Hands-On “PostGIS - Geographic Data Analysis for Smart Cities”

Adelson Araújo Júnior (SmartMetropolis & DCA/UFRN).

1. Preparing a database

    a. Download PostGIS extension with pgAdmin’s StackBuilder.
    b. Create a database, here called *handson*
    c. Go to *handson* then right click to New Extension
    d. In “name” field, select for postgis
    e. Now you can run PostGIS functions in your database

2. Importing data to PostGIS

    a. CSV files

For “comma separated values” file (.csv), your data must have at least georreferenced position and the Spatial Reference Identifier (SRID) that describe it correctly. There, we will import escolasNatal.csv, with points representing schools in Natal, Brazil.

You can run SQL query written in q0q1.txt for “point” based geometries with FILE. There is an CREATE TABLE and CREATE INDEX statement, a COPY (you must adjust your csv file) statement to put data into the table, an UPDATE statement to create geometry field with georreferenced position (latitude, longitude).

Finally, if you want every data in the same SRID, you can do this with the last piece of SQL, with ST_Transform(geom,newSRID). In future, probably you will want to cross two geometries and you must have both with the same SRID. Transform it will changes completely the data arrangement, but if you just want to change the value itself of SRID, without any transformation or convertion, use UpdateGeometrySRID(table,geom,newSRID). All of these have documentation on postgis.net. To check SRID of a geometry column, try:

```
SELECT ST_SRID(geom) FROM escolasnatal LIMIT 1
```

With next queries, it will be important to relate escolasnatal with others geometries that are in SRID 31985. Please execute the last query in the file to transform escolas natal from 4326 (lat,lon) to 31985 (UTM).

b. SHP files

To import shapefiles, it becomes a little bit easier if you don't have character encoding problems in the process. Let's check it out with bairrosNatal.rar. First, extract it.

First, in pgAdmin plugins, we must have the PostGIS Shapefile and DBF Loader. Set connections details to your database. In Options.. you have some configs that you may have to adjust depending the case. After that, click in Add File and select your shp. Then, change the SRID to 31985 (it's the right arrangement of this dataset) and enter. Finally, click Import. If you have problems with character encoding, you can change it in Options.. setting the right encoding that your data is using. In this case, try LATIN1 if the problem appears.

3. Exploring Spatial Joins Concepts

a. Knowing spatial join

To relate two geometry tables, you'll need a spatial join. Let's show some of main possibilities.

- ST_Intersects(geom,geom): Returns TRUE if a geometry shares a portion of it's shape in other geometry.
- ST_DWithin(geom,geom,d): Returns TRUE if the geometries are within the specified d distance of one another.
- ST_Contains(geom1,geom2): Returns TRUE if a geometry2 is fully inside geometry1.

b. Basic structure of a spatial query (q2q3.txt)

```
--Q2: Points inside a polygon (schools inside a neighborhood)
SELECT e.gid, e.nome, b.bairro
FROM escolasnatal e
JOIN bairrosnatal b ON ST_Intersects(e.geom,b.geom)
WHERE b.bairro = 'Cidade Alta'
```

And with some of creativity, more complex queries are possible...

```
--Q3: Density points per area unit inside a polygon (schools/km²)
SELECT count(e.*)*1000000/ST_Area(b.geom)
FROM escolasnatal e
JOIN bairrosnatal b ON ST_Intersects(e.geom, b.geom)
WHERE b.bairro = 'Cidade Alta'
GROUP BY b.geom
                                        --ST_Area returns square meters
```

4. Going beyond

    a. Inserting polygon identifier on points table (representing which polygon contains that point).

    Example: With points as schools, and polygons as neighborhoods, add in schools the column that represent in which neighborhood it is located.

q4.txt

    b. Measuring how far the points are from each other.

    Example: With points as schools, how far are schools from each other in a specific neighborhood. (possible result: 2000m).

q5.txt

    c. Clustering points.

    Example: With points as schools, what are good clusterings of these, considering the parameters.

q6.txt

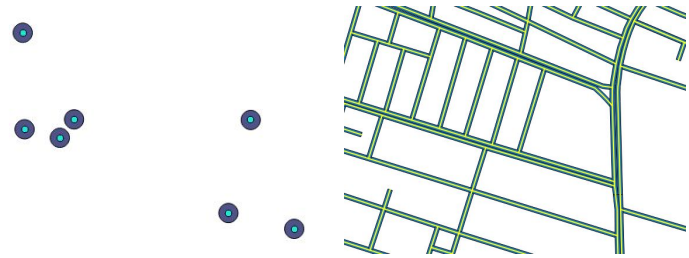
    *Is recommended to open query's result table into QGIS (See next section).

d. Inserting buffer in points or lines.

Example: Suppose points as schools and lines as streets (datasetsq/logradourosnatal.rar contains streets shapefile). They never will really intersects, because of the infinitesimal width of them. You must add a buffer to one of these to check intersection or other joins.

q7q8.txt

*Remember to import logradouros' shapefile with SRID 31985 before Q8.



e. Measuring the percentage of lines close to points.

Example: With lines as streets (logradourosnatal.rar dataset) and points as schools, how many % of streets does have schools on a maximum of 50 meters of distance? (possible result: 12% of streets does have school in a 50 meters range distance).

q9.txt

5. Visualizing geometry tables

a. QGIS

To visualize your geometries, QGIS do connections with PostGIS databases and show them in easy steps. First, install it. Then, follow the instructions on next figures.

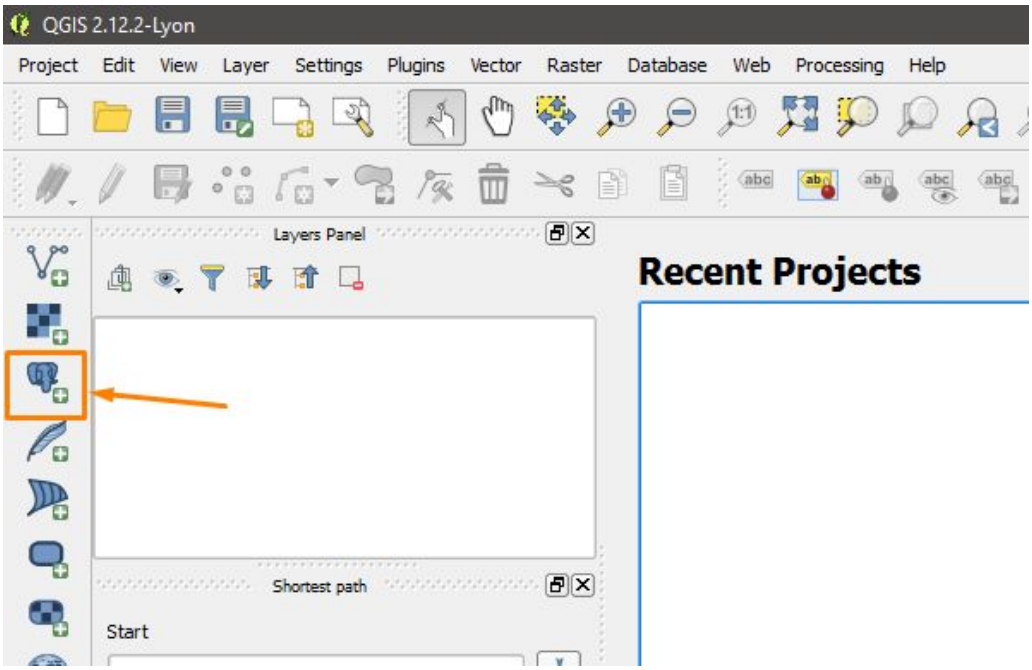


Figure 1 - Click to add a PostGIS layer and click New button.

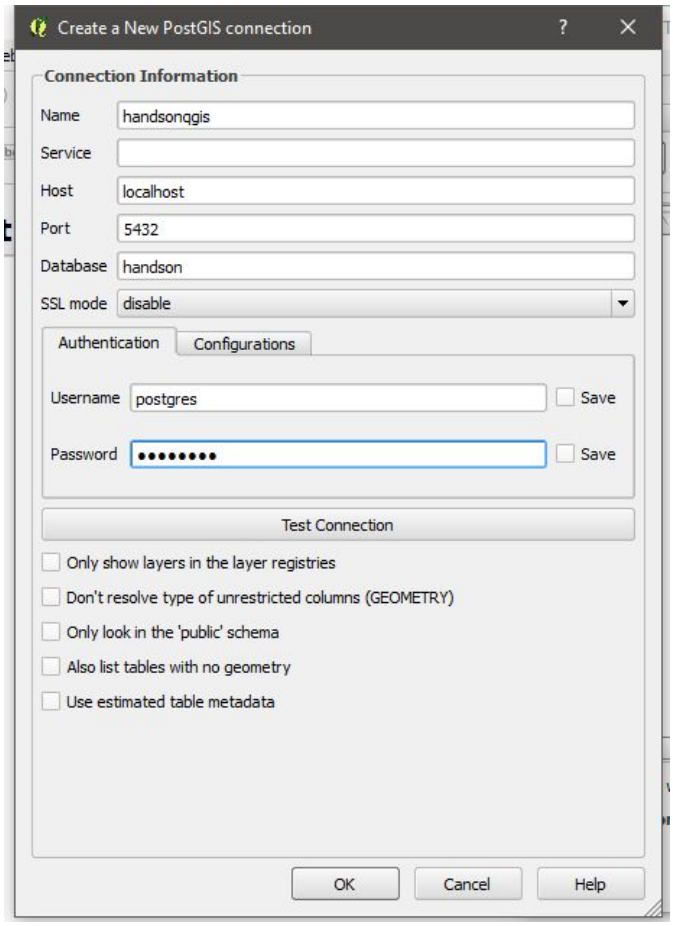


Figure 2 - Configuring a connection with our handson database, created before. After the ‘OK’, click Connect. using your postgres user and password.

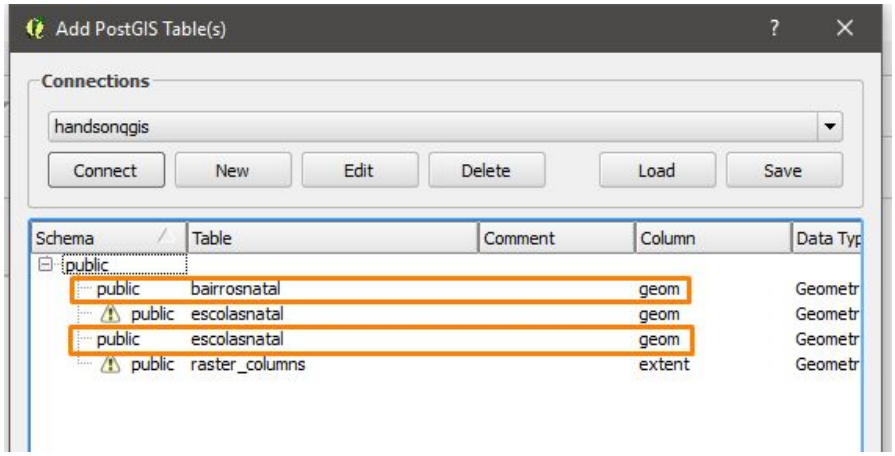


Figure 3 - Select your geometries' tables and click Add.

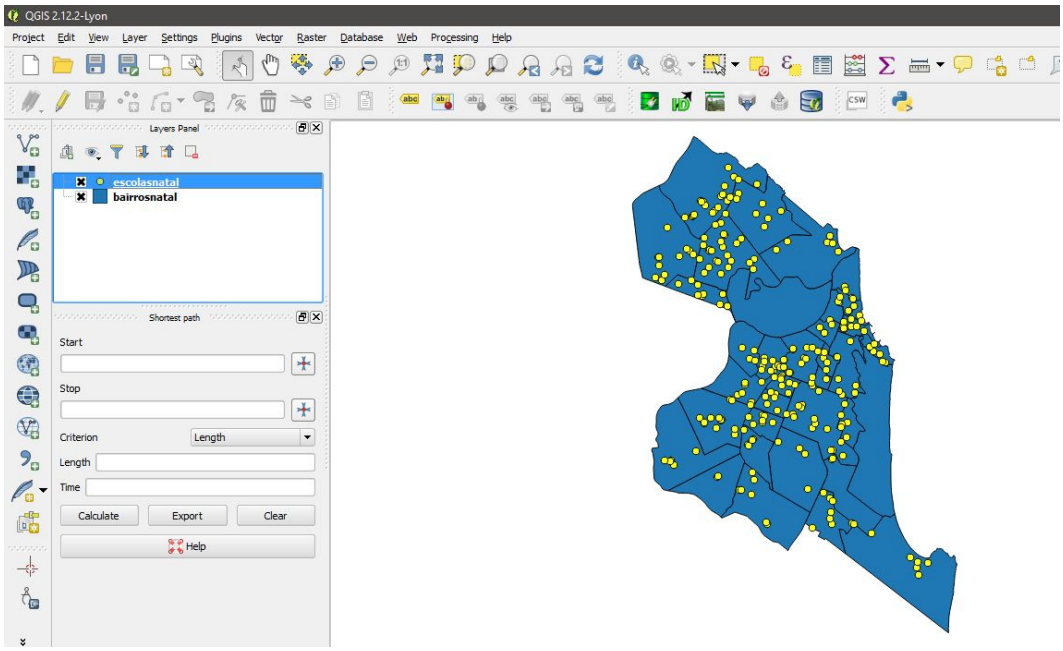


Figure 4 - Done! In Layers Panel, you can order which layer would be on top by drag it up. Double click on the layer will open a properties window.