



El interior de un Geopackage

El caso del proyecto DRAIN

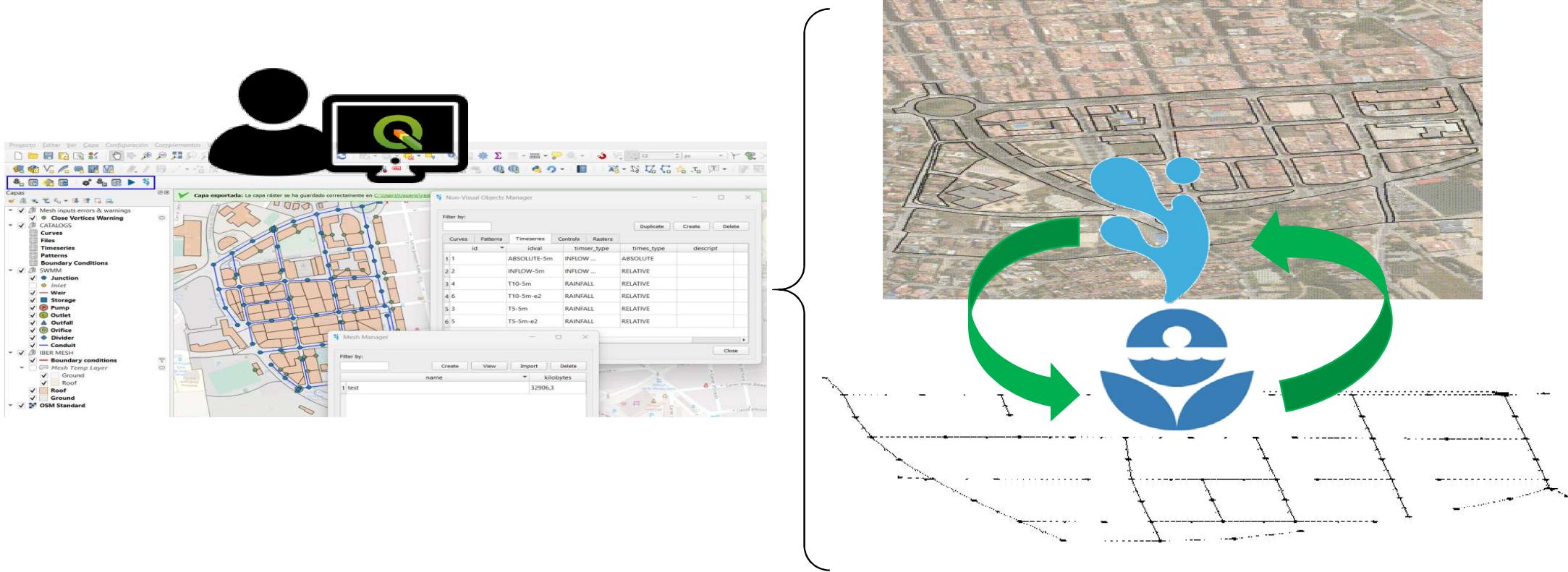
Contenidos



- Proyecto DRAIN
- Condiciones
- Contexto
- Situación 1: hacer un gpkg funcional desde SQL
- Situación 2: modificar el modelo de datos
- Situación 3: dar robustez a la base de datos

Proyecto Drain

B'GEO
OPEN GIS & WATER SOLUTIONS

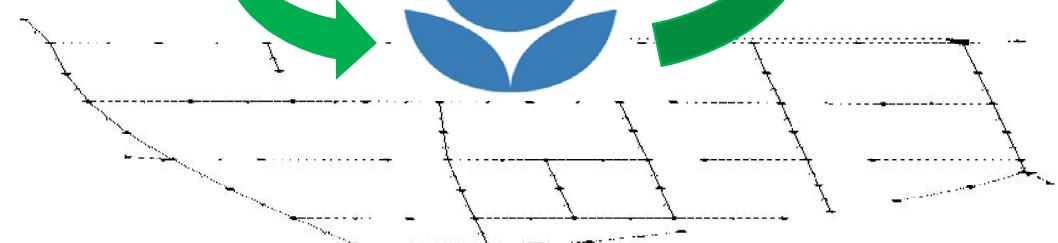


MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES



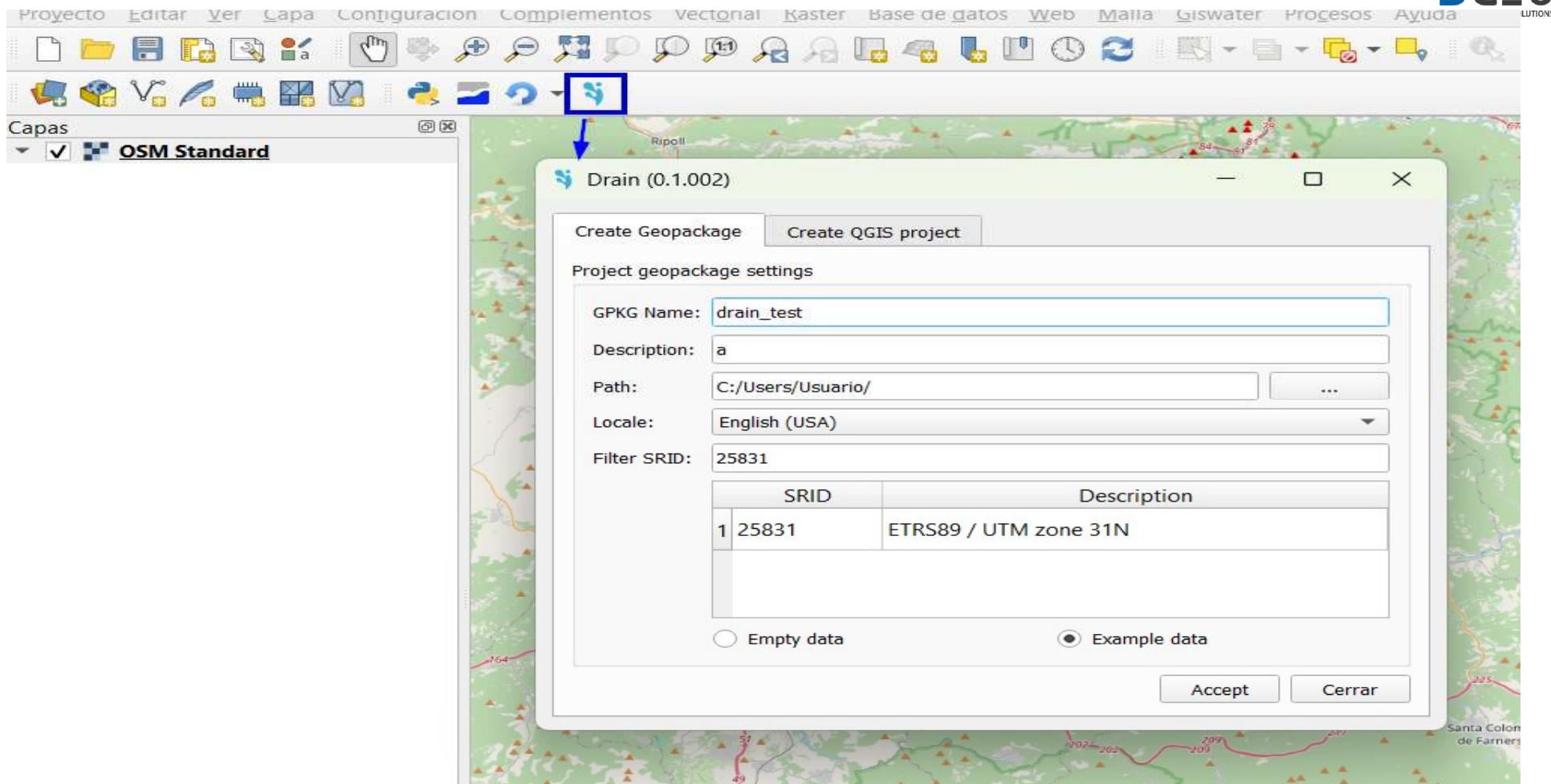
Condiciones

- Software libre
- Entorno: QGIS
- Soporte de datos: GeoPackage



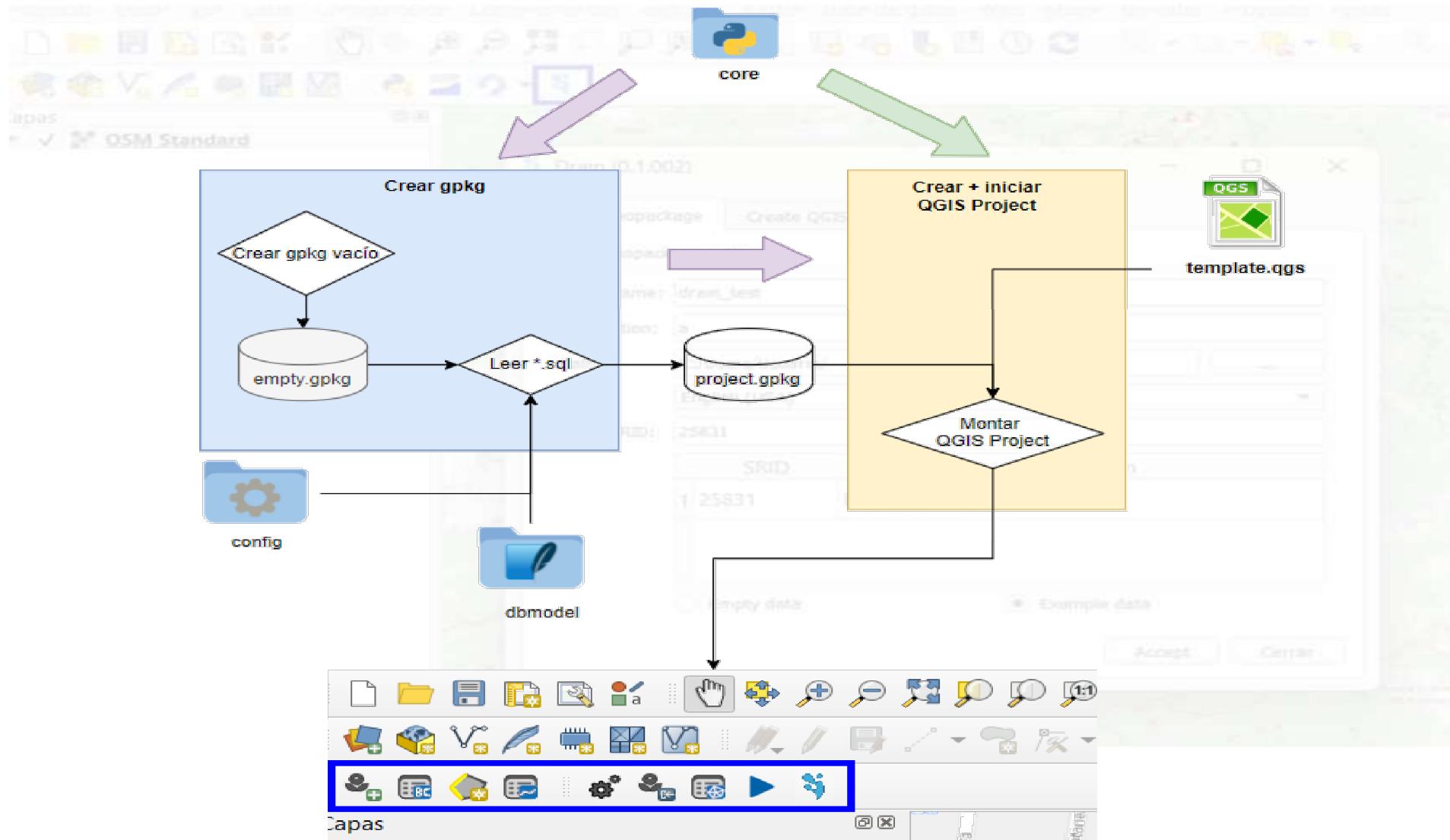
Contexto

B'GEO
OLUTIONS

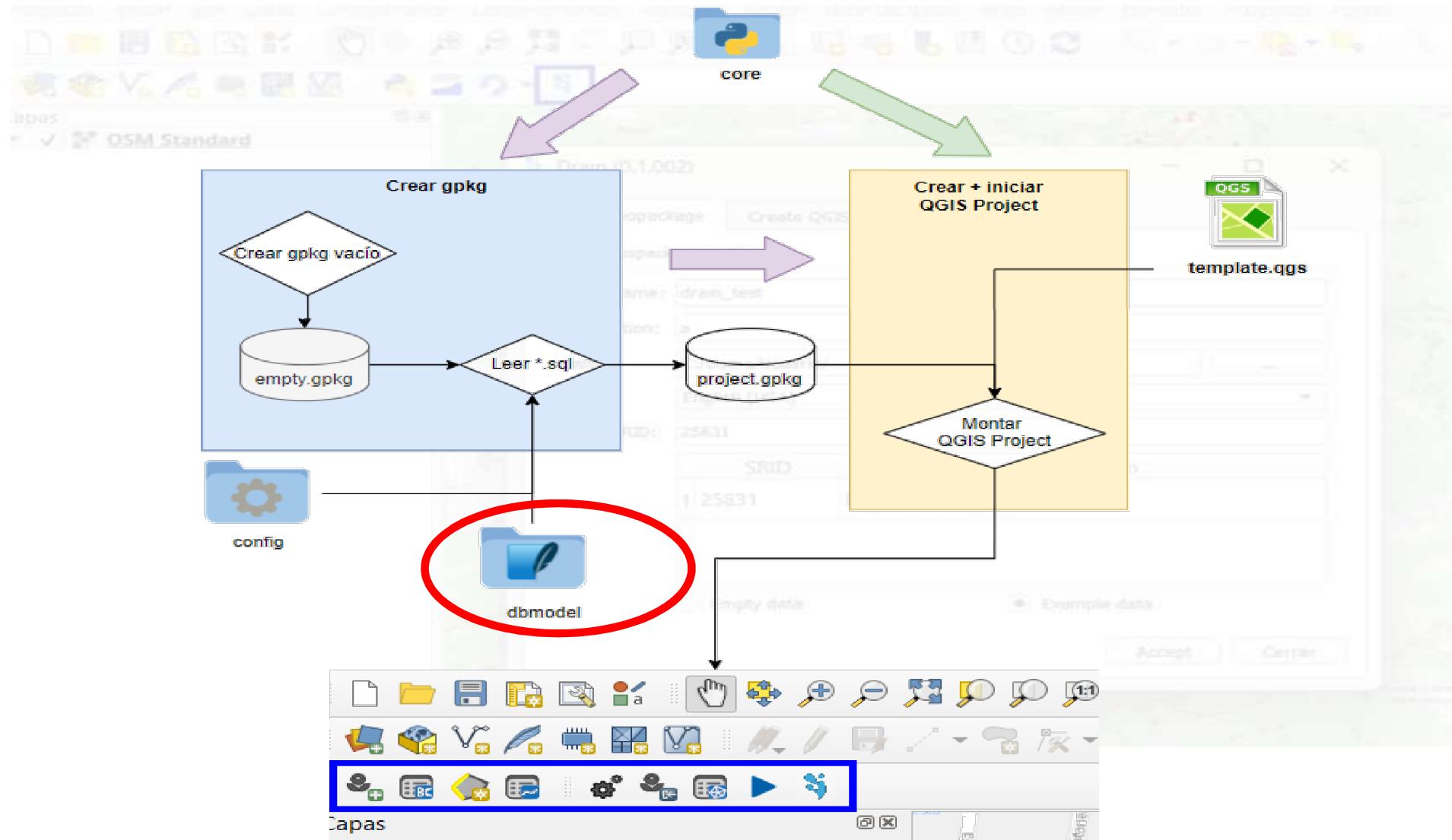


Contexto

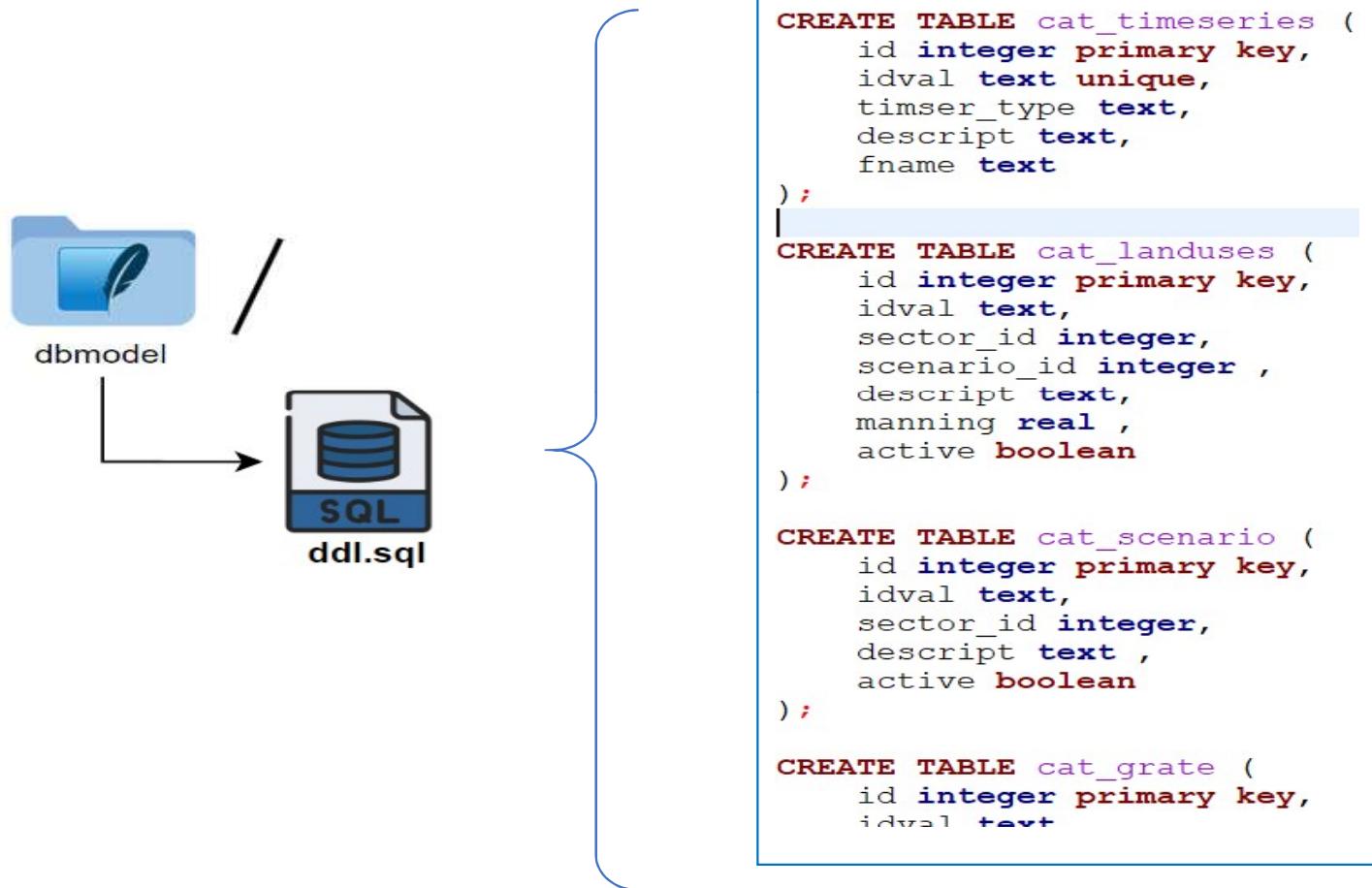
B'GEO
OLUTIONS



Contexto



Dbmodel: punto de partida

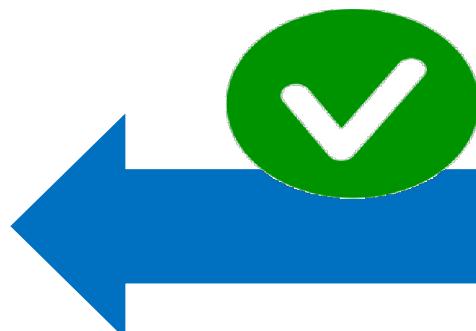
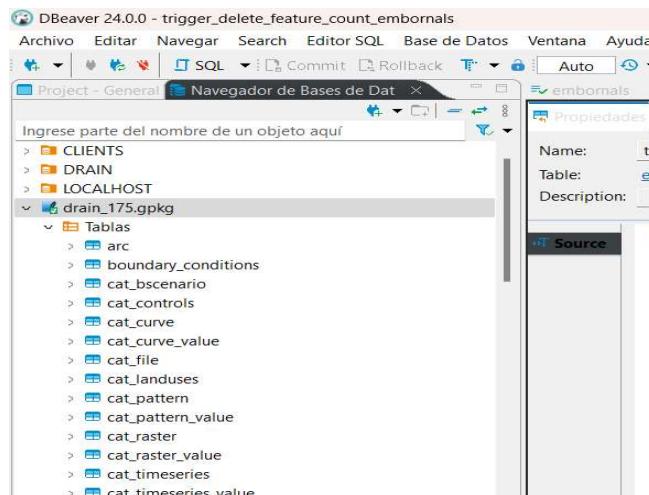
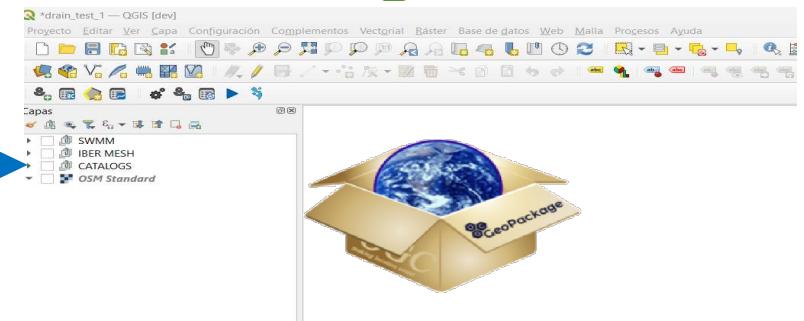
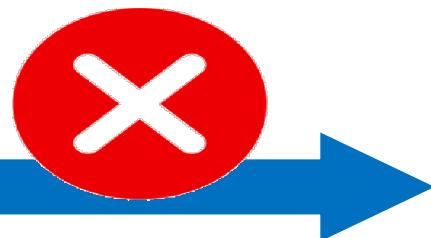


Situación 1: hacer un gpkg funcional desde SQL

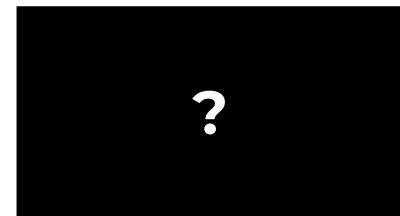
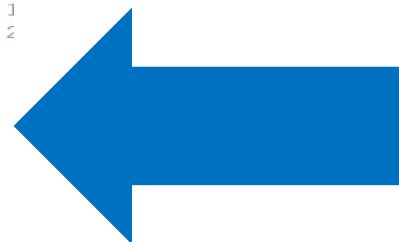
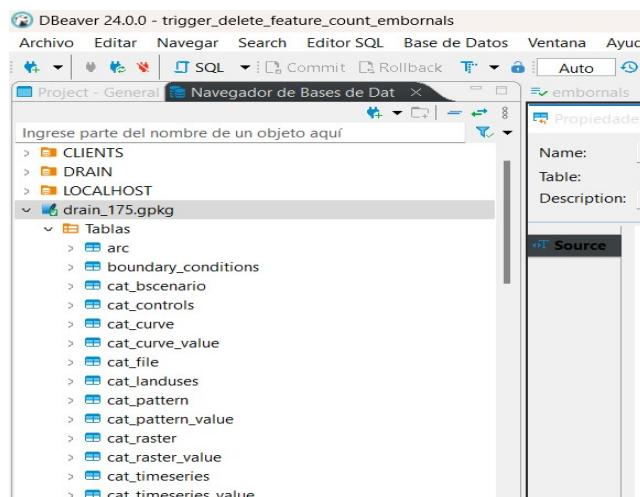
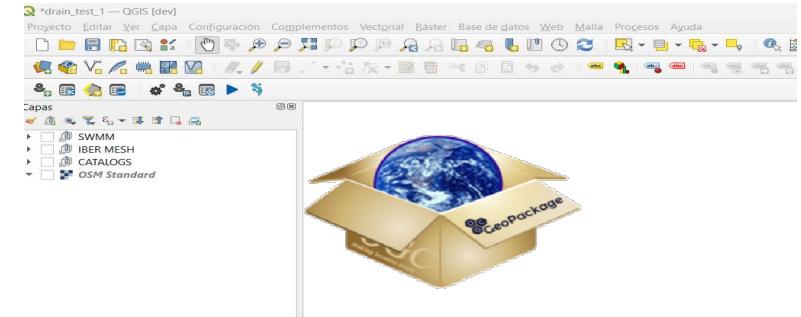
Situación 1: hacer un gpkg funcional desde SQL



```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```



Reto



Resolución



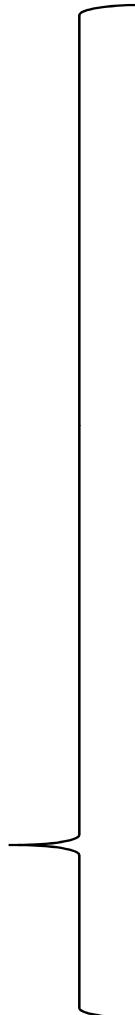
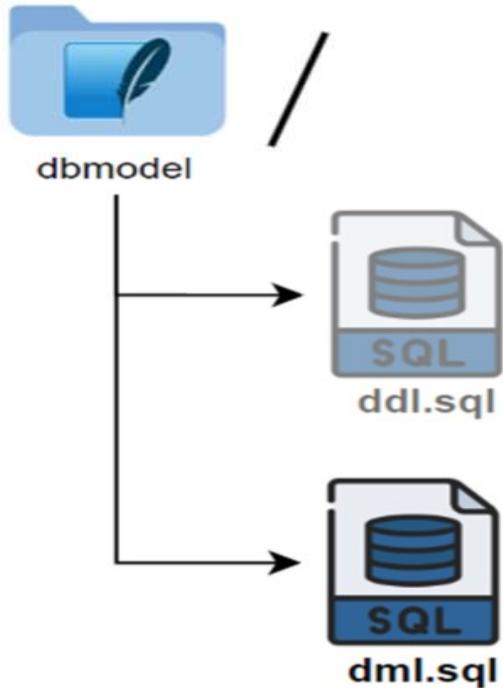
Resolución



- gpkg_contents
- gpkg_ogr_contents
- gpkg_geometry_columns

Resolución

- gpkg_contents
- gpkg_ogr_contents
- gpkg_geometry_columns



```
INSERT INTO gpkg_contents (tab VALUES('sys_selector', 'at
INSERT INTO gpkg_contents (tab VALUES('selector_sector',
INSERT INTO gpkg_contents (tab VALUES('selector_scenario',
INSERT INTO gpkg_contents (tab VALUES('config_param_user',
INSERT INTO gpkg_contents (tab VALUES('edit_typevalue',
INSERT INTO gpkg_contents (tab VALUES('cat_scenario', 'at
INSERT INTO gpkg_contents (tab VALUES('cat_curve', 'attr
INSERT INTO gpkg_contents (tab VALUES('cat_curve_value',
INSERT INTO gpkg_contents (tab VALUES('cat_timeseries',
INSERT INTO gpkg_contents (tab VALUES('cat_timeseries_val
INSERT INTO gpkg_contents (tab VALUES('cat_losses', 'attr
INSERT INTO gpkg_contents (tab VALUES('cat_losses_values
INSERT INTO gpkg_contents (tab VALUES('cat_landuses', 'at
INSERT INTO gpkg_contents (tab VALUES('cat_grate', 'attr
INSERT INTO gpkg_contents (tab VALUES('cat_pattern', 'attr
```

```
INSERT INTO gpkg_ogr_contents (tab VALUES('sys_selector', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('selector_sector', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('selector_scenario', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('config_param_user', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('edit_typevalue', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_scenario', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_curve', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_curve_value', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_timeseries', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_timeseries_value', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_losses', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_losses_values', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_landuses', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_grate', 0);
INSERT INTO gpkg_ogr_contents (tab VALUES('cat_pattern', 0);
```

```
INSERT INTO gpkg_geometry_columns (tab m) VALUES('sector', 'geom', 'MULTIPOLYGON', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('polygon', 'geom', 'MULTIPOLYGON', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('manzone', 'geom', 'MULTIPOLYGON', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('losszone', 'geom', 'MULTIPOLYGON', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('roof', 'geom', 'MULTIPOLYGON', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('elem_tin', 'geom', 'MULTIPOLYGON', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('elem_edge', 'geom', 'LINESTRING', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('elem_vertex', 'geom', 'POINT', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('raingage', 'geom', 'POINT', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('link', 'geom', 'LINESTRING', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('gully', 'geom', 'POINT', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('conduit', 'geom', 'LINESTRING', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('subcatchment', 'geom', 'MULTIPOLYGON', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('outlet', 'geom', 'LINESTRING', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('orifice', 'geom', 'LINESTRING', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('weir', 'geom', 'LINESTRING', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('pump', 'geom', 'LINESTRING', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('outfall', 'geom', 'POINT', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('divider', 'geom', 'POINT', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('storage', 'geom', 'POINT', <SRID_VALUE>, 0, 0);
INSERT INTO gpkg_geometry_columns (tab m) VALUES('junction', 'geom', 'POINT', <SRID_VALUE>, 0, 0);
```

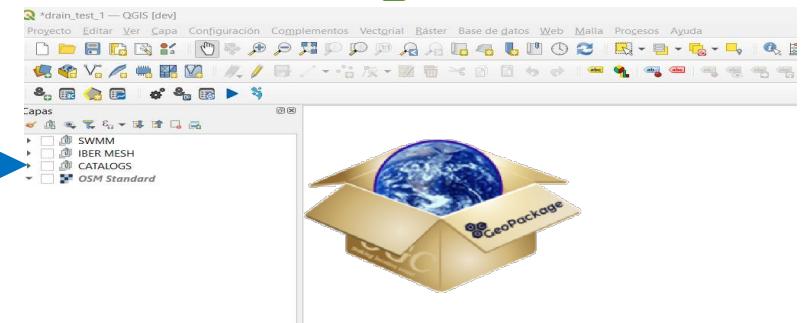
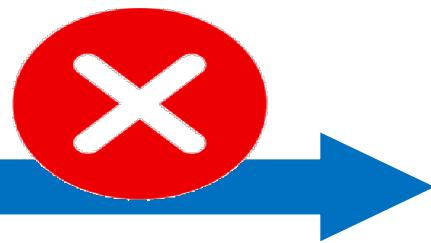
Resultado



Resultado



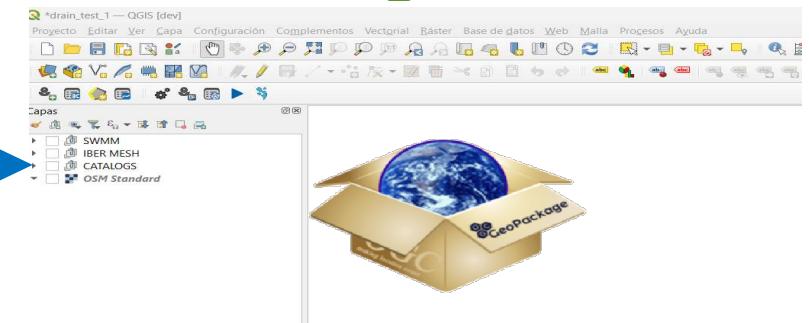
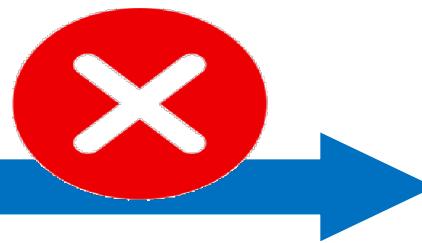
```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```



Resultado



```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```

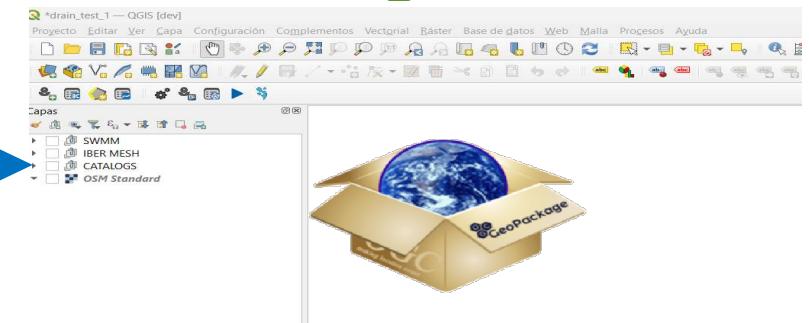
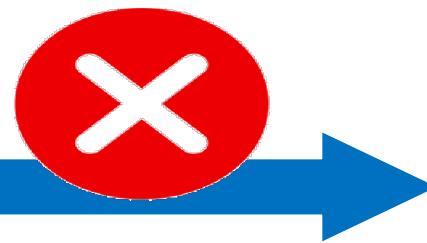


- > inp_divider
- ✓ inp_dwf
 - > Columnas
 - > Claves
 - > Columnas de clave externa
 - > Índices
 - > Referencias
 - ✓ Triggers
 - trigger_delete_feature_count_inp_dwf
 - trigger_insert_feature_count_inp_dwf
- > inp_files

Resultado



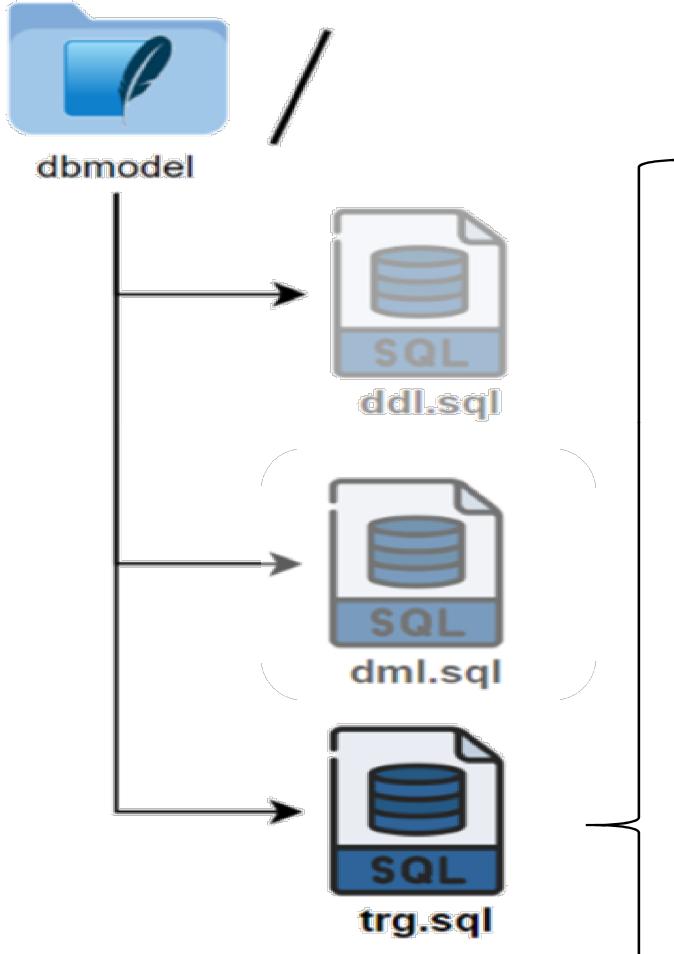
```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```



```
> inp_divider
< inp_dwf
  > Columns
  > Claves
  > Columnas de clave externa
  > Índices
  > Referencias
< Triggers
  > trigger_delete_feature_count_inp_dwf
  > trigger_insert_feature_count_inp_dwf
> inp_files
```



Resolución



```
CREATE TRIGGER "trigger_delete_feature_count_sys_selector" AFTER DELETE ON "sys_selector" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'sys_selector' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_selector_scenario" AFTER DELETE ON "selector_scenario" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'selector_scenario' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_selector_sector" AFTER DELETE ON "selector_sector" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'selector_sector' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_config_param_user" AFTER DELETE ON "config_param_user" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'config_param_user' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_scenario" AFTER DELETE ON "cat_scenario" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_scenario' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_curve" AFTER DELETE ON "cat_curve" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_curve' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_curve_value" AFTER DELETE ON "cat_curve_value" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_curve_value' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_timeseries" AFTER DELETE ON "cat_timeseries" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_timeseries' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_timeseries_value" AFTER DELETE ON "cat_timeseries_value" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_timeseries_value' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_landuses" AFTER DELETE ON "cat_landuses" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_landuses' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_grate" AFTER DELETE ON "cat_grate" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_grate' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_cat_pattern" AFTER DELETE ON "cat_pattern" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'cat_pattern' AND feature_count > 1;

CREATE TRIGGER "trigger_delete_feature_count_sector" AFTER DELETE ON "sector" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'sector' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_polygon" AFTER DELETE ON "polygon" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'polygon' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_manzone" AFTER DELETE ON "manzone" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'manzone' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_losszone" AFTER DELETE ON "losszone" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'losszone' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_roof" AFTER DELETE ON "roof" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'roof' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_elem_tin" AFTER DELETE ON "elem_tin" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'elem_tin' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_elem_edge" AFTER DELETE ON "elem_edge" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'elem_edge' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_elem_vertex" AFTER DELETE ON "elem_vertex" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'elem_vertex' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_raingage" AFTER DELETE ON "raingage" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'raingage' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_conduit" AFTER DELETE ON "conduit" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'conduit' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_subcatchment" AFTER DELETE ON "subcatchment" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'subcatchment' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_outlet" AFTER DELETE ON "outlet" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'outlet' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_orifice" AFTER DELETE ON "orifice" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'orifice' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_weir" AFTER DELETE ON "weir" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'weir' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_pump" AFTER DELETE ON "pump" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'pump' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_outfall" AFTER DELETE ON "outfall" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'outfall' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_divider" AFTER DELETE ON "divider" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'divider' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_storage" AFTER DELETE ON "storage" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'storage' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_junction" AFTER DELETE ON "junction" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'junction' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_link" AFTER DELETE ON "link" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'link' AND feature_count > 1;
CREATE TRIGGER "trigger_delete_feature_count_gully" AFTER DELETE ON "gully" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count - 1 WHERE Lower(table_name) = 'gully' AND feature_count > 1;

CREATE TRIGGER "trigger_insert_feature_count_sys_selector" AFTER INSERT ON "sys_selector" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'sys_selector';
CREATE TRIGGER "trigger_insert_feature_count_selector_sector" AFTER INSERT ON "selector_sector" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'selector_sector';
CREATE TRIGGER "trigger_insert_feature_count_selector_scenario" AFTER INSERT ON "selector_scenario" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'selector_scenario';
CREATE TRIGGER "trigger_insert_feature_count_config_param_user" AFTER INSERT ON "config_param_user" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'config_param_user';
CREATE TRIGGER "trigger_insert_feature_count_cat_scenario" AFTER INSERT ON "cat_scenario" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'cat_scenario';
CREATE TRIGGER "trigger_insert_feature_count_cat_curve" AFTER INSERT ON "cat_curve" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'cat_curve';
CREATE TRIGGER "trigger_insert_feature_count_cat_curve_value" AFTER INSERT ON "cat_curve_value" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'cat_curve_value';
CREATE TRIGGER "trigger_insert_feature_count_cat_timeseries" AFTER INSERT ON "cat_timeseries" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'cat_timeseries';
CREATE TRIGGER "trigger_insert_feature_count_cat_timeseries_value" AFTER INSERT ON "cat_timeseries_value" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'cat_timeseries_value';
CREATE TRIGGER "trigger_insert_feature_count_cat_landuses" AFTER INSERT ON "cat_landuses" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'cat_landuses';
CREATE TRIGGER "trigger_insert_feature_count_cat_grate" AFTER INSERT ON "cat_grate" BEGIN UPDATE gpkg_ogr_contents SET feature_count = feature_count + 1 WHERE Lower(table_name) = 'cat_grate';
```

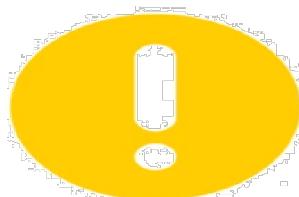
Resultado



Resultado



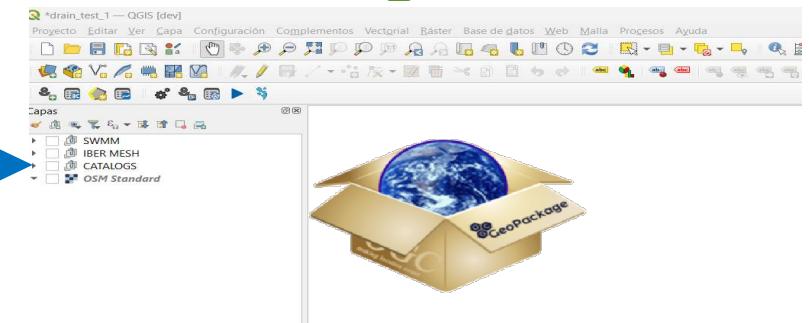
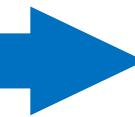
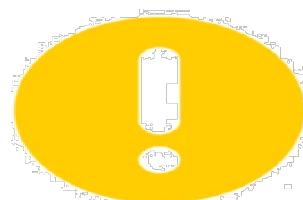
```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```



Resultado



```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```



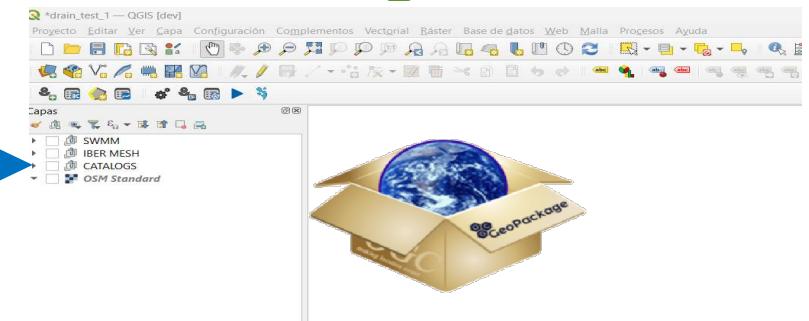
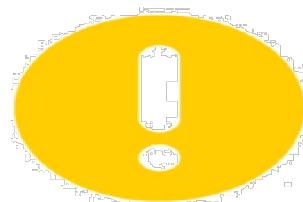
- > ground
- ✓ hyetograph
 - > Columns
 - > Claves
 - > Columnas de clave externa
 - > Índices
 - > Referencias
 - ✓ Triggers
 - ↳ rtree_hyetograph_geom_delete
 - ↳ rtree_hyetograph_geom_insert
 - ↳ rtree_hyetograph_geom_update1
 - ↳ rtree_hyetograph_geom_update2
 - ↳ rtree_hyetograph_geom_update3
 - ↳ rtree_hyetograph_geom_update4
 - ↳ trigger_delete_feature_count_hyetograph
 - ↳ trigger_insert_feature_count_hyetograph

Resultado



```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```

```
> ground
✓ hyetograph
  > Columns
  > Claves
  > Columnas de clave externa
  > Índices
  > Referencias
  ✓ Triggers
    > rtree_hyetograph_geom_delete
    > rtree_hyetograph_geom_insert
    > rtree_hyetograph_geom_update1
    > rtree_hyetograph_geom_update2
    > rtree_hyetograph_geom_update3
    > rtree_hyetograph_geom_update4
    > trigger_delete_feature_count_hyetograph
    > trigger_insert_feature_count_hyetograph
```



Resolución



Resolución

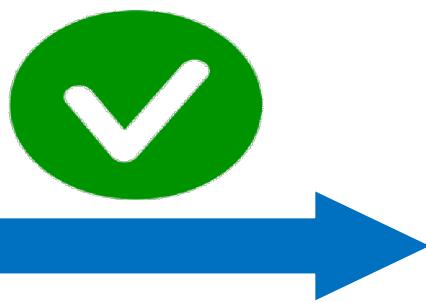
Resultado



Resultado



```
*<drain_175.gpkg> Script-149 ×
1 CREATE TABLE cat_timeseries (
2     id integer primary key,
3     idval text unique,
4     timser_type text,
5     descript text,
6     fname text
7 );
```



Situación 2: modificar el modelo de datos

Situación 2: modificar el modelo de datos



"Hay que borrar esta tabla"

Situación 2: modificar el modelo de datos

"Hay que borrar esta tabla"

dbmodel/dev/ddl/ddl.sql

```
Hunk 1 : Lines 10-15
10 10
11 11
12 12
13 --- NO-GEOM TABLES -----
14
15 CREATE TABLE sys_selector (
16     selector_id integer PRIMARY KEY,
17     parameter_text CHECK (typeof(parameter)='text' OR parameter=NULL),
18     value integer CHECK (typeof(value)=integer OR value=NULL)
19 );
20
21 CREATE TABLE sys_parameter (
22     parameter_id integer PRIMARY KEY,
23     parameter_text CHECK (typeof(parameter)='text' OR parameter=NULL),
24     value integer CHECK (typeof(value)=integer OR value=NULL)
```

dbmodel/dev/dml/dml.sql

```
Hunk 1 : Lines 9-14
9  9
10 10
11 11
12 --- INSERTS INTO SYS_GPKG_TABLES
13
14 --- INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
15     INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
16     INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
17     INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
```

dbmodel/dev/trg/trg.sql

```
Hunk 1 : Lines 10-15
10 10
11 11
12 12
13 --- TRIGGERS FOR SYS_GPKG_TABLESPACE -----
14
15 CREATE TRIGGER "trigger_delete_feature_count_sys_selector" AFTER DELETE ON "sys_selector" BEGIN
16     CREATE TRIGGER "trigger_delete_feature_count_sys_parameter" AFTER DELETE ON "sys_parameter" BEGIN
17         CREATE TRIGGER "trigger_delete_feature_count_sys_typevalue" AFTER DELETE ON "sys_typevalue" BEGIN
18             CREATE TRIGGER "trigger_delete_feature_count_cat_scenario" AFTER DELETE ON "cat_scenario" BEGIN
19             END
20         END
21     END
22 END
```



```
Hunk 2 : Lines 30-35
31 30
32 31
33 32
34 --- TRIGGERS FOR VERTEX -----
35
36 CREATE TRIGGER "trigger_delete_feature_count_vertex" AFTER DELETE ON "vertex" BEGIN UPDATE gpkg_contents SET last_change = now() WHERE table_name = 'vertex';
37     CREATE TRIGGER "trigger_delete_feature_count_rainage" AFTER DELETE ON "rainage" BEGIN UPDATE gpkg_contents SET last_change = now() WHERE table_name = 'rainage';
38
39 --- TRIGGERS FOR SYS_SELECTOR -----
40
41     CREATE TRIGGER "trigger_insert_feature_count_sys_selector" AFTER INSERT ON "sys_selector" BEGIN
42         CREATE TRIGGER "trigger_insert_feature_count_sys_parameter" AFTER INSERT ON "sys_parameter" BEGIN
43             CREATE TRIGGER "trigger_insert_feature_count_sys_typevalue" AFTER INSERT ON "sys_typevalue" BEGIN
44                 CREATE TRIGGER "trigger_insert_feature_count_cat_scenario" AFTER INSERT ON "cat_scenario" BEGIN
45                 END
46             END
47         END
48     END
49 END
```



Situación 2: modificar el modelo de datos

"Hay que borrar esta tabla"

dbmodel/dev/ddl/ddl.sql

```
Hunk 1 : Lines 10-15
10 10
11 11
12 12
13 --- NO-GEOM TABLES -----
14
15 CREATE TABLE sys_selector (
16     selector_id integer PRIMARY KEY,
17     parameter_text CHECK (typeof(parameter)='text' OR parameter=NULL),
18     value integer CHECK (typeof(value)=integer OR value=NULL)
19 );
20
21 CREATE TABLE sys_parameter (
22     parameter_id integer PRIMARY KEY,
23     parameter_text CHECK (typeof(parameter)='text' OR parameter=NULL),
24     value integer CHECK (typeof(value)=integer OR value=NULL)
```

dbmodel/dev/dml/dml.sql

```
Hunk 1 : Lines 9-14
9  9
10 10
11 11
12 --- INSERTS INTO SYS_GPKG_TABLES -----
13
14 --- INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
15     INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
16     INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
17     INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_value, max_value)
```

dbmodel/dev/trg/trg.sql

```
Hunk 1 : Lines 10-15
10 10
11 11
12 12
13 --- TRIGGERS FOR SYS_GPKG_TABLESPACE -----
14
15 CREATE TRIGGER "trigger_delete_feature_count_sys_selector" AFTER DELETE ON "sys_selector" BEGIN
16     CREATE TRIGGER "trigger_delete_feature_count_sys_parameter" AFTER DELETE ON "sys_parameter" BEGIN
17         CREATE TRIGGER "trigger_delete_feature_count_sys_typevalue" AFTER DELETE ON "sys_typevalue" BEGIN
18             CREATE TRIGGER "trigger_delete_feature_count_cat_scenario" AFTER DELETE ON "cat_scenario" BEGIN
```



```
Hunk 2 : Lines 30-35
31 30
32 31
33 32
34 --- TRIGGERS FOR VERTEX -----
35
36 CREATE TRIGGER "trigger_delete_feature_count_vertex" AFTER DELETE ON "vertex" BEGIN UPDATE gpkg_geometry_columns SET feature_count = feature_count - 1 WHERE column_name = 'vertex';
37     CREATE TRIGGER "trigger_delete_feature_count_rainage" AFTER DELETE ON "rainage" BEGIN UPDATE gpkg_geometry_columns SET feature_count = feature_count - 1 WHERE column_name = 'rainage';
38
39 --- TRIGGERS FOR INSERT -----
40
41     CREATE TRIGGER "trigger_insert_feature_count_sys_selector" AFTER INSERT ON "sys_selector" BEGIN
42         CREATE TRIGGER "trigger_insert_feature_count_sys_parameter" AFTER INSERT ON "sys_parameter" BEGIN
43             CREATE TRIGGER "trigger_insert_feature_count_sys_typevalue" AFTER INSERT ON "sys_typevalue" BEGIN
44                 CREATE TRIGGER "trigger_insert_feature_count_cat_scenario" AFTER INSERT ON "cat_scenario" BEGIN
```



"Hay que añadir otra tabla"

Situación 2: modificar el modelo de datos

"Hay que borrar esta tabla"

```

dbmodel/dev/ddl/ddl.sql
Hunk 1 : Lines 10-15
10 10
11 11
12 12
13
14 - CREATE TABLE sys_selector (
15 -   selector_id integer PRIMARY KEY,
16 -   parameter_text CHECK (typeof(parameter)='text' OR parameter=NULL),
17 -   selector_name text CHECK (typeof(selector_name)='text' OR selector_name=NULL),
18 -   value_integer CHECK (typeof(value)=integer OR value=NULL)
19 - );
20
21 - CREATE TABLE sys_parameter (
22 -   parameter_id integer PRIMARY KEY,
23 -   parameter_text CHECK (typeof(parameter)='text' OR parameter=NULL),
24 -   parameter_name text CHECK (typeof(parameter)='text' OR parameter=NULL),
25 -   parameter_value integer CHECK (typeof(value)=integer OR value=NULL),
26 -   parameter_type text CHECK (typeof(type)='text' OR type=NULL),
27 -   parameter_order integer CHECK (typeof(order)=integer OR order=NULL)
28
29
30
31
32
33
34
35

```

```

dbmodel/dev/dml/dml.sql
Hunk 1 : Lines 9-14
9 9
10 10
11 11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

```

dbmodel/dev/trg/trg.sql
Hunk 1 : Lines 10-15
10 10
11 11
12 12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```



```

dbmodel/dev/rtree/rtree.sql
Hunk 1 : Lines 10-16
10 10
11 11
12 12
13
14
15
16

```

"Hay que añadir otra tabla"

```

dbmodel/dev/ddl/ddl.sql
Hunk 1 : Lines 95-130
95 95
96 96
97 97
98 98
99 99
100 100
101 101
102 102
103 103
104 104
105 105
106 106
107 107
108 108
109 109
110 110
111 111
112 112
113 113
114 114
115 115
116 116
117 117

```

```

dbmodel/dev/dml/dml.sql
Hunk 1 : Lines 22-28
22 22
23 23
24 24
25 25
26 26
27 27
28 28

```

```

dbmodel/dev/trg/trg.sql
Hunk 1 : Lines 20-26
20 20
21 21
22 22
23 23
24 24
25 25
26 26
27 27
28 28

```



```

dbmodel/dev/trg/trg.sql
Hunk 1 : Lines 20-26
20 20
21 21
22 22
23 23
24 24
25 25
26 26

```

```

dbmodel/dev/trg/trg.sql
Hunk 1 : Lines 53-59
52 53
53 54
54 55
55 56
56 57
57 58
58 59

```

```

dbmodel/dev/trg/trg.sql
Hunk 2 : Lines 60-66
60 60
61 61
62 62
63 63
64 64
65 65
66 66

```

```

dbmodel/dev/trg/trg.sql
Hunk 3 : Lines 77-83
75 77
76 78
77 79
78 80
79 81
80 82
81 83

```

```

dbmodel/dev/trg/trg.sql
Hunk 4 : Lines 100-106
97 100
98 101
99 102
100 103
101 104
102 105

```

```

dbmodel/dev/trg/trg.sql
Hunk 5 : Lines 123-129
119 123
120 124
121 125

```

Situación 2: modificar el modelo de datos



“Te acuerdas de aquella tabla que...
etc. etc.? Pues ahora no hace falta”

Situación 2: modificar el modelo de datos

"Te acuerdas de aquella tabla que...
etc. etc.? Pues ahora no hace falta"



Situación 2: modificar el modelo de datos

"Te acuerdas de aquella tabla que...
etc. etc.? Pues ahora no hace falta"

The image shows a screenshot of a code editor with multiple windows open, illustrating a database schema modification process. The code is written in SQL and includes several triggers and table definitions.

dbmodel/dev/dml/dml.sql

```
25 25
26 26
27 27
28 28
29 29
30 30
31 30
```

Hunk 1 : Lines 25-30

```
-- INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_
-- INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_
-- INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_
-- INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_
-- INSERT INTO gpkg_contents (table_name, data_type, identifier, description, last_change, min_
```

dbmodel/dev/trg/trg.sql

```
24 24
25 25
26 26
```

Hunk 1 : Lines 24-25

```
-- CREATE TRIGGER "trigger_delete_feature_count_polygon" AFTER DELETE ON "polygon" -- BEGIN UPDATE gpkg_ogr_co
-- CREATE TRIGGER "trigger_delete_feature_count_point" AFTER DELETE ON "point" -- BEGIN UPDATE gpk
-- CREATE TRIGGER "trigger_delete_feature_count_manzone" AFTER DELETE ON "manzone" -- BEGIN UPDATE gpk
```

Hunk 2 : Lines 44-45

```
-- CREATE TRIGGER "trigger_insert_feature_count_polygon" AFTER INSERT ON "poly
-- CREATE TRIGGER "trigger_insert_feature_count_point" AFTER INSERT ON "point"
-- CREATE TRIGGER "trigger_insert_feature_count_manzone" AFTER INSERT ON "manz
```

Hunk 3 : Lines 54-55

```
-- CREATE TRIGGER "rtree_polygon_geom_delete" AFTER DELETE ON "polygon" WHEN old.fid = NEW.fid
-- CREATE TRIGGER "rtree_point_geom_delete" AFTER DELETE ON "point" WHEN old.fid = NEW.fid
-- CREATE TRIGGER "rtree_manzone_geom_delete" AFTER DELETE ON "manzone" WHEN old.fid = NEW.fid
```

Hunk 4 : Lines 63-64

```
-- CREATE TRIGGER "rtree_polygon_geom_insert" AFTER INSERT ON "polygon" WHEN (new.fid != OLD.fid)
-- CREATE TRIGGER "rtree_point_geom_insert" AFTER INSERT ON "point" WHEN (new.fid != OLD.fid)
-- CREATE TRIGGER "rtree_manzone_geom_insert" AFTER INSERT ON "manzone" WHEN (new.fid != OLD.fid)
```

Hunk 5 : Lines 72-73

```
-- CREATE TRIGGER "rtree_polygon_geom_update1" AFTER UPDATE OF "geom" ON "polygon" WHEN OLD.fid = NEW.fid
-- CREATE TRIGGER "rtree_point_geom_update1" AFTER UPDATE OF "geom" ON "point" WHEN OLD.fid = NEW.fid AND NEW.fid != OLD.fid
-- CREATE TRIGGER "rtree_manzone_geom_update1" AFTER UPDATE OF "geom" ON "manzone" WHEN OLD.fid = NEW.fid
```

Hunk 6 : Lines 81-82

```
-- CREATE TRIGGER "rtree_polygon_geom_update2" AFTER UPDATE OF "geom" ON "polygon" WHEN OLD.fid = NEW.fid
-- CREATE TRIGGER "rtree_point_geom_update2" AFTER UPDATE OF "geom" ON "point" WHEN OLD.fid = NEW.fid AND NEW.fid != OLD.fid
-- CREATE TRIGGER "rtree_manzone_geom_update2" AFTER UPDATE OF "geom" ON "manzone" WHEN OLD.fid = NEW.fid
```

Hunk 7 : Lines 90-91

```
-- CREATE TRIGGER "rtree_polygon_geom_update3" AFTER UPDATE ON "polygon" WHEN OLD.fid != NEW.fid AND (NEW.geom_id != OLD.geom_id)
-- CREATE TRIGGER "rtree_point_geom_update3" AFTER UPDATE ON "point" WHEN OLD.fid != NEW.fid AND (NEW.geom_id != OLD.geom_id)
-- CREATE TRIGGER "rtree_manzone_geom_update3" AFTER UPDATE ON "manzone" WHEN OLD.fid != NEW.fid AND (NEW.manzone_id != OLD.manzone_id)
```

dbmodel/dev/ddl/ddl.sql

```
100 100
101 101
102 102
```

Hunk 1 : Lines 100-105

```
-- descript text CHECK (typeof(descript)='text' OR descript=NULL)
-- );
-- CREATE TABLE point (
-- fid integer PRIMARY KEY,
-- geom geometry,
-- sector_id integer CHECK (typeof(sector_id)='integer' OR sector_id=NULL),
-- scenario_id integer CHECK (typeof(scenario_id)='integer' OR scenario_id=NULL),
-- elevation real CHECK (typeof(elevation)='real' OR elevation = NULL)
-- );
-- 
-- CREATE TABLE manzone (
-- fid integer PRIMARY KEY,
-- manzone_id integer CHECK (typeof(manzone_id)='integer' OR manzone_id=NULL),
```

Situación 2: modificar el modelo de datos



Situación 2: modificar el modelo de datos



Resolución

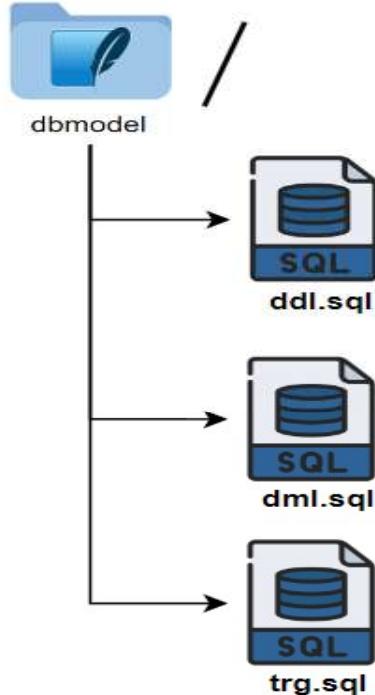


	table_name	isgeom	index_col
1	ground	MULTIPOLYGON	code
2	roof	MULTIPOLYGON	code
3	mesh_anchor_points	POINT	[NULL]
4	mesh_anchor_lines	LINESTRING	[NULL]
5	boundary_conditions	MULTILINESTRING	code
6	inlet	POINT	code
7	inp_conduit	LINESTRING	code
8	inp_outlet	LINESTRING	code
9	inp_orifice	LINESTRING	code
10	inp_weir	LINESTRING	code
11	inp_pump	LINESTRING	code
12	inp_outfall	POINT	code
13	inp_divider	POINT	code
14	inp_storage	POINT	code
15	inp_junction	POINT	code
16	node	POINT	fid
17	arc	LINESTRING	fid

	table_name	index_col
1	config_params_user	[NULL]
2	cat_bsenario	idval
3	cat_file	[NULL]
4	cat_landuses	idval
5	cat_transects	idval
6	cat_transects_value	id
7	cat_curve	idval
8	cat_curve_value	[NULL]
9	cat_timeSeries	idval
10	cat_timeSeries_value	[NULL]
11	cat_pattern	idval
12	cat_pattern_value	[NULL]
13	cat_raster	idval
14	cat_raster_value	[NULL]
15	cat_controls	[NULL]
16	inp_files	idval
17	inp_dwf	code
18	inp_inflow	code

```
def _execute_trg_creation(self):  
  
    # Geom tables  
    sql = "SELECT table_name, index_col FROM tables_geom;"  
    rows = tools_db.get_rows(sql)  
    list_tbl_geom = [(row[0], row[1]) for row in rows]  
  
    for tablename, index_col in list_tbl_geom:  
        if index_col:  
            sql = f"""CREATE INDEX idx_{index_col}_{tablename} ON {tablename} ({index_col});"""  
            tools_db.execute_sql(sql, commit=False)  
  
            aux_str = "AFTER"  
            if 'v_' in tablename or 'vi_' in tablename:  
                aux_str = "INSTEAD OF"  
            sql = f"""CREATE VIRTUAL TABLE rtree_{tablename}_geom USING rtree(id, minx, maxx, miny, maxy);"""  
            tools_db.execute_sql(sql, commit=False)  
            sql = f"""CREATE TRIGGER trigger_delete_feature_count_{tablename} {aux_str} DELETE ON {tablename} BEGIN UPDATE  
            tools_db.execute_sql(sql, commit=False)  
            sql = f"""CREATE TRIGGER trigger_insert_feature_count_{tablename} {aux_str} INSERT ON {tablename} BEGIN UPDATE  
            tools_db.execute_sql(sql, commit=False)  
            sql = f"""CREATE TRIGGER rtree_{tablename}_geom_delete {aux_str} DELETE ON {tablename} WHEN (old.geom NOT NULL)
```

Resolución



```
intvl text check (typeof(intvl) = 'text' or intvl = null),
scf real check (typeof(scf) = 'real' or scf = null),
fname text check (typeof(fname) = 'text' or fname = null),
sta text check (typeof(sta) = 'text' or sta = null),
units text check (typeof(units) = 'text' or units = null),
annotation text check (typeof(annotation) = 'text' or annotation = null),
source_fid integer check (typeof(source_fid) = 'integer' or source_fid = null),
geom geometry
);

CREATE TABLE link (
    fid integer primary key,
    code text check (typeof(code) = 'text' or code = null),
    sector_id integer check (typeof(sector_id) = 'integer' or sector_id = null),
    scenario_d integer check (typeof(scenario_d) = 'integer' or scenario_d = null),
    descript text check (typeof(descript) = 'text' or descript = null),
    feature_type text check (typeof(feature_type) = 'text' or feature_type = null),
    feature_id text check (typeof(feature_id) = 'text' or feature_id = null),
    exit_type text check (typeof(exit_type) = 'text' or exit_type = null),
    exit_id text check (typeof(exit_id) = 'text' or exit_id = null),
    annotation text check (typeof(annotation) = 'text' or annotation = null)
);
```

```
-- -- CREATE SYS GPKG REQUIREMENTS DYNAMICALLY
-- --
create table tables_nogeom (table_name text primary key, index_col text);
create table tables_geom (table_name text primary key, isgeom text NOT NULL, index_col text);

create trigger "trigger_tables_nogeom" after insert on "tables_nogeom"
BEGIN
    select new.table_name from tables_nogeom;
    INSERT INTO gpkg_ogr_contents (table_name, feature_count)
    VALUES(new.table_name, 0);

    insert into gpkg_contents (table_name, data_type, identifier, description, last_change, min_x, min_y, max_x, max_y, srs_id)
    values (new.table_name, 'attributes', new.table_name, '', 0, 0, 0, 0, 0, 0);

END;

create trigger trigger_tables_geom after insert on "tables_geom"
BEGIN
    select new.table_name, new.isgeom from tables_geom;
    INSERT INTO gpkg_ogr_contents (table_name, feature_count)
    VALUES(new.table_name, 0);

    insert into gpkg_contents (table_name, data_type, identifier, description, last_change, min_x, min_y, max_x, max_y, srs_id)
    values (new.table_name, 'features', new.table_name, '', 0, 0, 0, 0, 0, 0);
    insert into gpkg_geometry_columns (table_name, column_name, geometry_type_name, srs_id, z, m) VALUES(new.table_name, 'geom', new.isgeom, <SRID_VALUE>, 0, 0);
END;
```

Resolución



Resolución



dbmodel



```
-- TRIGGERS TO CREATE AN AUTOINDEX FOR EACH ELEMENT
create trigger "trg_ins_code_inp_outlet" AFTER INSERT on "inp_outlet" FOR EACH ROW BEGIN update inp_outlet set code = 'T'||fid; END;
create trigger "trg_ins_code_inp_weir" AFTER INSERT on "inp_weir" FOR EACH ROW BEGIN update inp_weir set code = 'W'||fid; END;
create trigger "trg_ins_code_inp_orifice" AFTER INSERT on "inp_orifice" FOR EACH ROW BEGIN update inp_orifice set code = 'R'||fid; END;
create trigger "trg_ins_code_inp_pump" AFTER INSERT on "inp_pump" FOR EACH ROW BEGIN update inp_pump set code = 'P'||fid; END;
create trigger "trg_ins_code_inp_conduit" AFTER INSERT on "inp_conduit" FOR EACH ROW BEGIN update inp_conduit set code = 'C'||fid; END;
create trigger "trg_ins_code_inp_storage" AFTER INSERT on "inp_storage" FOR EACH ROW BEGIN update inp_storage set code = 'S'||fid; END;
create trigger "trg_ins_code_inp_junction" AFTER INSERT on "inp_junction" FOR EACH ROW BEGIN update inp_junction set code = 'J'||fid; END;
create trigger "trg_ins_code_inp_outfall" AFTER INSERT on "inp_outfall" FOR EACH ROW BEGIN update inp_outfall set code = 'O'||fid; END;
create trigger "trg_ins_code_inp_divider" AFTER INSERT on "inp_divider" FOR EACH ROW BEGIN update inp_divider set code = 'D'||fid; END;
create trigger "trg_ins_code_roof" AFTER INSERT on "roof" FOR EACH ROW BEGIN update inp_divider set code = 'RF'||fid; END;

CREATE TRIGGER trg_upd_code_inp_outlet AFTER UPDATE of code on inp_outlet FOR EACH ROW BEGIN update arc set code = NEW.code where
CREATE TRIGGER trg_upd_code_inp_weir AFTER UPDATE of code on inp_weir FOR EACH ROW BEGIN update arc set code = NEW.code where
CREATE TRIGGER trg_upd_code_inp_orifice AFTER UPDATE of code on inp_orifice FOR EACH ROW BEGIN update arc set code = NEW.code where
CREATE TRIGGER trg_upd_code_inp_pump AFTER UPDATE of code on inp_pump FOR EACH ROW BEGIN update arc set code = NEW.code where
CREATE TRIGGER trg_upd_code_inp_conduit AFTER UPDATE of code on inp_conduit FOR EACH ROW BEGIN update arc set code = NEW.code where

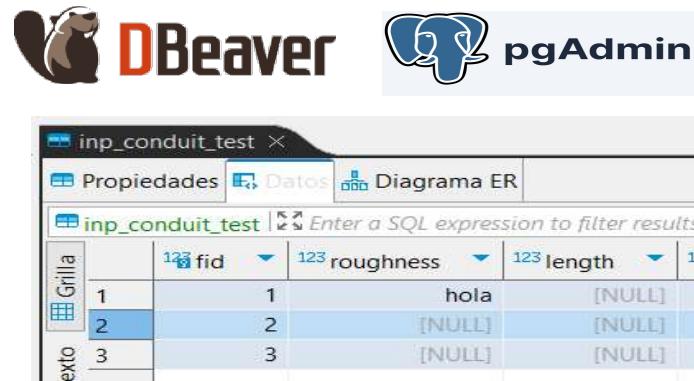
CREATE TRIGGER trg_del_inp_outlet AFTER DELETE on inp_outlet FOR EACH ROW BEGIN delete from arc where code = OLD.code and table
CREATE TRIGGER trg_del_inp_orifice AFTER DELETE on inp_orifice FOR EACH ROW BEGIN delete from arc where code = OLD.code and tab
CREATE TRIGGER trg_del_inp_conduit AFTER DELETE on inp_conduit FOR EACH ROW BEGIN delete from arc where code = OLD.code and tab
CREATE TRIGGER trg_del_inp_weir AFTER DELETE on inp_weir FOR EACH ROW BEGIN delete from arc where code = OLD.code and table_nam
CREATE TRIGGER trg_del_inp_pump AFTER DELETE on inp_pump FOR EACH ROW BEGIN delete from arc where code = OLD.code and table_nam

-- TOPOCONTROL TRIGGERS: SET node_1 AND node_2 TO ARCS BY PROXIMITY (conduits, pumps, outlets, orifices, weir)
----- inp_conduit
CREATE TRIGGER trg_ins_nodes_inp_conduit AFTER INSERT ON inp_conduit FOR EACH ROW
BEGIN
    UPDATE inp_conduit SET
        node_2 = (SELECT node.code FROM node WHERE ST_Intersects(ST_Buffer(node.geom, 0.1), ST_EndPoint(NEW.geom)) LIMIT 1),
        node_1 = (SELECT node.code FROM node WHERE ST_Intersects(ST_Buffer(node.geom, 0.1), ST_StartPoint(NEW.geom)) LIMIT 1),
        WHERE fid = NEW.fid;-- AND (node_1 IS NULL OR node_2 IS NULL);
END;

CREATE TRIGGER trg_upd_nodes_inp_conduit AFTER UPDATE OF geom ON inp_conduit FOR EACH ROW
BEGIN
    UPDATE inp_conduit SET
        node_2 = (SELECT node.code FROM node WHERE ST_Intersects(ST_Buffer(node.geom, 0.1), ST_EndPoint(NEW.geom)) LIMIT 1),
        node_1 = (SELECT node.code FROM node WHERE ST_Intersects(ST_Buffer(node.geom, 0.1), ST_StartPoint(NEW.geom)) LIMIT 1),
        WHERE fid = NEW.fid;-- AND (node_1 IS NULL OR node_2 IS NULL);
END;
```

Situación 3: dar robustez a la base de datos

Situación 3: dar robustez a la base de datos



	fid	roughness	length	z1
1	1	hola	[NULL]	
2	2	[NULL]	[NULL]	
3	3	[NULL]	[NULL]	



	fid	roughness	length	z1
1	1	NULL	NULL	
2	2	NULL	NULL	
3	3	NULL	NULL	

Situación 3: dar robustez a la base de datos

A screenshot of the QGIS application interface. In the center is a table view showing four columns: 'fid', 'roughness', 'length', and 'z1'. There are three rows of data. The first row has 'fid' value 1, 'roughness' value 'NULL', 'length' value 'NULL', and 'z1' value 'NULL'. The second row has 'fid' value 2, 'roughness' value 'NULL', 'length' value 'NULL', and 'z1' value 'NULL'. The third row has 'fid' value 3, 'roughness' value 'NULL', 'length' value 'NULL', and 'z1' value 'NULL'.

fid	roughness	length	z1
1	NULL	NULL	NULL
2	NULL	NULL	NULL
3	NULL	NULL	NULL

Reto

Restringir el datatype en la base de datos

Resolución

Restricciones CHECK en todos los campos

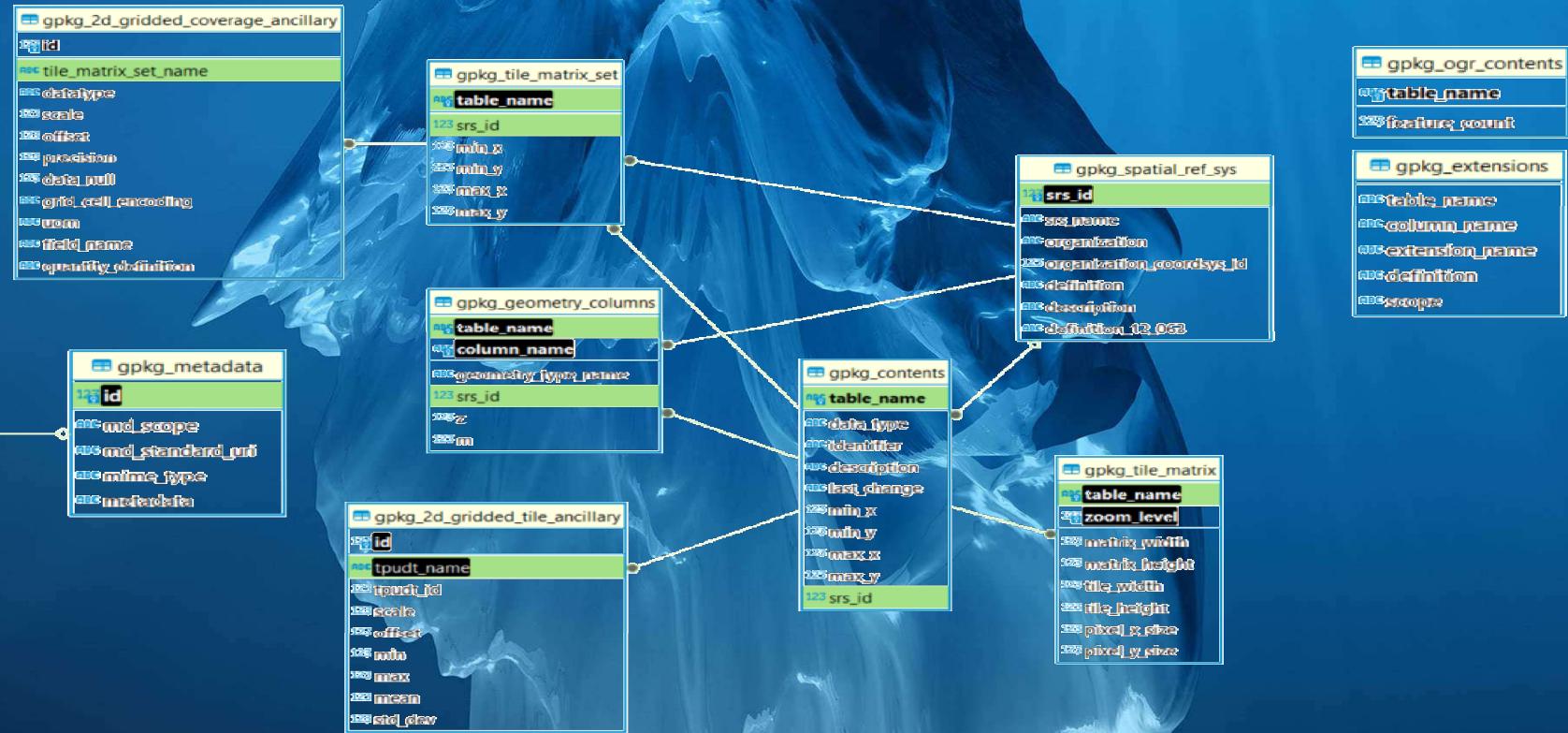
```
roughness real check (typeof(roughness) = 'real' or roughness = null),
length real check (typeof(length) = 'real' or length = null),
z1 real check (typeof(z1) = 'real' or z1 = null),
z2 real check (typeof(z2) = 'real' or z2 = null),
q0 real check (typeof(q0) = 'real' or q0 = null),
qmax real check (typeof(qmax) = 'real' or qmax = null),
```

...también para restringir a un dominio de valores

```
create table inp_files (
    id integer primary key,
    idval text unique,
    actio_type text check (typeof(actio_type) in ('text', null) and actio_type in ('SAVE', 'USE')),
    file_type text check (typeof(file_type) in ('text', null) and file_type in ('HOTSTART', 'INFLows', 'OUTFLOws', 'RAINFALL', 'RDII', 'RUNOFF')),
    fname text check (typeof(fname) = 'text' or fname = null),
    descript text CHECK (typeof(descript)='text' OR descript=NULL)
);

create table inp_dwf (
    fid integer primary key,
    code text unique check (typeof(code) = 'text' or code = null),
```

Para finalizar



B'GEO

OPEN GIS & WATER SOLUTIONS



www.bgeo.es/en

T. +34 938 600 293

info@bgeo.es

