

PostGIS & QGIS

tips & tricks

FOSS4G 2022 - Firenze - 26/08/2022



Julien Cabieces - Oslandia

 @troopa81

 @CabiecesJ

I'll talk about...

- Relations
- PostgreSQL Notify
- Processing to database
- Managing DB from browser
- Save project in database
- SQL Logging

Not an exhaustive feature list

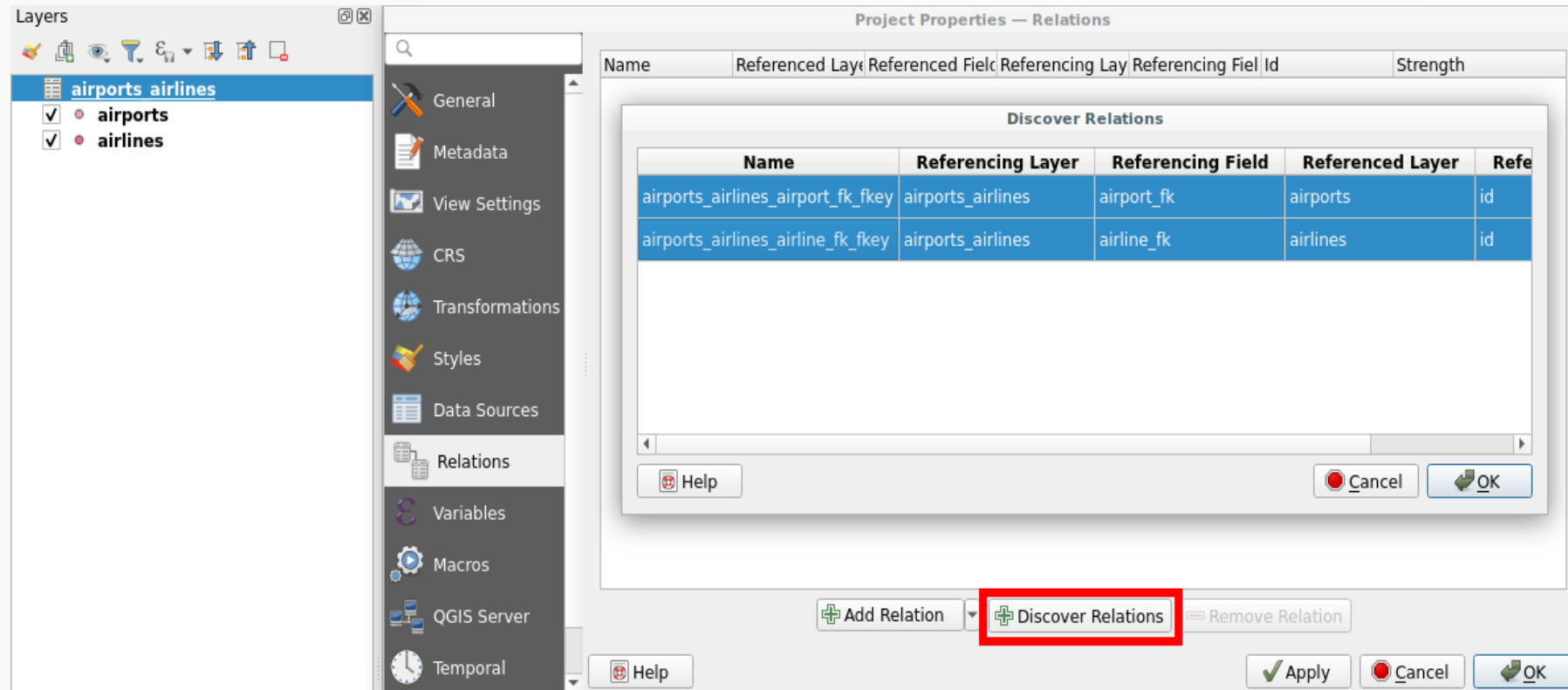
Relations

Use case



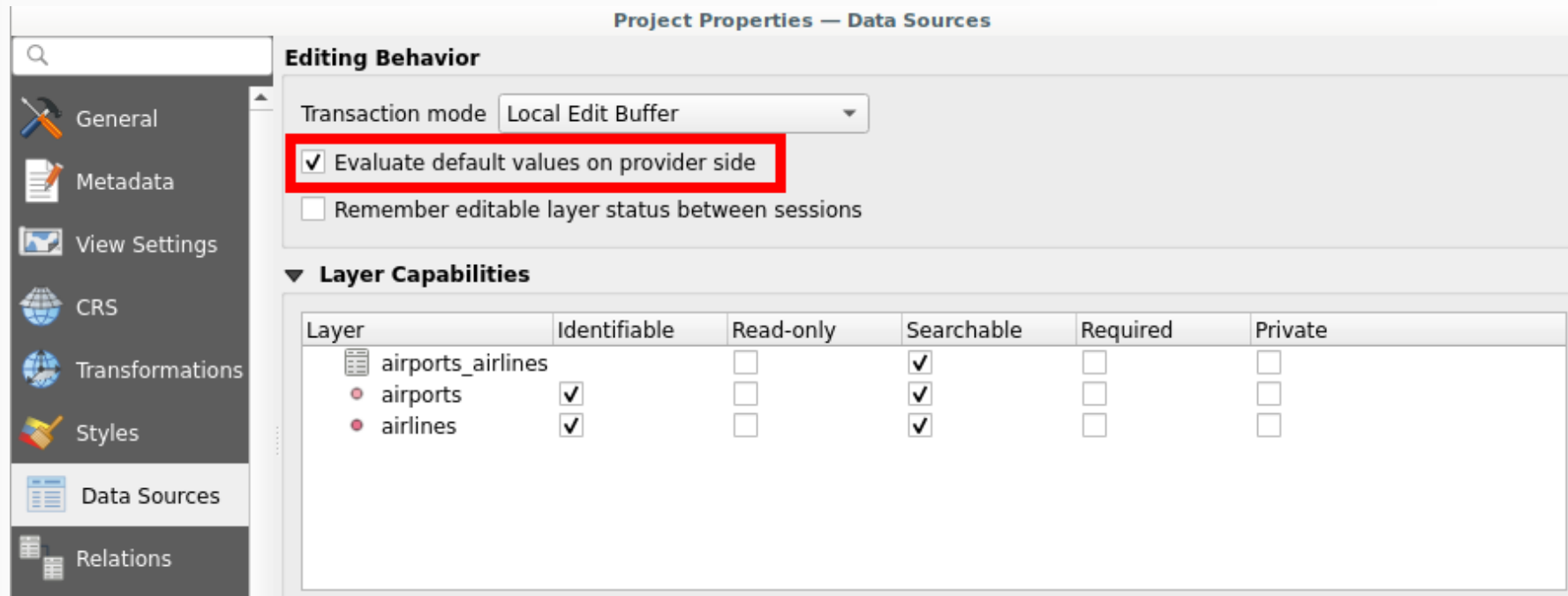
- **NM Relation**
- Airports have airlines...
- and airlines have airports
- see [QGIS doc](#) for more informations

Discover relations



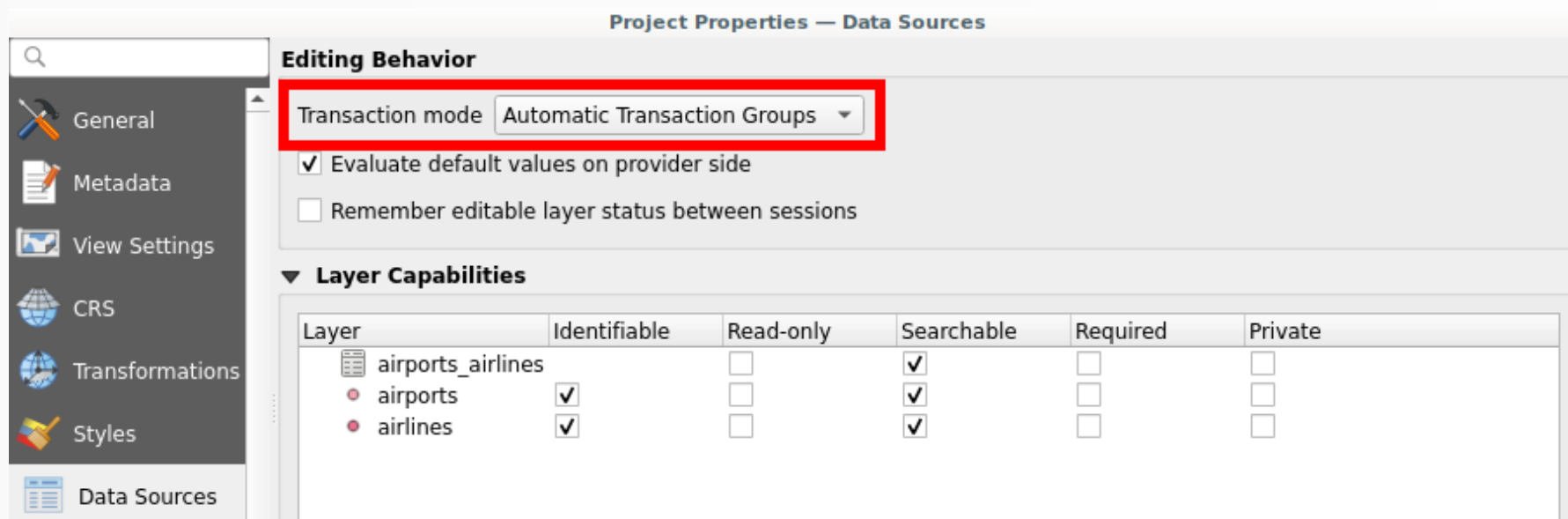
- Load all targeted layers
- *Project > Properties > Relations*
- *Discover Relations*
- Select all > *OK*

Evaluate primary keys



- Relations need valid PKs, not temporary ones
 - e.g. `nextval('locations.airports_id_seq'::regclass)`
- ⚠ Not enough...
 - Edit all concerned layers
 - Save in the appropriate order (referenced first!)
 - What if errors!

Automatic transaction groups



- Open a DB transaction → Close it on save/finish edit
- All layers from database edited at once!
- 👍 Evaluate triggers
- 👎 Long living transaction → [caveats](#)
 - lock on modified rows → block other transactions

Buffered Transaction Group ★ New in 3.26



Project Properties — Data Sources

Editing Behavior

Transaction mode: Buffered Transaction Groups

Evaluate default values on provider side

Remember editable layer status between sessions

Layer Capabilities

Layer	Identifiable	Read-only	Searchable	Required	Private
airports_airlines		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
airports	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
airlines	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- All modifications buffered and replayed on save
- in the appropriate order
- in one transaction
- 🙅 Don't evaluate triggers live

Form configuration

Layer Properties — airports — Attributes Form

Drag and Drop Designer

Show Form on Add Feature (global settings)

Available Widgets

- Fields
 - 123 id
 - abc airport_name
- Relations
 - airports airlines airport_fk_fkey
- Actions
- Other Widgets
 - QML Widget
 - HTML Widget

Form Layout

- id
- airport_name
- airports airlines airport_fk_fkey

Widget Display

- Show Label
- Override Label Color
- Override Label Font
- Relation**
 - Label:
 - Cardinality: airlines (id)
 - Force hide form on add feature
 - Widget Type: Relation Editor
- Widget Configuration**
 - Automatically select first child feature and show attribute form
- Toolbar buttons**
 - Link child feature
 - Unlink child feature
 - Save child layer edits
 - Add child feature
 - Duplicate child feature
 - Delete child feature
 - Zoom to child feature

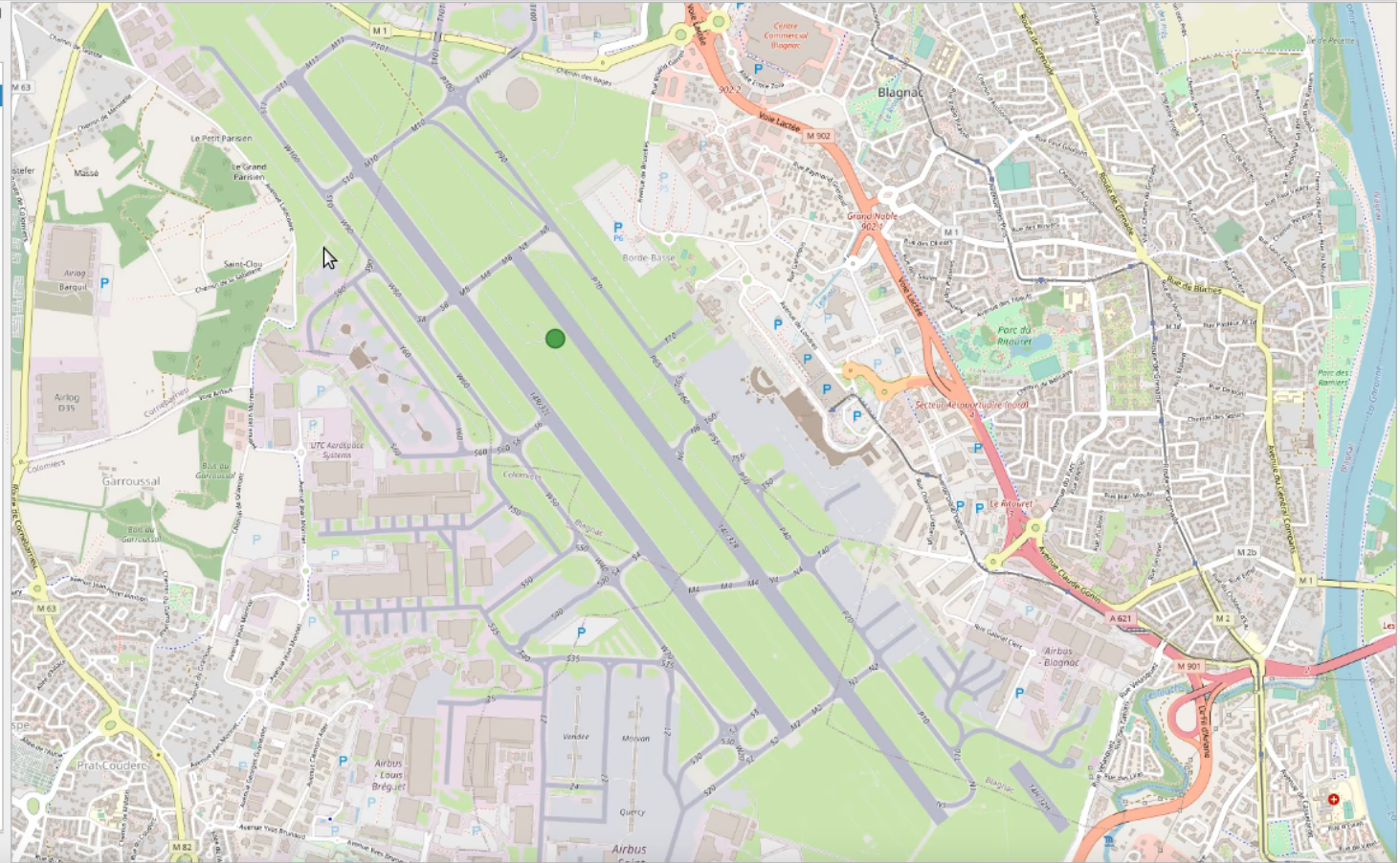
Apply Cancel OK



DataManagement

Layers

- airports airlines
- airports
- airlines
- OpenStreetMap



Layers Browser

Type to locate (Ctrl+K)

Coordinate 148313,5409578 Scale 1:26602 Magnifier 100% Rotation 0.0° Render EPSG:3857

0:00 / 0:43



PostgreSQL Notify

- Trigger a layer refresh on PostgreSQL "event"
- Insert, Update, Delete, ...

PostgreSQL configuration

```
CREATE FUNCTION notify_points_updated() RETURNS trigger
LANGUAGE plpgsql
AS $$
BEGIN
NOTIFY qgis, 'points updated';
RETURN NULL;
END;
$$;

CREATE TRIGGER notify_points_updated
AFTER INSERT OR UPDATE OR DELETE OR TRUNCATE ON my_points
FOR EACH STATEMENT EXECUTE PROCEDURE notify_points_updated();
```

QGIS configuration

The screenshot shows the 'Layer Properties' dialog for a layer named 'my_points (défaut)'. The 'Rendering' tab is active. The 'Scale Dependent Visibility' section is expanded, showing a minimum scale of 1:100000000 and a maximum scale of 0. The 'Simplify Geometry' section is also expanded, with a simplification threshold of 1.00 pixels and a distance algorithm. The 'Fixed Reference Scale' section is expanded, showing a reference scale of 1:1218542. The 'Refresh layer on notification' checkbox is checked and highlighted with a red box, with the message 'points updated' entered in the adjacent text field. Other options include 'Force layer to render as a raster', 'Refresh layer at interval (seconds)', and 'Only if message is'. The 'Apply', 'Cancel', and 'OK' buttons are visible at the bottom right.

Layer Properties — my_points (défaut) — Rendering

Scale Dependent Visibility

Minimum (exclusive) 1:100000000 Maximum (inclusive) 0

Simplify Geometry

Note: Feature simplification may speed up rendering but can result in rendering inconsistencies

Simplification threshold (higher values result in more simplification) 1.00 pixels

Simplification algorithm Distance

Simplify on provider side if possible

Maximum scale at which the layer should be simplified (1:1 always simplifies) 1:1

Fixed Reference Scale

If set, the reference scale indicates the map scale at which symbology and labeling sizes which uses paper-based units (such as millimeters or points) relate to. The sizes will be scaled accordingly whenever the map is viewed at a different scale.

For instance, a line layer using a 2mm wide line with a 1:2000 reference scale set will be rendered using 4mm wide lines when the map is viewed at 1:1000.

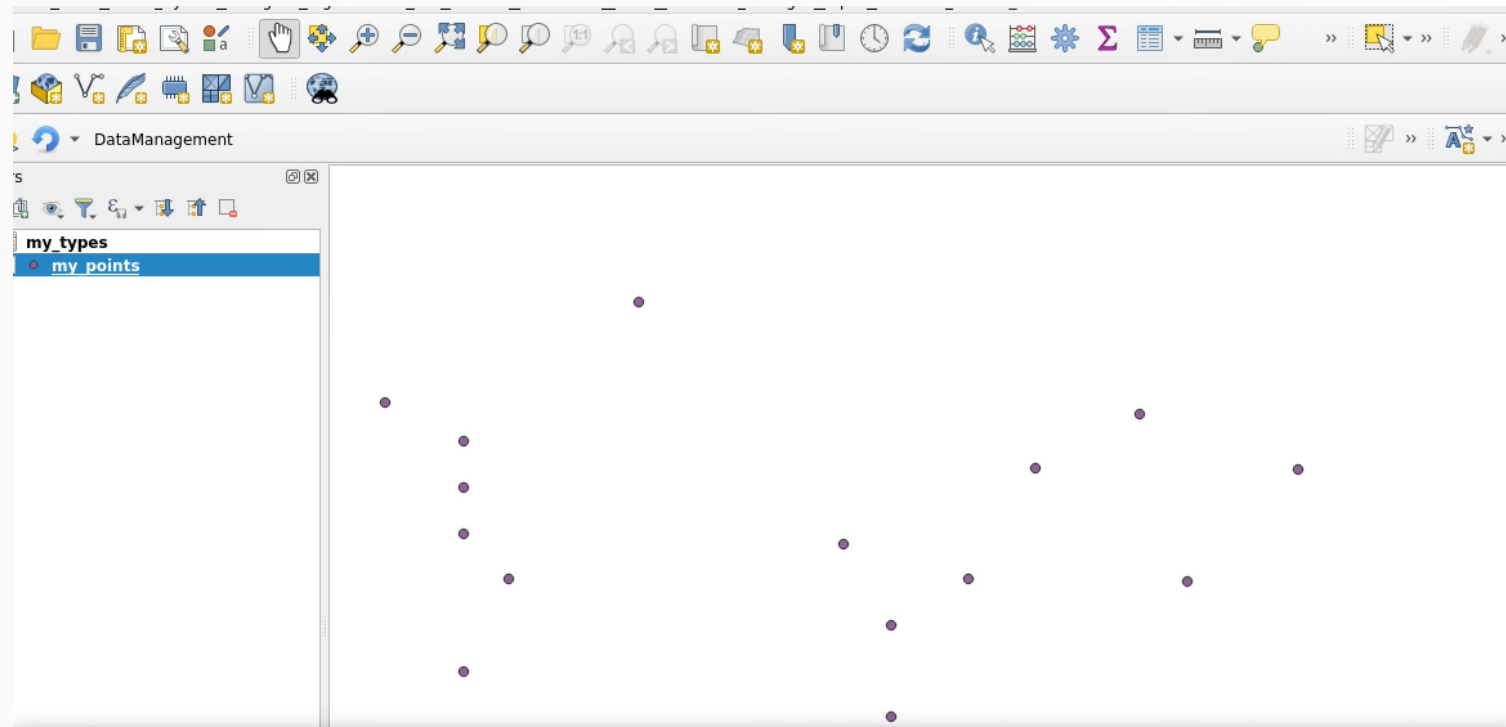
Reference scale 1:1218542

Force layer to render as a raster (may result in smaller export file sizes)

Refresh layer at interval (seconds) 0.00

Refresh layer on notification Only if message is points updated

Help Style Apply Cancel OK



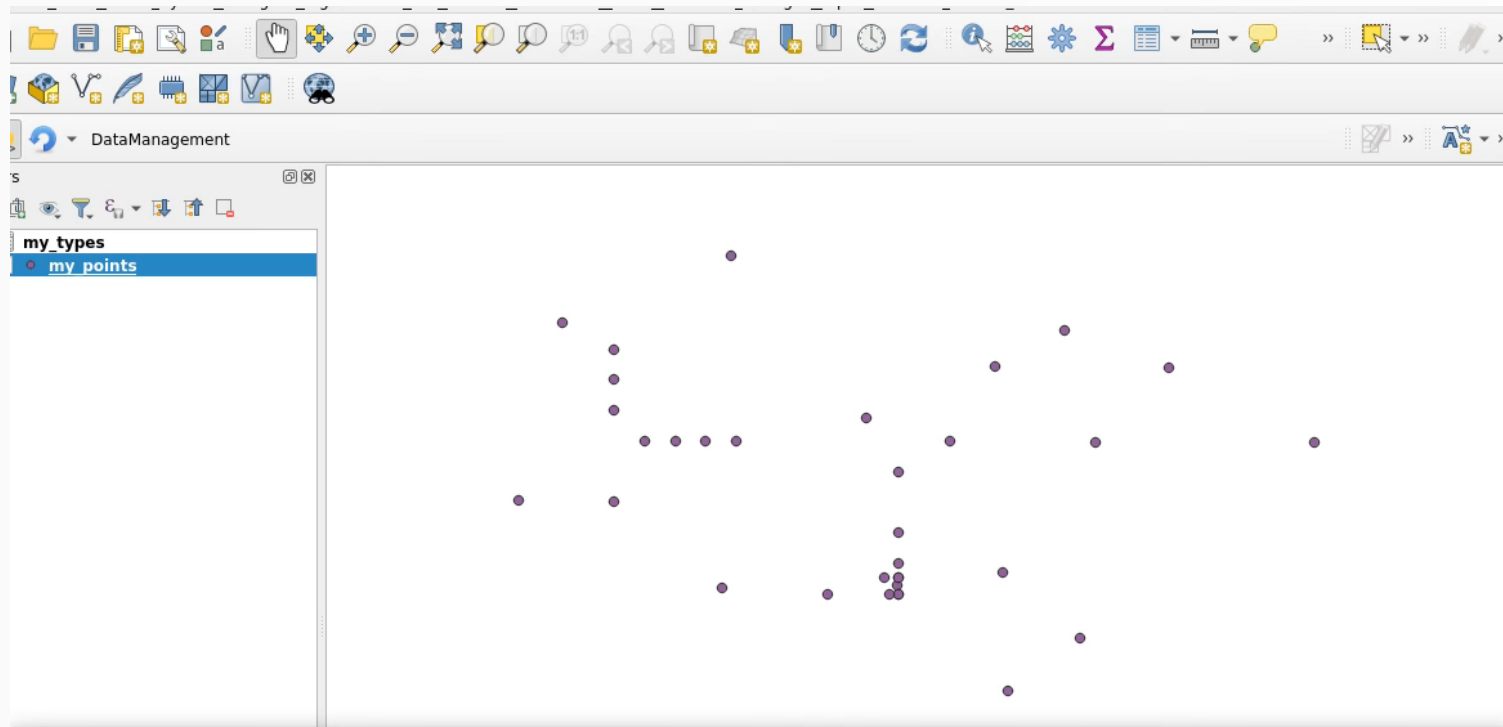
```
psql test_val_rel
Fichier  Édition  Affichage  Rechercher  Terminal  Onglets  Aide
reveal-md . --theme /oslandia.css --watch x julien@julien-laptop:~/work/Communication/ev... x psql test_val_rel x +
→ ~ psql test_val_rel
psql (14.3 (Debian 14.3-1))
Saisissez « help » pour l'aide.

test_val_rel=# insert into my_points (geom) values (st_setsrid(st_makepoint(-0.9,0.1),4326));
INSERT 0 1
test_val_rel=#
```

Not only refresh

- **How ?** With a little bit of Python magic
- **Where ?** Plugin / Startup script / Macros / Python console
- **To do what ?** Anything from QGIS API

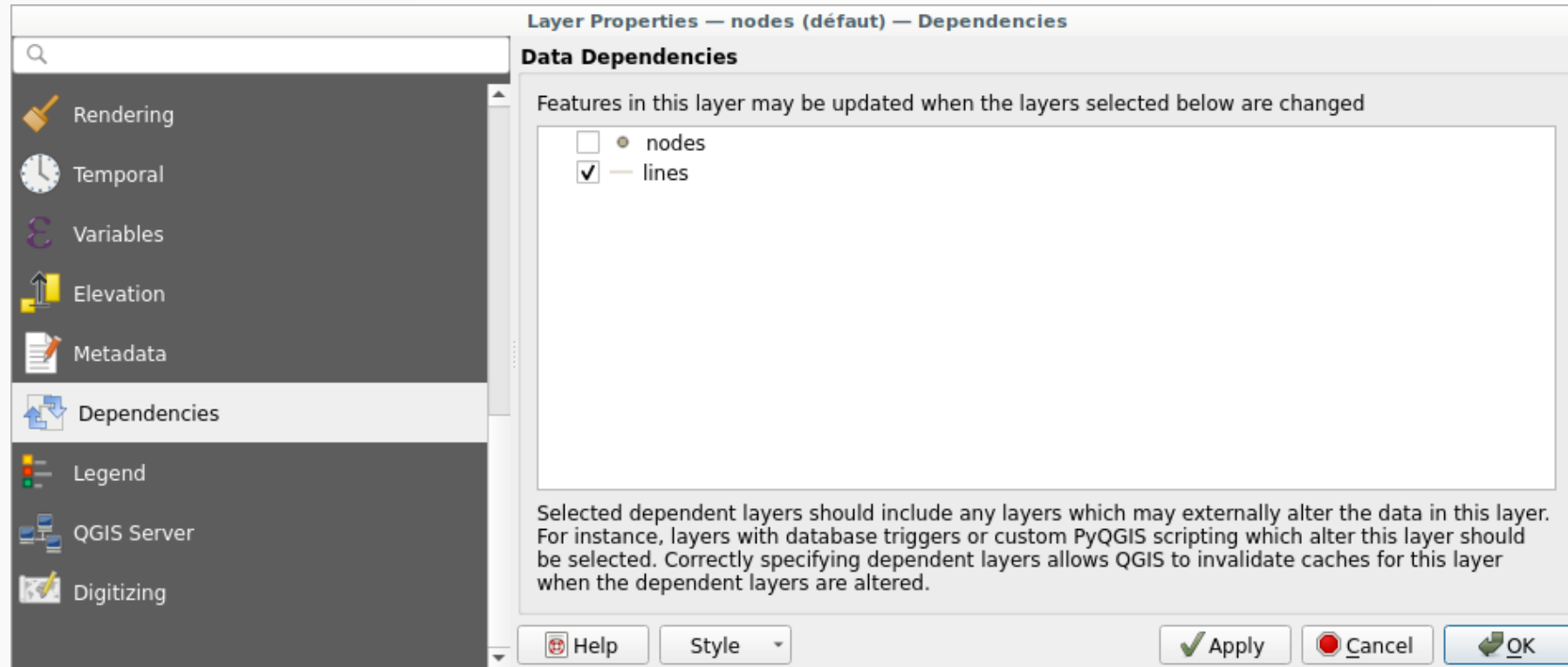
```
def onNotify():  
    iface.messageBar().pushInfo("Notify", "Points updated")  
  
iface.activeLayer().dataProvider().notify.connect(onNotify)
```



```
psql test_val_rel  
Fichier  Édition  Affichage  Rechercher  Terminal  Onglets  Aide  
reveal-md . --theme /oslandia.css --watch  x julien@julien-laptop:~/work/Communication/ev...  x psql test_val_rel  x +  
→ ~ psql test_val_rel  
psql (14.3 (Debian 14.3-1))  
Saisissez « help » pour l'aide.  
test_val_rel=#
```

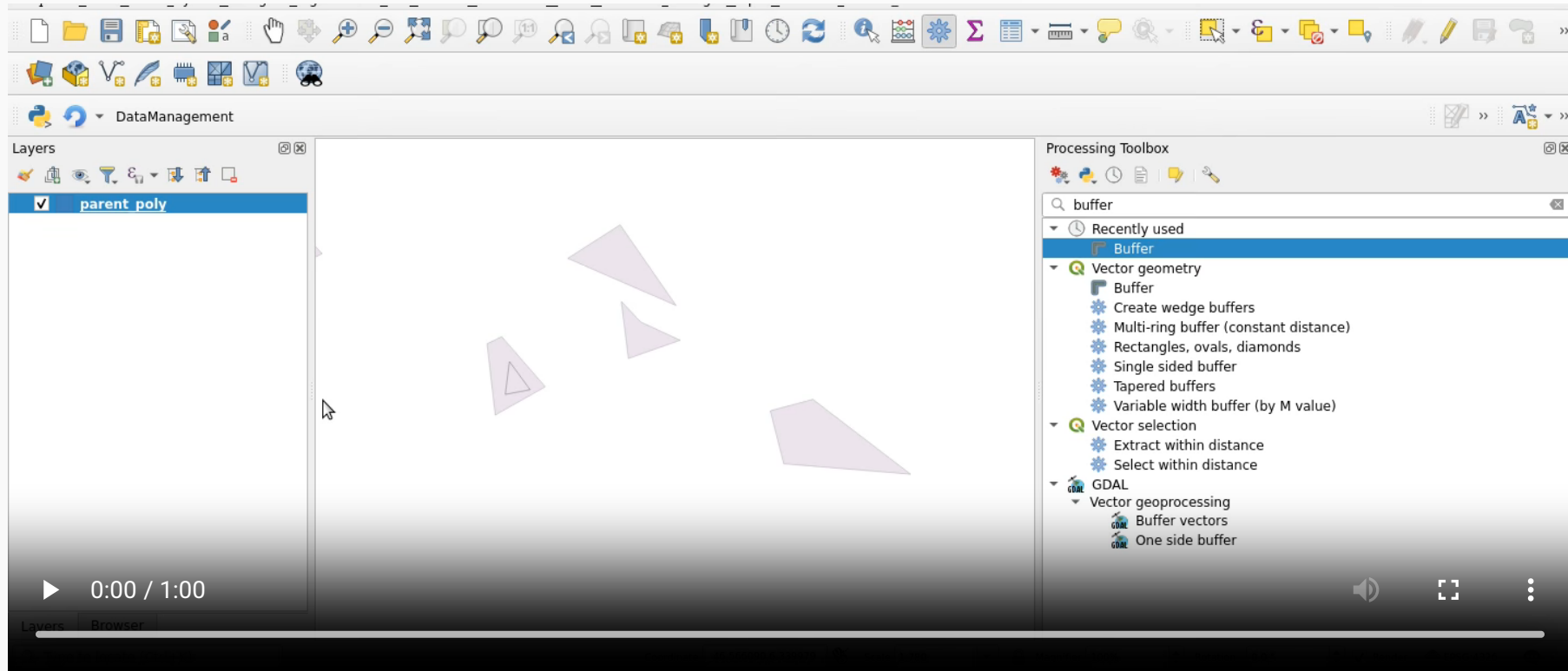

Refresh other layers ?

- **Use case** : Refresh nodes according to intersecting lines using triggers
- **NO !!** : Prefer **data dependencies**

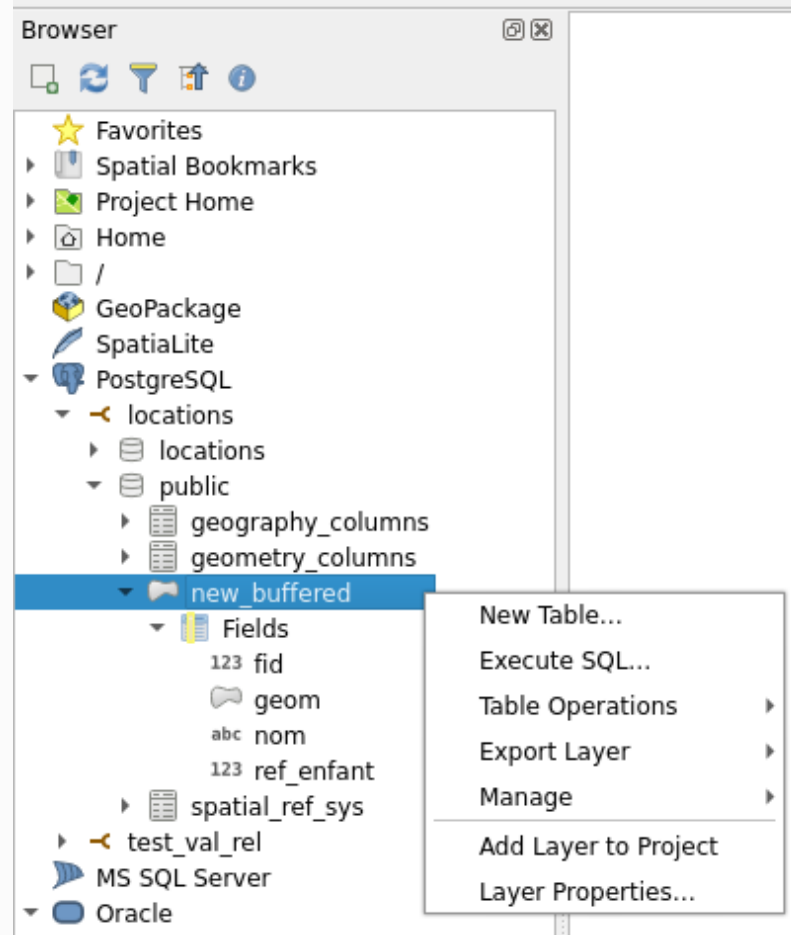


Processing to database

Output processing result directly to database



Managing DB from browser



You can create table

New Table

Schema:



Name:

Name	Type	Provider type	Length	Precision	Comment
123 id	Whole Number (smallint - 16bit)	int2	-1	0	
abc text	Text, unlimited length (text)	text	-1	-1	

Geometry type:

Geometry column name:

Dimensions Include Z dimension Include values (M)

CRS:  

Create spatial index

but also ...

- Create/Rename/Delete schema
- Create/Rename/Delete/Truncate table
- Add/Delete field
- Export to file

... and execute SQL

new_buffered — Execute SQL

```
SELECT * FROM "public"."new_buffered" LIMIT 10
```

Clear Fetched rows: 10/10 1 ms Execute Stop

	fid	geom	nom	ref_enfant
1	3	0106000020...	titi	
2	4	0106000020...	{1.2}	

▼ Load as new layer

Column(s) with unique values

Geometry column

Subset filter ...

Avoid selecting by feature ID

Layer name

Load Layer

Close

Does it replace DB Manager ?

- Yes ... but **not completely**
- Some features are missing
 - Create/Manage constraint
 - Create view from query
 - Modify a column
 - Logging table (Is someone using this ?!!)
 - ...

DB Manager

- No use of the QGIS provider API
- Not-well-tested

Prefer the browser feature everytime possible!

Save project in database

Enable option

Edit PostGIS Connection

Connection Information

Name:

Service:

Host:

Port:

Database:

SSL mode:

Authentication

Configurations: **Basic**

Choose or create an authentication configuration

Configurations store encrypted credentials in the QGIS authentication database.

Only show layers in the layer registries

Don't resolve type of unrestricted columns (GEOMETRY)

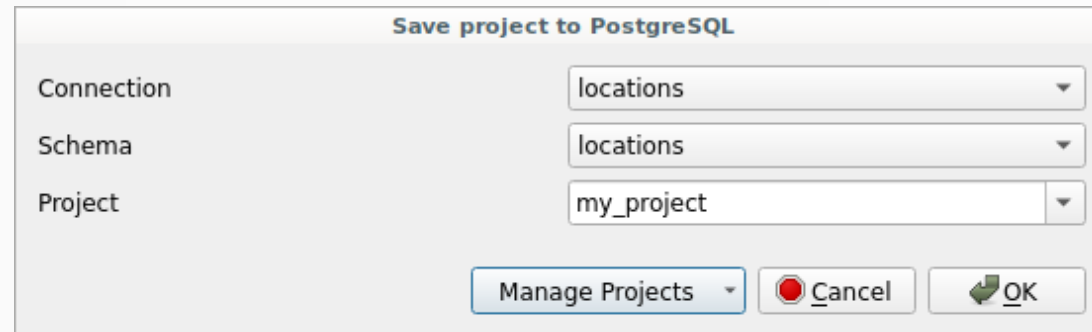
Only look in the 'public' schema

Also list tables with no geometry

Use estimated table metadata

Allow saving/loading QGIS projects in the database

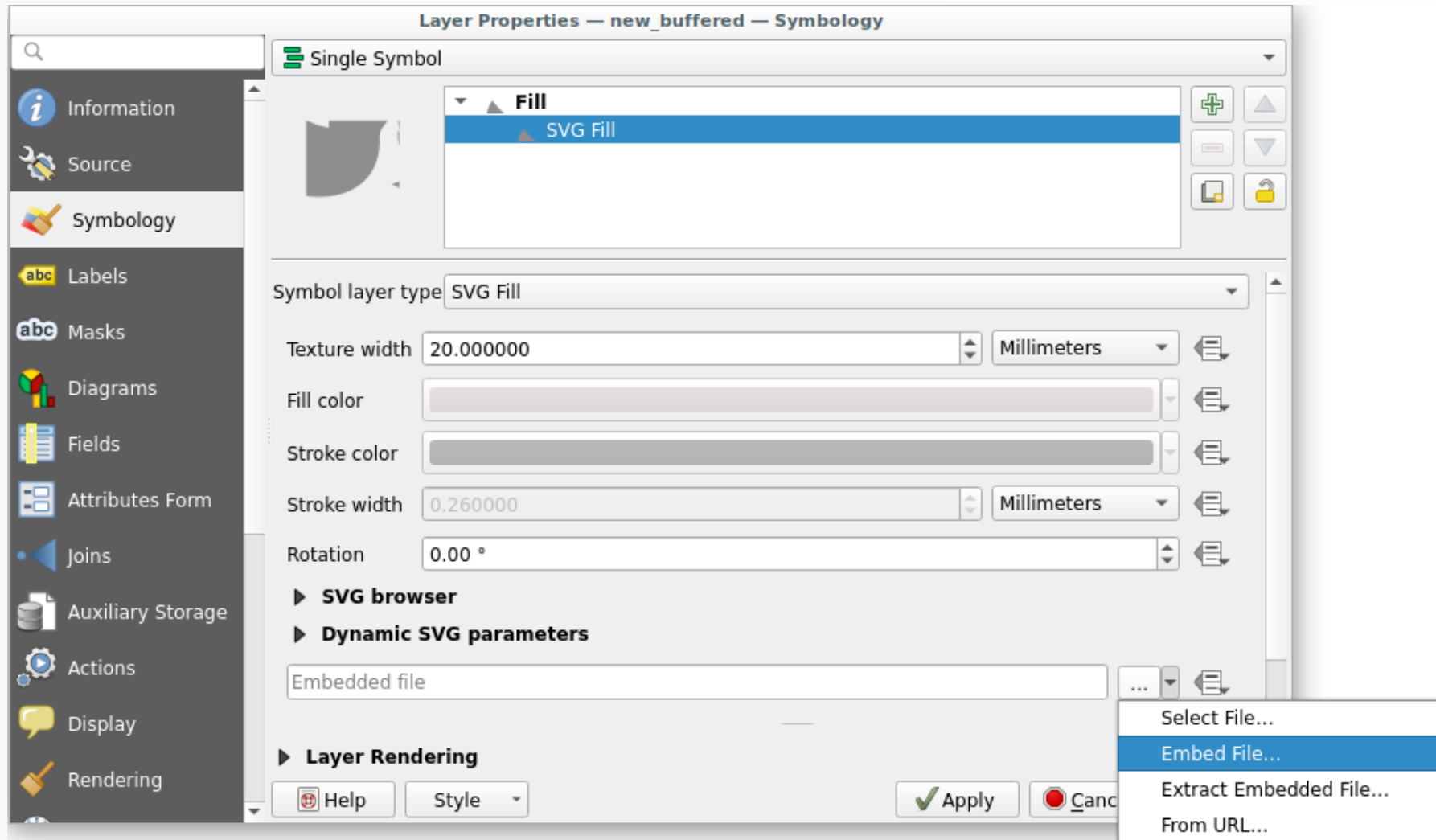
Project > Save To > PostgreSQL...



The screenshot shows a dialog box titled "Save project to PostgreSQL". It contains three dropdown menus for selection: "Connection" is set to "locations", "Schema" is set to "locations", and "Project" is set to "my_project". At the bottom of the dialog, there are three buttons: "Manage Projects" (with a dropdown arrow), "Cancel" (with a red stop icon), and "OK" (with a green checkmark icon).

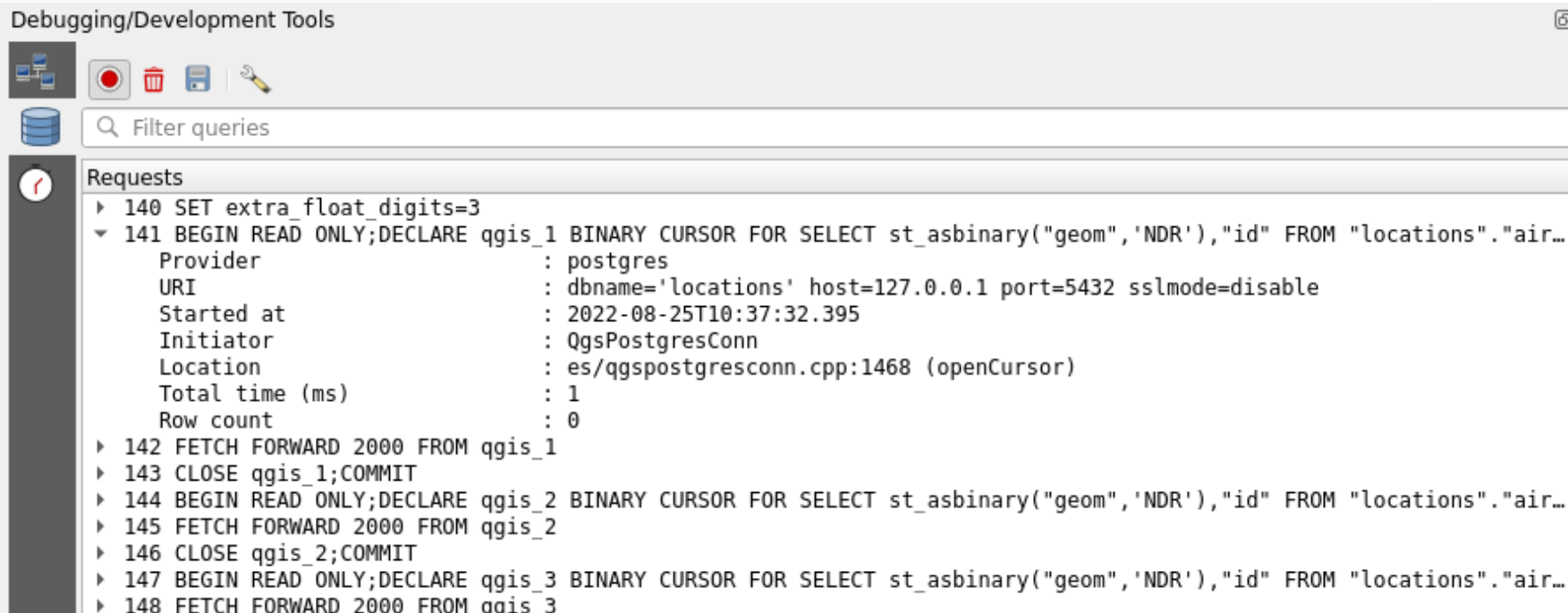
Project > Open From > PostgreSQL...

External files



- You can embed external file in project (SVG for instance)

SQL Logging ★ New in 3.26 ★



The screenshot shows the 'Debugging/Development Tools' panel in QGIS. It features a toolbar with icons for running, deleting, saving, and refreshing. Below the toolbar is a search bar labeled 'Filter queries'. The main area is titled 'Requests' and displays a list of SQL queries and their execution details. The log shows several queries, including setting float digits, opening binary cursors, and fetching data from a table named 'air...'. The details for the first cursor opening query are expanded, showing metadata like provider, URI, start time, and location.

```
Debugging/Development Tools
Filter queries
Requests
▶ 140 SET extra_float_digits=3
▼ 141 BEGIN READ ONLY;DECLARE qgis_1 BINARY CURSOR FOR SELECT st_asbinary("geom",'NDR'),'id" FROM "locations"."air...
    Provider          : postgres
    URI                : dbname='locations' host=127.0.0.1 port=5432 sslmode=disable
    Started at        : 2022-08-25T10:37:32.395
    Initiator         : QgsPostgresConn
    Location          : es/qgspostgresconn.cpp:1468 (openCursor)
    Total time (ms)   : 1
    Row count         : 0
▶ 142 FETCH FORWARD 2000 FROM qgis_1
▶ 143 CLOSE qgis_1;COMMIT
▶ 144 BEGIN READ ONLY;DECLARE qgis_2 BINARY CURSOR FOR SELECT st_asbinary("geom",'NDR'),'id" FROM "locations"."air...
▶ 145 FETCH FORWARD 2000 FROM qgis_2
▶ 146 CLOSE qgis_2;COMMIT
▶ 147 BEGIN READ ONLY;DECLARE qgis_3 BINARY CURSOR FOR SELECT st_asbinary("geom",'NDR'),'id" FROM "locations"."air...
▶ 148 FETCH FORWARD 2000 FROM qgis_3
```

- *View > Panels > Debugging/Development Tools*
- Log ALL SQL requests (select, insert, update...)
- Log also HTTP requests

Questions ?

FOSS4G 2022 - Firenze - 26/08/2022



Julien Cabieces - Oslandia

 @troopa81

 @CabiecesJ