Section : 2. Data Providers

Module : 2.5. PostGIS Overview

Introduction to PostGIS

“PostGIS is a spatial extension for the PostgreSQL relational database management system”

PostgreSQL is an open source Relational Database Management System (RDBMS), which stores data in a series of related tables. Each row in a table constitutes a record, and each record can be related to other records or tables in various ways. These relationships can be queried so that a subset of the records are returned as a response, which is filtered by the supplied query criteria.

PostGIS is an extension for PostgreSQL which introduces spatial functionality. This includes the ability to store a records information with its geometry data, as well as additional functionality such as support for various coordinate reference systems. In addition, PostGIS allows the queries to the database to use spatial analysis in the criteria and response, so that one could, for example, request all the positions that are found within a certain geographic region, or return a response with the feature geometry, such as returning a point object from the latitude and longitude positions in a table.

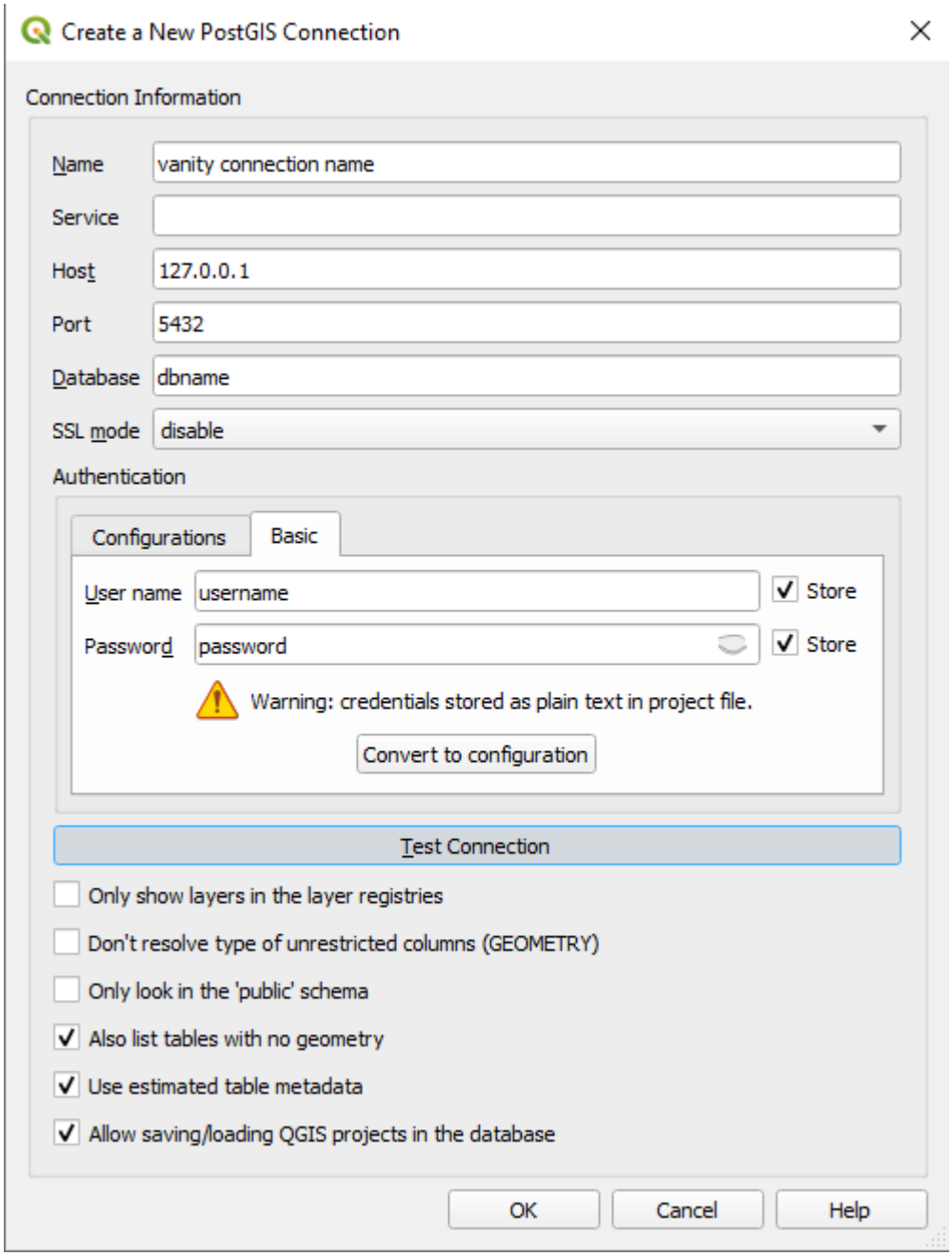
"PostGIS", "Postgres" and "PostgreSQL" are terms that are often used interchangeably to refer to a PostgreSQL database that has PostGIS functionality.

The use of database data management systems such as PostGIS offer many benefits over traditional flat-file storage options, including:

- Access control and permissions management
- Workflow and quality assurance control
- Performance, data resilience, and integrity control
- Complex data modeling
- Shared datastores, multi-user environments, and version management systems
- Advanced analysis, queries, publishing abilities, and additional service exposure
- Recording, data analysis, data visualisation, and more

Challenges to using these database systems are additional complexity, and performance of network services depending on the infrastructure architecture utilised.

In this exercise, we introduce users to PostGIS Connections in QGIS, which stores the information required for a user to connect to a database from within QGIS. With a connection established, users are then able to interact with PostGIS data in a similar manner to typical flat-file workflows such as using shapefiles and GPKG.



You try:

Course instructors should provide access to a publicly accessible PostGIS database connection and credentials for users to utilise for setting

You can also set up your own postgis database locally or by using a docker container, such as https://github.com/kartoza/docker-postgis

It is recommended for administrators to download the exercise data and load it into the database prior to the start of this exercise.

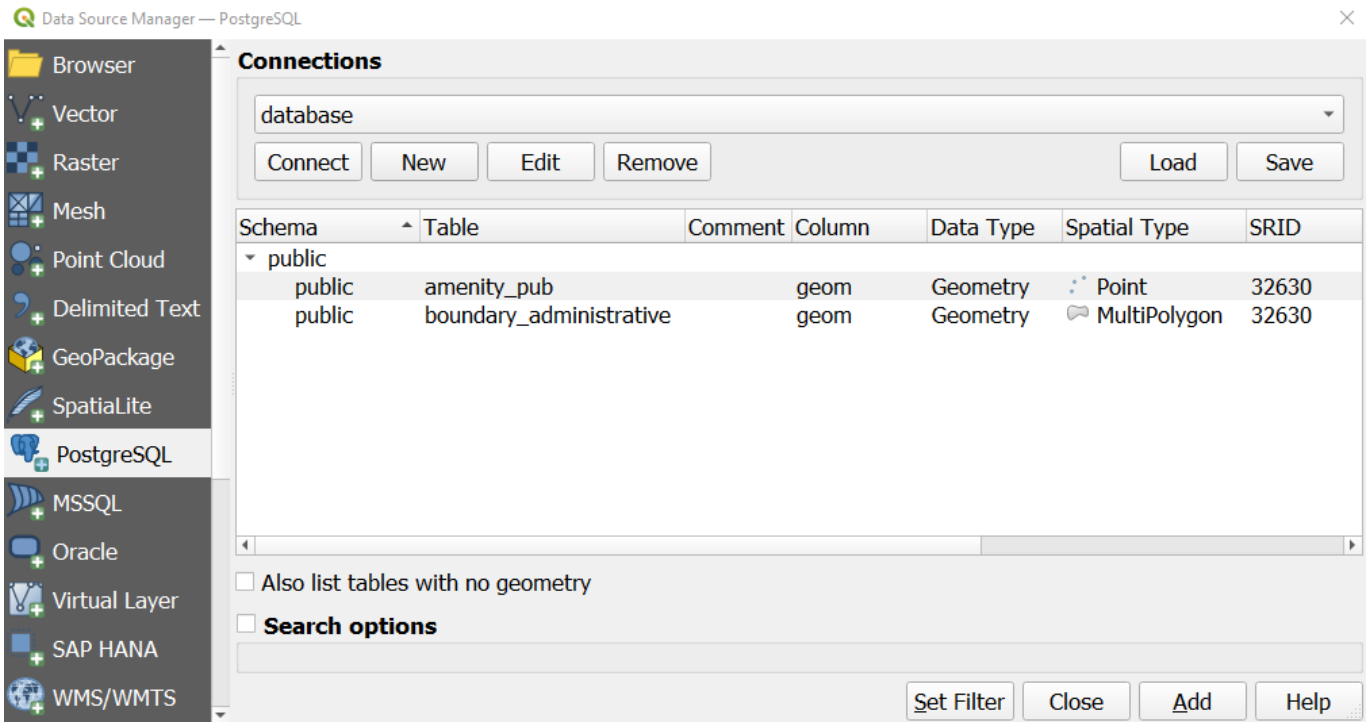
With the available connection credentials:

- Create a new PostGIS data provider connection.
- This may be done in the QGIS Browser Panel using the right click context menu on the PostGIS provider, using the New Connection option.

- It may also be configured from the data source manager under the PostgreSQL tab by adding a new connection

- Use the connection configuration to specify the database host, port, and database name
- The connection name is for your personal use and will not affect the connection
- Specify your connection credentials. Basic authentication will store your username and password in plain text in the QGIS project and is considered dangerous. Creating a new credential will store your authentication information in the auth.db of the current QGIS userprofile
- Take note of the additional options available in the connection configuration. Of particular interest should be the "Also list tables with no geometry" option and the check box to enable the ability to save QGIS projects within the database.
- Use the test connection button to ensure that the configuration is valid
- Once your connection has been established, view the available data in the database
- The QGIS Browser by expanding the connection under the PostGIS data provider
- Via the data source manager
- Note that establishing the connection and listing the available data may take some time. This is because the database connection is a *Network Service*

What happens when you try to load data from the database?
Can you figure out how to export and store data in the database?
What is the difference between the available layers when the "List tables with no geometry" options are used
Are you able to use the edit mode to update existing data in a database?
When traditional export options may not be available so you may need to use a geoprocessing tool such as *Export to PostgreSQL* to store new data in a database, or resort to using the DB Manager.



More about PostGIS

Historically, QGIS exposed advanced database functionalities via the *DB Manager* plugin. This is a core QGIS plugin and is installed by default and is available from the *Database>>DB Manager* menu item.

There have been significant efforts to incorporate the functionality of the DB manager into the QGIS browser panel, but at this time much of the complex functionality is available between these two features.

The database tools in QGIS allow users to import data from the filesystem into a database connection

For example, consider the following SQL expression:

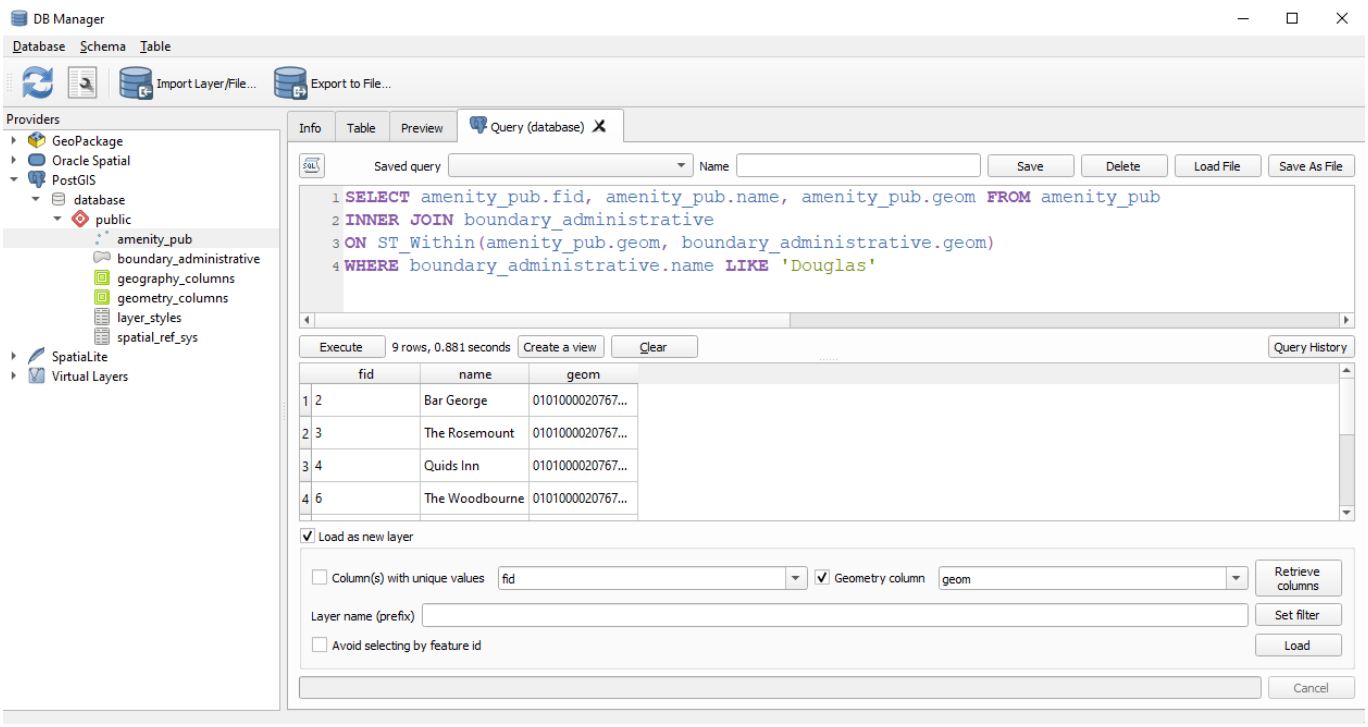
```
SELECT amenity_pub.fid, amenity_pub.name, amenity_pub.geom FROM amenity_pub
INNER JOIN boundary_administrative
ON ST_Within(amenity_pub.geom, boundary_administrative.geom)
WHERE boundary_administrative.name LIKE 'Douglas'
```

This will filter the pub points and only return those positions which are located geographically within the administrative area named Douglas.

If you inspect the filter options in the Data Source Manager or Layer Filter options, you may find that there is no way to access the properties of another layer. Filtering objects in this way would require additional geoprocessing steps, such as a spatial join.

Virtual Layers can be configured to handle this but it is more complex to handle external data sources which have not been loaded into the map.

By using the DB Manager, is is possible to filter the data at the provider level and only load the filtered subset of data that is required into the project.



Download the sample data for the lesson from http://changelog.qgis.org/media/images/lesson/worksheet/external_data/f9c432960b24610f512eda46a7023563e56545a8.zip.