN mei_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
WHEN mei_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
WHEN mei_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
WHEN mei_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
WHEN mei_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
WHEN mei_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
ELSE 'WGEEL-2018-1' END AS int_ver_code

```

```

FROM datwgeel.t_metricindsamp_meisa meisa
JOIN datwgeel.t_fishsamp_fisa ON meisa.mei-fi-id = fi-id
JOIN datwgeel.t_samplinginfo_sai AS tss ON fisa-sai-id = sai-id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai-id::text
WHERE mei_mty-id = 10)
SELECT * FROM fatmeter WHERE int-fi-id NOT IN
(SELECT int-fi-id FROM dateel.t_indivtrait_int WHERE int-tra_code = 'Muscle_lipid')
; --100

-- Qualitative traits series

INSERT INTO dateel.t_indivtrait_int
(int_ser-id,
int_wkg_code,
int_spe_code,
int-id,
int-fi-id,
int-tra_code,
int_value,
int_trv_code,
int_trm_code,
int_last-update,
int_qal_code,
int_ver_code)
WITH a1 AS (
SELECT mei-fi-id,
      mei_value AS mei_method-sex
FROM datwgeel.t_metricindseries_meiser
WHERE mei_mty-id = 27
AND mei_value IS NOT NULL),
a2 AS (
SELECT mei-fi-id,
      mei_value AS mei_method-anguillicola
FROM datwgeel.t_metricindseries_meiser
WHERE mei_mty-id = 28
AND mei_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.mei-fi-id, a2.mei-fi-id) AS mei-fi-id,
mei_method-sex,
mei_method-anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.mei-fi-id = a2.mei-fi-id)
SELECT

```

```

tss2.ser_id AS int_ser_id,
'WGEEL' AS int_wkg_code,
'127186' AS int_spe_code,
fi_id AS int_id,
meiser.mei_fi_id AS int_fi_id,
CASE
WHEN mei_mty_id = 6 THEN 'Sex'
WHEN mei_mty_id = 7 THEN 'Is_differentiated'
WHEN mei_mty_id = 8 THEN 'Anguillicola_presence'
WHEN mei_mty_id = 13 THEN 'Evex_presence'
WHEN mei_mty_id = 14 THEN 'Hva_presence'
ELSE 'problem' END AS int_tra_code,
NULL AS int_value,
CASE
WHEN mei_mty_id = 6 AND mei_value = 1 THEN 'F'
WHEN mei_mty_id = 6 AND mei_value = 0 THEN 'M'
WHEN mei_mty_id = 7 AND mei_value = 1 THEN 'Y'
WHEN mei_mty_id = 7 AND mei_value = 0 THEN 'N'
WHEN mei_mty_id = 8 AND mei_value = 1 THEN 'Y'
WHEN mei_mty_id = 8 AND mei_value = 0 THEN 'N'
WHEN mei_mty_id = 13 AND mei_value= 1 THEN 'Y'
WHEN mei_mty_id = 13 AND mei_value= 0 THEN 'N'
WHEN mei_mty_id = 14 AND mei_value= 1 THEN 'Y'
WHEN mei_mty_id = 14 AND mei_value= 0 THEN 'N'
ELSE NULL END AS int_trv_code,
CASE
WHEN mei_mty_id = 6 AND mei_method_sex = 1 THEN 'Gonadal_inspection'
WHEN mei_mty_id = 6 AND mei_method_sex = 0 THEN 'Length_based_sex'
WHEN mei_mty_id = 8 AND mei_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN mei_mty_id = 8 AND mei_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
ELSE NULL END AS int_trm_code,
mei_last_update AS int_last_update,
mei_qal_id AS int_qal_code,
CASE WHEN mei_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN mei_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN mei_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN mei_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN mei_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN mei_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS int_ver_code
FROM datwgeel.t_metricindseries_meiser meiser
JOIN datwgeel.t_fishseries_fiser fiser ON meiser.mei_fi_id = fi_id
JOIN datwgeel.t_series_ser AS tss ON fiser_ser_id = ser_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= ser_nameshort
LEFT JOIN mm ON meiser.mei_fi_id = mm.mei_fi_id -- joining subquery

```

```

WHERE mei_mty_id IN (6,7,8,13,14); --189327 (pigment stage is not yet in the db)

-- Qualitative trait sampling

INSERT INTO dateel.t_indivtrait_int
(int_ser_id,
int_wkg_code,
int_spe_code,
int_id,
int-fi_id,
int_tra_code,
int_value,
int_trv_code,
int_trm_code,
int_last_update,
int_qal_code,
int_ver_code)
-- extract method from table
-- this will extract 3 columns, mei-fi_id, mei_method_anguillicola and mei_method_sex
WITH a1 AS (
SELECT mei-fi_id,
       mei_value AS mei_method_sex
FROM datwgeel.t_metricindsamp_meisa meisa
WHERE mei_mty_id = 27
AND mei_value IS NOT NULL),
a2 AS (
SELECT mei-fi_id,
       mei_value AS mei_method_anguillicola
FROM datwgeel.t_metricindsamp_meisa meisa
WHERE mei_mty_id = 28
AND mei_value IS NOT NULL),
mm AS (
SELECT coalesce(a1.mei-fi_id, a2.mei-fi_id) AS mei-fi_id,
       mei_method_sex,
       mei_method_anguillicola
FROM a1 FULL OUTER JOIN a2 ON a1.mei-fi_id = a2.mei-fi_id)
-- Insert select query
SELECT
tss2.ser_id AS int_ser_id,
'WGEEL' AS int_wkg_code,
'127186' AS int_spe_code,
mei_id AS int_id,
meisa.mei-fi_id AS int-fi_id,
CASE

```

```

WHEN mei_mty_id = 6 THEN 'Sex'
WHEN mei_mty_id = 7 THEN 'Is_differentiated'
WHEN mei_mty_id = 8 THEN 'Anguillicola_presence'
WHEN mei_mty_id = 13 THEN 'Evex_presence'
WHEN mei_mty_id = 14 THEN 'Hva_presence'
ELSE 'problem' END AS int_tra_code,
NULL AS int_value,
CASE
WHEN mei_mty_id = 6 AND mei_value = 1 THEN 'F'
WHEN mei_mty_id = 6 AND mei_value = 0 THEN 'M'
WHEN mei_mty_id = 7 AND mei_value = 1 THEN 'Y'
WHEN mei_mty_id = 7 AND mei_value = 0 THEN 'N'
WHEN mei_mty_id = 8 AND mei_value = 1 THEN 'Y'
WHEN mei_mty_id = 8 AND mei_value = 0 THEN 'N'
WHEN mei_mty_id = 13 AND mei_value= 1 THEN 'Y'
WHEN mei_mty_id = 13 AND mei_value= 0 THEN 'N'
WHEN mei_mty_id = 14 AND mei_value= 1 THEN 'Y'
WHEN mei_mty_id = 14 AND mei_value= 0 THEN 'N'
ELSE NULL END AS int_trv_code,
CASE
WHEN mei_mty_id = 6 AND mei_method_sex = 1 THEN 'Gonadal_inspection'
WHEN mei_mty_id = 6 AND mei_method_sex = 0 THEN 'Length_based_sex'
WHEN mei_mty_id = 8 AND mei_method_anguillicola = 1 THEN 'Anguillicola_stereomicroscope_count'
WHEN mei_mty_id = 8 AND mei_method_anguillicola = 0 THEN 'Anguillicola_visual_count'
ELSE NULL END AS int_trm_code,
mei_last_update AS int_last_update,
mei_qal_id AS int_qal_code,
CASE WHEN mei_dts_datasource = 'dc_2019' THEN 'WGEEL-2019-1'
      WHEN mei_dts_datasource = 'dc_2020' THEN 'WGEEL-2020-1'
      WHEN mei_dts_datasource = 'dc_2021' THEN 'WGEEL-2021-1'
      WHEN mei_dts_datasource = 'dc_2022' THEN 'WGEEL-2022-1'
      WHEN mei_dts_datasource = 'dc_2023' THEN 'WGEEL-2023-1'
      WHEN mei_dts_datasource = 'dc_2024' THEN 'WGEEL-2024-1'
      ELSE 'WGEEL-2018-1' END AS int_ver_code
FROM datwgeel.t_metricindsamp_meisa meisa
JOIN datwgeel.t_fishsamp_fisa ON meisa.mei-fi_id = fi_id
JOIN datwgeel.t_samplinginfo_sai AS tss ON fisa_sai_id = sai_id
JOIN dateel.t_series_ser AS tss2 ON ser_code= sai_id::TEXT
LEFT JOIN mm ON meisa.mei-fi_id = mm.mei-fi_id -- joining subquery
WHERE mei_mty_id IN (6,7,8,13,14); --163914 (pigment stage is not yet in the db)

```

Table 5.2: Table t_indiv

int_ser_id	int_wkg_code	int_spe_code	int_id	int_fi_id	int_tra_
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413553	3413553	Pectoral_
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413553	3413553	Eye_diar
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413553	3413553	Weightg
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413553	3413553	Lengthm
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413535	3413535	Pectoral_
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413535	3413535	Eye_diar
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413535	3413535	Weightg
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413535	3413535	Lengthm
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413502	3413502	Pectoral_
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413502	3413502	Eye_diar
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413502	3413502	Weightg
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413502	3413502	Lengthm
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413498	3413498	Pectoral_
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413498	3413498	Eye_diar
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413498	3413498	Weightg
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413498	3413498	Lengthm
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413562	3413562	Pectoral_
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413562	3413562	Eye_diar
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413562	3413562	Weightg
0424851e-9533-4e08-91c8-239e05dd5b73	WGEEL	126281	3413562	3413562	Lengthm

[TODO] Link the ccm wso_id to the hydrosched db identifiers of the basins to get the best fit from tr_area_are. git issue #25 Import data from WGEEL](https://github.com/DIASPARAproject/WP3_migdb/issues/23)

The release date for that one is :

[Milestone release alpha](#)

So the metric release will be after wgeel, the work will be finished and this can be presented during wgeel.

Chapter 6

Import data from WP2 (TODO)

Chapter 7

WGBAST

7.1 series table t_series_ser

7.1.1 Creating inherited series

SQL code to create table `datbast.t_series_ser`

```
-- when dumping this is inherited => I have a not null constraint on the server
-- when the table is recreated

DROP TABLE IF EXISTS datbast.t_series_ser;
CREATE TABLE datbast.t_series_ser (
  CONSTRAINT pk_ser_id PRIMARY KEY (ser_id),
  CONSTRAINT uk_ser_code UNIQUE (ser_code),
  CONSTRAINT uk_ser_name UNIQUE (ser_name),
  CONSTRAINT fk_ser_spe_code FOREIGN KEY (ser_spe_code)
    REFERENCES ref.tr_species_spe(spe_code)
    ON UPDATE CASCADE ON DELETE RESTRICT ,
  CONSTRAINT fk_ser_lfs_code_ser_spe_code FOREIGN KEY (ser_lfs_code, ser_spe_code)
    REFERENCES ref.tr_lifestage_lfs (lfs_code, lfs_spe_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_ser_are_code FOREIGN KEY (ser_are_code)
    REFERENCES refbast.tr_area_are (are_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT fk_ser_wkg_code FOREIGN KEY (ser_wkg_code)
    REFERENCES ref.tr_icworkinggroup_wkg(wkg_code)
    ON UPDATE CASCADE ON DELETE RESTRICT,
```

```

CONSTRAINT fk_ser_ver_code FOREIGN KEY (ser_ver_code)
  REFERENCES refbast.tr_version_ver(ver_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_cou_code FOREIGN KEY (ser_cou_code)
  REFERENCES ref.tr_country_cou(cou_code)
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_uni_code FOREIGN KEY (ser_uni_code)
  REFERENCES ref.tr_units_uni(uni_code)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_ser_effort_uni_code FOREIGN KEY (ser_effort_uni_code)
  REFERENCES ref.tr_units_uni(uni_code)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_ser_wltyp_code FOREIGN KEY( ser_wltyp_code)
  REFERENCES ref."WLTY"("Key")
ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT fk_ser_gea_code FOREIGN KEY (ser_gea_code)
  REFERENCES ref.tr_gear_gea(gea_code)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT fk_ser_fiw_code FOREIGN KEY (ser_fiw_code)
  REFERENCES ref.tr_fishway_fiw(fiw_code)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT fk_ser_mon_code FOREIGN KEY (ser_mon_code)
  REFERENCES ref.tr_monitoring_mon(mon_code)
ON UPDATE CASCADE ON DELETE CASCADE
) inherits (dat.t_series_ser);

-- In the wgeel schema the default is WGEEL
ALTER TABLE datbast.t_series_ser ALTER COLUMN ser_wkg_code SET DEFAULT 'WGBAST';

COMMENT ON TABLE datbast.t_series_ser IS
  'Table of time series, or sampling data identifier. This corresponds to a multi-annual data
  It can correspond to time series data or individual metrics collection or both. This table i
COMMENT ON COLUMN dat.t_series_ser.ser_id IS
  'UUID, identifier of the series, primary key';
COMMENT ON COLUMN dat.t_series_ser.ser_code IS
  'Code of the series, use country_code + name series (4 letters) + stage';
COMMENT ON COLUMN dat.t_series_ser.ser_name IS
  'Name of the series';
COMMENT ON COLUMN dat.t_series_ser.ser_spe_code IS
  'Species, one of use aphiaID eg ''127186'' for Salmon and ''127187''for Trutta. references
  it should correspond to the main species target by the sampling';
COMMENT ON COLUMN dat.t_series_ser.ser_lfs_code IS
  'Life stage see tr_lifestage_lfs, Code of the lifestage see tr_lifestage_lfs, the constrain
  both lfs_code, and lfs_spe_code (as two species can have the same lifestage code. The lifest
COMMENT ON COLUMN dat.t_series_ser.ser_wkg_code IS

```

```

'Code of the working group, one of WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN dat.t_series_ser.ser_ver_code IS
'Version code from ref.tr_version_ver the data call e.g. WGNAS_2020_1, WGEEL_2016_1';
COMMENT ON COLUMN dat.t_series_ser.ser_cou_code IS
'Code of the country';
COMMENT ON COLUMN dat.t_series_ser.ser_wltyp_code IS
'Code of the habitat type, one of MO (marine open), MC (Marine coastal), T (Transitional w
COMMENT ON COLUMN dat.t_series_ser.ser_hab_code IS
'Code of the habitat, see tr_habitat_hab';
COMMENT ON COLUMN dat.t_series_ser.ser_are_code IS
'Code of the area, areas are geographical sector most often corresponding to stock units, s
COMMENT ON COLUMN dat.t_series_ser.ser_uni_code IS
'Annual value Unit, references table tr_unit_uni.';
COMMENT ON COLUMN dat.t_series_ser.ser_effort_uni_code IS
'Annual data collection effort unit code, references table tr_unit_uni.';
COMMENT ON COLUMN dat.t_series_ser.ser_description IS
'Concise description of the series. Should include species, stage targeted, location and g
COMMENT ON COLUMN dat.t_series_ser.ser_locationdescription IS
'This should provide a description of the site, e.g. if ist far inland, in the middle of a
(e.g. `Bresle river trap 3 km from the sea` or `IYFS/IBTS sampling in the Skagerrak-Kattege
COMMENT ON COLUMN dat.t_series_ser.ser_gea_code IS
'Code of the gear used, see tr_gear_gea, if a fishway or scientific monitoring device leave
COMMENT ON COLUMN dat.t_series_ser.ser_fiw_code IS
'Code the fishway, eg P0 for pool type fishway';
COMMENT ON COLUMN dat.t_series_ser.ser_mon_code IS
'Code the Monitoring device, eg S0 for Sonar';
COMMENT ON COLUMN dat.t_series_ser.ser_stocking IS
'Boolean, Is there restocking (for eel) or artificial reproduction in the river / basin, af
COMMENT ON COLUMN dat.t_series_ser.ser_stockingcomment IS
'Comment on stocking';
COMMENT ON COLUMN dat.t_series_ser.ser_protocol IS
'Describe sampling protocol';
COMMENT ON COLUMN dat.t_series_ser.ser_samplingstrategy IS
'Describe sampling strategy';
COMMENT ON COLUMN dat.t_series_ser.ser_datarightsholder IS
'Code of the data rights holder of the series, this field will be used in DATSU to acknowle
COMMENT ON COLUMN dat.t_series_ser.ser_datelastupdate IS
'Last modification in the series, from a trigger';
COMMENT ON COLUMN dat.t_series_ser.geom IS
'Series geometry column EPSG 4326, can be more detailed than the geometry for station';

GRANT ALL ON datbast.t_series_ser TO diaspara_admin;
GRANT SELECT ON datbast.t_series_ser TO diaspara_read;

```

7.1.2 Import t_series_ser from trutta database

```
ser <- readxl::read_xlsx(file.path(datawd,
  "WGBAST_TRS_densities2024filled.xlsx"), sheet = "dane")
ser <- janitor::clean_names(ser)
ser <- ser[,1:9]
# res <- dbGetQuery(con_diaspara, "SELECT * FROM datbast.t_series_ser")
# clipr::write_clip(colnames(res))
riv <- dbGetQuery(con_diaspara_admin,
  "SELECT are_code, riv_rivername FROM refbast.tr_rivernames_riv
    JOIN refbast.tr_area_are on are_code = riv_are_code")
riv$riv = stringi::stri_trans_general(riv$riv_rivername, "latin-ascii")
ser$riv <- stringi::stri_trans_general(ser$river, "latin-ascii")
ser <- ser |> inner_join(riv, by= join_by(riv))
# 584 on 10303 => 604 with stringi
# Need to add electrofishing types to gear
save(ser, file = "data/ser_wgbast.Rdata")
t_series_ser <-
  data.frame(
    "ser_id" = uuid::UUIDgenerate(n=nrow(ser)),
    "ser_code" = paste0(ser$country, "_",
      substr(stringi::stri_trans_general(ser$river, "latin-ascii"),1,4), "_P_TRS"),
    "ser_name" = ser$river,
    "ser_spe_code" = "TRS",
    "ser_lfs_code" = "P",
    "ser_are_code" = ser$are_code,
    "ser_wkg_code" = "WGBAST",
    "ser_ver_code" = "WGBAST-2025-1",
    "ser_cou_code" = ser$country,
    "ser_hab_code" = NA, # TODO integrate habitats from eiunis https://eunis.eea.europa.eu/ha
    "ser_gea_code" = '10.4', # electric fishing
    "ser_fiw_code" = NA,
    "ser_mon_code" = NA,
    "ser_uni_code" = "nr",
    "ser_effort_uni_code" = "nr/m2",
    "ser_description" = NA,
    "ser_locationdescription" =
      paste0("main_river="ser$main_river,", river = ",ser$river,",
        sub_div= ",ser$sub_div),
    "ser_wltyp_code" = "FW",
    "ser_stocking" = NA,
    "ser_stockingcomment" = NA,
```

```

        "ser_protocol" = NA,
        "ser_samplingstrategy" = NA,
        "ser_datarightsholder" = NA,
        "ser_datelastupdate" = '2025-06-09' #,
        # geom
    )

res <- dbWriteTable(con_diaspara_admin, "t_series_ser_temp",
                    t_series_ser, overwrite = TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM datbast.t_series_ser;")
dbExecute(con_diaspara_admin, "INSERT INTO datbast.t_series_ser
SELECT distinct on (ser_code)
  ser_id::uuid,
  ser_code,
  ser_name,
  ser_spe_code,
  ser_lfs_code,
  ser_are_code::TEXT,
  ser_wkg_code,
  ser_ver_code,
  ser_cou_code,
  ser_hab_code,
  ser_gea_code,
  ser_fiw_code,
  ser_mon_code,
  ser_uni_code,
  ser_effort_uni_code,
  ser_description,
  ser_locationdescription,
  ser_wltyp_code,
  ser_stocking,
  ser_stockingcomment,
  ser_protocol,
  ser_samplingstrategy,
  ser_datarightsholder,
  ser_datelastupdate::date
FROM t_series_ser_temp") # 25

```

The table `t_series_ser` contains an extract of the series. Only a few were joined when trying to match the names. A data integration using the habitat referential would allow to integrate at sub basin level though maybe a table of subbasins (with geometries) would need to be created.

ser_id	ser_code	ser_name	ser_spe_code	ser_lfs_code
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	EE_Valg_P_TRS	Valgejõgi	127187	P
b038c443-25aa-4ed5-9ffe-9e7e002800f1	FI_Akas_P_TRS	Äkäsjoki	127187	P
138859ce-3086-461d-8167-13eef6e0fe13	FI_Paka_P_TRS	Pakajoki	127187	P
7aa04bdc-fe00-4865-a959-6f9e8ef07fef	LT_Bart_P_TRS	Bartuva	127187	P
1f27a1af-75b9-4d61-8086-ad929d06fee2	LT_Duby_P_TRS	Dubysa	127187	P
7bf51353-0f9c-4c8b-b370-25f0f2fec4a6	LT_Jura_P_TRS	Jūra	127187	P
eaad449-6808-4e2f-998f-e550f3a9ead9	LT_Neri_P_TRS	Neris	127187	P
a78e8a8c-85c5-4c68-8a04-1b19312d1cd9	LT_Zeim_P_TRS	Žeimena	127187	P
5814f7ac-1fb2-40c6-8867-31cecc3b2d23	LV_Amat_P_TRS	Amata	127187	P
03487eee-e307-412d-9b0d-7b94a7e2be59	LV_Bras_P_TRS	Brasla	127187	P
6bd47a9d-b92b-45f2-aca1-9ae86a7a4e6f	LV_Durb_P_TRS	Durbe	127187	P
bfa6e3ff-c681-4a17-9b91-41d9f0452365	LV_Jaun_P_TRS	Jaunupe	127187	P
c261642a-848b-49ba-9eed-b3b06188c738	LV_Korg_P_TRS	Korģe	127187	P
9b1f00f6-7e57-40ec-aa84-7449c1bf8b5f	LV_Liel_P_TRS	Lielā Jugla	127187	P
15f6eab9-8b72-4cdb-8dbd-a03513ea7967	LV_Liga_P_TRS	Līgatne	127187	P
e2063839-865d-4001-87ef-10f8a9ad7951	LV_Maza_P_TRS	Mazā Jugla	127187	P
ed74d560-9ee0-4671-86f3-644935e0f907	LV_Svet_P_TRS	Svētupe	127187	P
a62624ef-e7fb-460a-9c12-e8fbf7eba01c	LV_Tebr_P_TRS	Tebra	127187	P
71a8bc1f-5a71-4c20-9e78-8636cbf7710a	PL_Draw_P_TRS	Drawa	127187	P
10b28540-c037-4575-a2e8-d0c051e17e78	PL_Drwe_P_TRS	Drwęca	127187	P

7.2 Creating table joining series and station

The table is created but empty. We need the stations in ICES first.

SQL code to create table `datbast.tj_seriesstation_ses`

```

DROP TABLE IF EXISTS datbast.tj_seriesstation_ses;
CREATE TABLE datbast.tj_seriesstation_ses (
  CONSTRAINT fk_ses_ser_id FOREIGN KEY (ses_ser_id)
    REFERENCES datbast.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT uk_ses_ser_id UNIQUE (ses_ser_id) ,
  CONSTRAINT fk_station_code FOREIGN KEY (ses_station_code)
    REFERENCES "ref"."StationDictionary" ("Station_Code")
) inherits (dat.tj_seriesstation_ses);

GRANT ALL ON datbast.tj_seriesstation_ses TO diaspara_admin;
GRANT SELECT ON datbast.tj_seriesstation_ses TO diaspara_read;

```

7.3 Creating inherited table for annual series

SQL code to create tables `datbast.t_serannual_san`

```
--DROP TABLE IF EXISTS datbast.t_serannual_san;

CREATE TABLE datbast.t_serannual_san (
  san_agegroup TEXT,
  CONSTRAINT ck_san_agegroup CHECK (san_agegroup IS NULL OR san_agegroup='0+' OR san_agegroup
  CONSTRAINT ck_san_wkg_code CHECK (san_wkg_code = 'WGBAST'),
  CONSTRAINT fk_san_ser_id FOREIGN KEY (san_ser_id)
    REFERENCES datbast.t_series_ser (ser_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT c_uk_san_id UNIQUE (san_id, san_wkg_code),
  CONSTRAINT fk_san_wkg_code FOREIGN KEY (san_wkg_code)
    REFERENCES ref.tr_icworkinggroup_wkg(wkg_code),
  CONSTRAINT fk_san_ver_code FOREIGN KEY (san_ver_code)
    REFERENCES refbast.tr_version_ver(ver_code),
  CONSTRAINT uk_san_year_svc UNIQUE(san_year, san_ser_id, san_agegroup)
) INHERITS (dat.t_serannual_san);

ALTER TABLE datbast.t_series_ser ALTER COLUMN ser_wkg_code SET DEFAULT 'WGBAST';

COMMENT ON TABLE datbast.t_serannual_san IS
  'Table of annual abundance data for series in datbast.t_series_ser, electrofishing series o
COMMENT ON COLUMN datbast.t_serannual_san.san_ser_id IS
  'UUID, identifier of the series, primary key, references the table ref.tr_seriesvocab_svc (
COMMENT ON COLUMN datbast.t_serannual_san.san_id IS
  'INTEGER, autoincremented, unique for one working group';
COMMENT ON COLUMN datbast.t_serannual_san.san_year IS
  'Year of monitoring, note that for some of the series this corresponds to the main migratio
COMMENT ON COLUMN datbast.t_serannual_san.san_agegroup IS
  'Age group of fish.';
COMMENT ON COLUMN datbast.t_serannual_san.san_comment IS
  'Comment on the annual value of the series';
COMMENT ON COLUMN datbast.t_serannual_san.san_effort IS
  'Eventually a measure of effort to collect the series, e.g. number of nr haul, nr fyke.day,
check the t_metadataannual table for the unit used';
COMMENT ON COLUMN datbast.t_serannual_san.san_datelastupdate IS
  'Date of last update on the annual data';
COMMENT ON COLUMN datbast.t_serannual_san.san_qal_id IS
  'Quality ID code of the series';
COMMENT ON COLUMN datbast.t_serannual_san.san_qal_comment IS
  'Comment related to data quality, e.g. why this year the series should not be used, or used
COMMENT ON COLUMN datbast.t_serannual_san.san_wkg_code IS
  'Code of the working group, one of
```

```

WGBAST, WGEEL, WGNAS, WKTRUTTA';
COMMENT ON COLUMN datbast.t_serannual_san.san_ver_code IS
  'Version code sourced from ref.tr_version_ver the data call e.g. NAS_2025dc_2020, wgeel_2020';

GRANT ALL ON datbast.t_serannual_san TO diaspara_admin;
GRANT SELECT ON datbast.t_serannual_san TO diaspara_read;

```

7.4 Import data for annual series

Data are imported from the electrofishing trutta db to `datbast.t_serannual_ser`.
Data are structured not by stage but by age group.

```

# start over from ser, previous chunk
load(file = "data/ser_wgbast.Rdata") #ser

ser0 <- dbGetQuery(con_diaspara_admin, "SELECT * FROM datbast.t_series_ser")
ser <- ser |> mutate(ser_code =
paste0(ser$country, "_", substr(
  stringi::stri_trans_general(ser$river, "latin-ascii"),1,4), "_P_TRS")) |>
  inner_join(ser0)

t_serannual_san_temp <-
  data.frame(
    "san_ser_id" = ser$ser_id,
    "san_id" = seq(along.with= ser$ser_id),
    "san_value"=ser$density_n_100m2,
    "san_year" = ser$year,
    "san_comment" = NA,
    "san_effort" = ser$number_of_sites,
    "san_datelastupdate" = Sys.Date(),
    "san_qal_id" = 1,
    "san_qal_comment" = NA,
    "san_wkg_code" = "WGBAST",
    "san_ver_code" = "WGBAST-2025-1",
    "san_agegroup" = ifelse(is.na(ser$age), "all", ser$age))

dbWriteTable(con_diaspara_admin, "t_serannual_san_temp",
             t_serannual_san_temp, overwrite = TRUE)
dbExecute(con_diaspara_admin, "DELETE FROM datbast.t_serannual_san;")
dbExecute(con_diaspara_admin, "INSERT INTO datbast.t_serannual_san
SELECT
  san_ser_id::uuid,
  san_id,

```

```

    san_value,
    san_year,
    san_comment,
    san_effort,
    san_datelastupdate,
    san_qal_id,
    san_qal_comment,
    san_wkg_code,
    san_ver_code,
    san_agegroup
FROM t_serannual_san_temp") # 604

```

i Note for age

Here age needs to be part of a constraint, values are 0+, >0+ but all is necessary to replace NULL.

```

dbGetQuery(con_diaspara, "SELECT * FROM databast.t_serannual_san limit 20;")|>
  knitr::kable()|>
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"))

```

Table 7.2: Table databast.t_serannual_san

san_ser_id	san_id	san_value	san_year	san_comment	san_effort	san_agegroup
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	1	0.0000000	1976	NA	1	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	2	0.8493366	1983	NA	1	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	3	7.9249550	1990	NA	1	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	4	0.0000000	1995	NA	1	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	5	2.0702579	1996	NA	2	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	6	0.4246683	1997	NA	1	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	7	0.2400354	1999	NA	7	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	8	0.5229656	2000	NA	5	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	9	0.1813328	2001	NA	4	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	10	0.0000000	2002	NA	1	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	11	0.0000000	2003	NA	3	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	12	0.7017058	2004	NA	2	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	13	2.4001195	2005	NA	3	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	14	4.0790627	2006	NA	3	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	15	4.2658571	2007	NA	3	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	16	4.5519545	2008	NA	3	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	17	5.7795221	2009	NA	3	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	18	2.9153959	2010	NA	3	2003
031889a1-9e58-4ffc-ac37-bd75ccd5b69a	19	1.0616671	2011	NA	3	2003

aaaad449-6808-4e2f-998f-e550f3a9ead9	20	3.3000000	2004	NA	NA	20
--------------------------------------	----	-----------	------	----	----	----

Chapter 8

Acknowledgements and references

Maria and Joana for their help. Rui and Carlos for their support.

ICES. 2024. “ICES Vocabularies and Codes.” ICES Data Flow Schematics.
<https://doi.org/10.17895/ICES.PUB.25288189.V1>.