
Table of Contents –Spatial DDL and DML

Part 1 – QGIS and PostGIS.....	2
Spatial References and EPSG Numbers.....	2
Exercise 1 – Familiarisation with QGIS and PostgreSQL/PostGIS	2
Exercise 2- Importing Spatial Data into PostgreSQL.....	8
Part 2- Data Management: Spatial Data	13
Exercise 1 – Spatial Data DDL and DML.....	13

Note: As Quantum GIS is Open Source, it tends to get updated very frequently. These updates often include interface changes which may move the menu options around. If you don't find the menu you need, spend a little time searching through the interface options until you find the required setting.

Note: QGIS 3.x is the latest version, but all practical notes are designed for 2.18 as this is the version that UCL has installed on its PCs.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Part 1 – QGIS and PostGIS

Spatial References and EPSG Numbers

The European Petroleum Survey Group (EPSG) number represents an internationally accepted code for projections (also known as coordinate reference systems). This can often be found associated with a Spatial Reference ID (SRID) that is also widely used but may vary from GIS to GIS, so where possible you should use the EPSG code (e.g. when importing data from QGIS into PostgreSQL).

- For Open Street Map datasets, as the data is collected using a GPS, the EPSG code required is: 4326 (WGS 84)
- For data from the Ordnance Survey in the UK, the data is in British National Grid, which has an EPSG of 27700

Projections/Coordinate Reference Systems in QGIS:

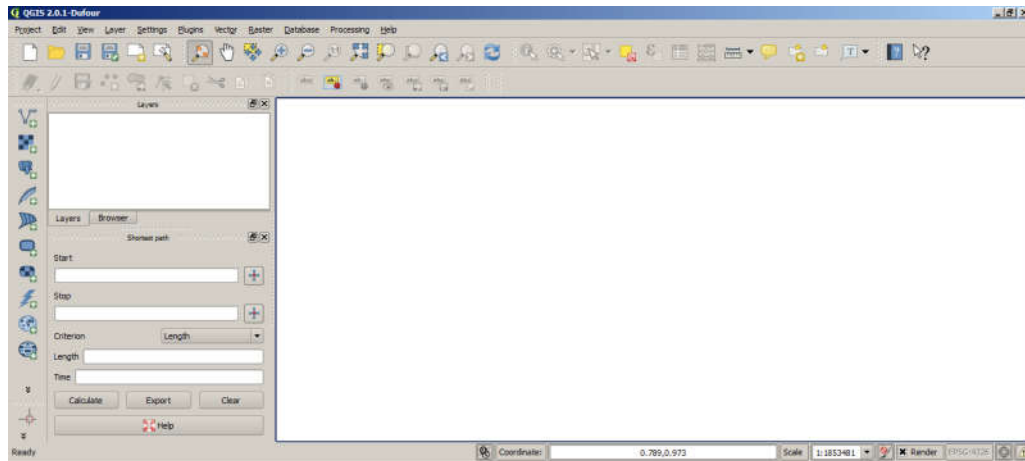
In QGIS, you need to set a default project projection before you add any data to the project. This will ensure that your data is shown correctly on the map. Exercise 1 will show you how to do this.

Exercise 1 – Familiarisation with QGIS and PostgreSQL/PostGIS

QGIS (Quantum GIS) is a free, open-source Geographical Information System which can be used to view, edit and analyse spatial data in multiple formats including ArcGIS shape files and MapInfo Tab files. It can also link directly to spatial data stored in PostgreSQL and display the data on a map. This is in contrast to the current version of ArcGIS (9.3) which requires an ArcSDE installation or a Data Interoperability license to view PostgreSQL spatial data.

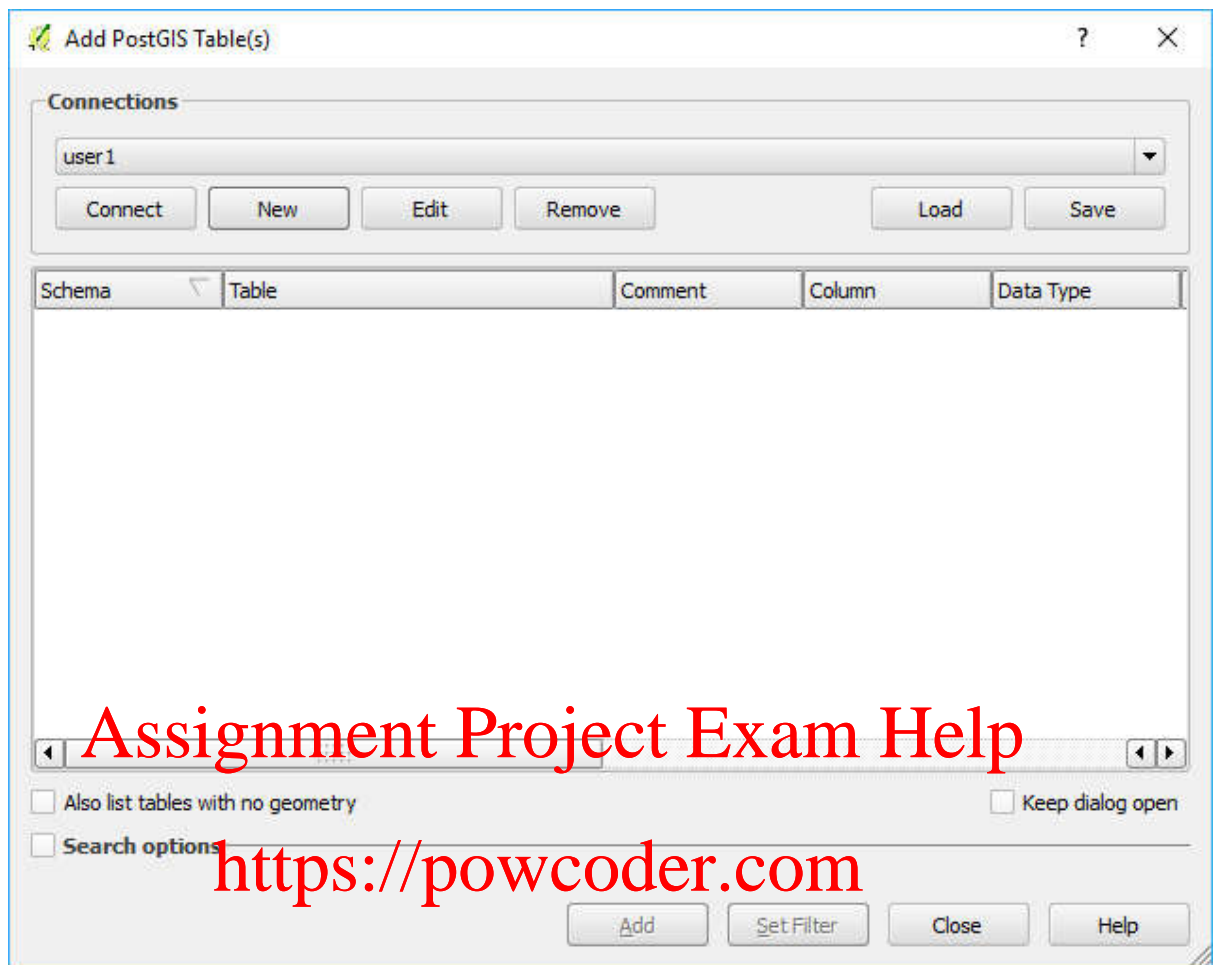
NB: For this module you are not expected to become QGIS or FME (see later weeks) experts – you just need to know how to connect to a database and display some map data.

1. Open QGIS. You will get the following basic interface:

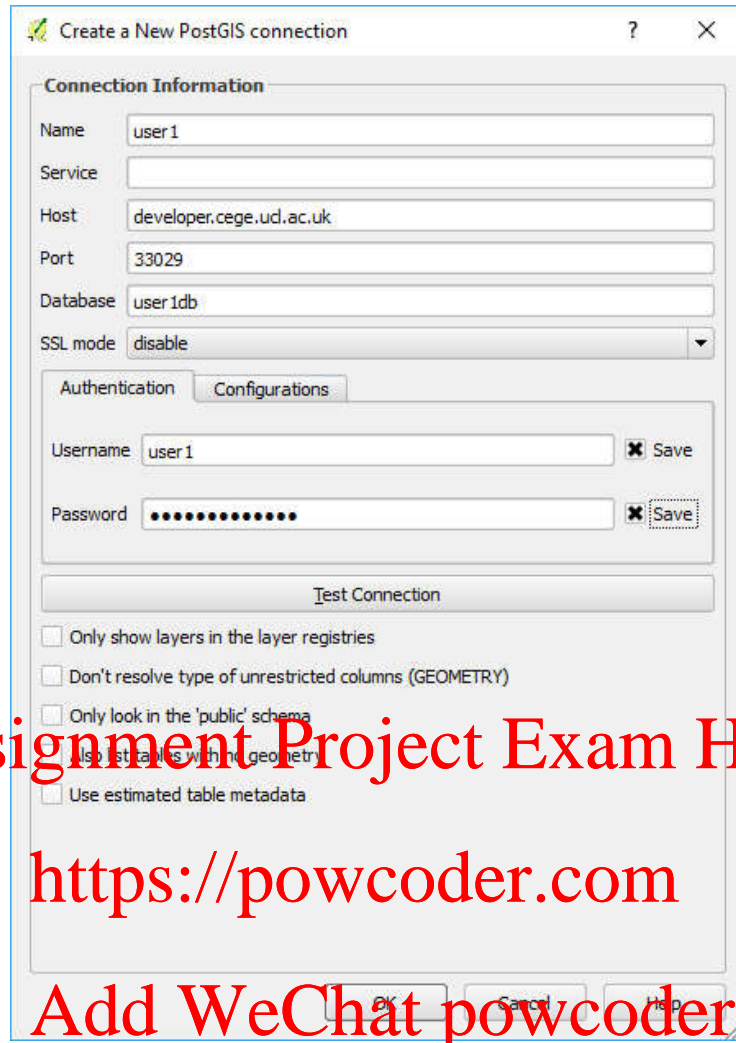


A list of layers will appear on the left hand side in the Layers Panel (sometimes at the top, sometimes at the bottom), with the map taking up the main bulk of the screen. A number of standard GIS tools are available on the tool buttons, as well as options to connect to various databases. Each button has an associated tool tip which pops up when you hover over the button, to tell you what the button does.

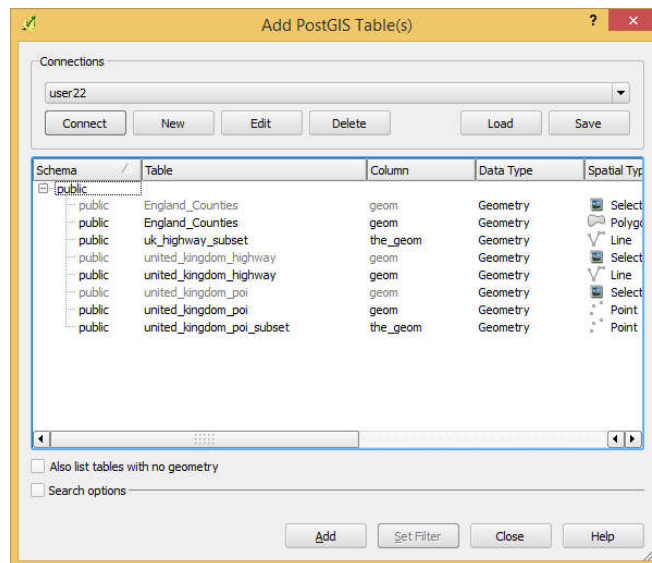
2. We start by creating a new project using PROJECT > NEW PROJECT. We must also set the default projection for the data in this project. As we will be using data from the United Kingdom, select the following options (see later on for a discussion about projection settings):
 - a. Click on PROJECT > PROJECT PROPERTIES
 - b. Click on COORDINATE REFERENCE SYSTEM (or CRS) to set the default projection
 - c. For UK data, select the OSGB 1936/British National Grid (EPSG 27700) option from the list (you can type 27700 in the filter to find this easily).
 - d. Make sure that the box marked 'Enable on-the-fly CRS transformation' is also checked. This will allow you to work with datasets in different projections.
 - e. Now click OK.
 - f. Save your project as practical_3.QGS
3. Now that we have created the project, we can add some data from our PostgreSQL database and using this to gain a basic understanding of QGIS. Create a connection to your database using the LAYER > ADD LAYER > ADD POSTGIS LAYER option



4. Click NEW to create the new connection
5. Enter your PostgreSQL connection details (developer.cege.ucl.ac.uk, the correct port, database and username and password details)



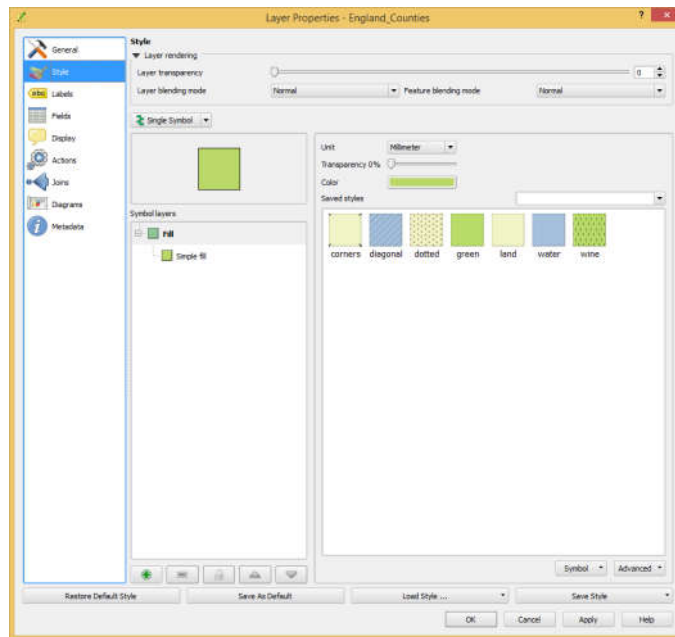
6. Save the username and password, then click OK to connect. You will get a warning as shown below – click OK again
7. Click on the LAYER > ADD POSTGIS LAYER menu option and click CONNECT. You will get a dialog similar to the following (NB: perhaps the table/layer list may be slightly different):



8. This interface can be used to connect to the database as follows:
 - a. If necessary, click on the **CONNECT** button and enter the password for your database OR CHANGE THE PARAMETERS AS REQUIRED. If you do not have an existing connection, you can create one using the **NEW** button.
 - b. Once you have done this, you will be given a list of map layers that are available in your database.
 - c. From this list, select *london_counties* and click **Add**. The layer will be added to your QGIS map.
9. Now spend some time exploring QGIS. Some example operations are as follows:
 - a. To zoom and pan around the map, use the tools similar to the following (the exact appearance may change between QGIS versions), which allow you to pan, zoom in, zoom out, zoom to extent, zoom to selection, zoom to layer, go back to the previous zoom and refresh the map.



- b. To change the display style of the map layer, double-click on the layer name. A layer properties dialog will appear as shown below.
 - c. Click on *style* to change the display style.



Edit data in QGIS and see the change in PostgreSQL

10. To edit a layer, right-click on the layer in the *Layers* list and select *toggle editing*. Additional menu buttons will appear and your map display will change.
 - a. To view the attributes of the data, right-click on the layer and select *OPEN ATTRIBUTE TABLE*. Note that if you do this while editing is switched on, you will be able to edit the attribute data.
11. Right-click on the *london_counties* layer and select *EDIT*. Open the attribute table (right click on the layer and select the *london_counties* attributes layer to change the name of one of the counties by adding XXXX to the text. (Remember to enable the editing options, then open the attribute table, and then save your changes by switching off the editing option.)
12. Now go to the pgAdmin tool and view the data in PostgreSQL. Has the change been saved?

Edit data in PostgreSQL and see the change in QGIS

13. Create a text file called practical3.txt for all your SQL queries this week.
14. Edit the data in PostgreSQL using an SQL update statement (remember that the SCHEMA is public this time). Do you see your changes in QGIS? Do you need to reload the data to see the changes?
15. **(Optional task)** Clicking on *PROJECT > PRINT COMPOSER* opens a layout window which you can use to create printed maps. This offers options including the addition of a scale bar to the map, adding a legend and exporting the resulting map to a PDF or image. **NB: creating a professional looking map is not required for your assignment – a screenshot of your data is sufficient.**

Exercise 2- Importing Spatial Data into PostgreSQL

Spatial data is often stored in shapefiles (which are the GIS equivalent of word documents or CAD files or excel spreadsheets). However, as we saw in week one, data in files is more difficult to manage than data stored in a central database, as shapefiles can be stored on local hard drives, easily deleted, easily copied (which means that you then miss any changes to the original) and are not designed to be used by multiple users at the same time.

In this exercise, we will see how to import shapefile data into the PostgreSQL database, where it can then be more easily backed up and shared with multiple users. Having the spatial data stored in the same database as non-spatial data also allows queries to be run that combine both types of datasets.

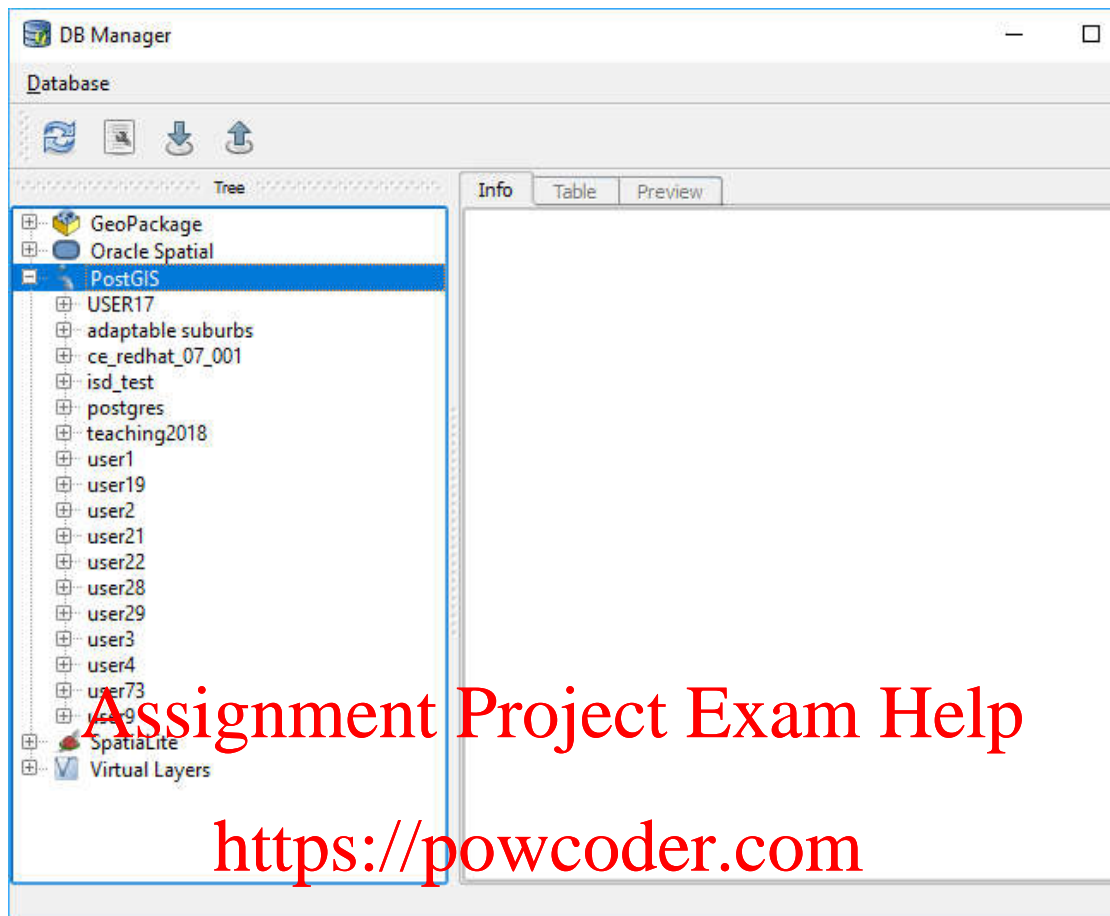
NB: You can import any spatial data into your database, and QGIS can work with all of the common spatial data formats.

Make sure you name the imported layers as follows:

- uk_counties_subset
- uk_highway_subset
- uk_poi_subset

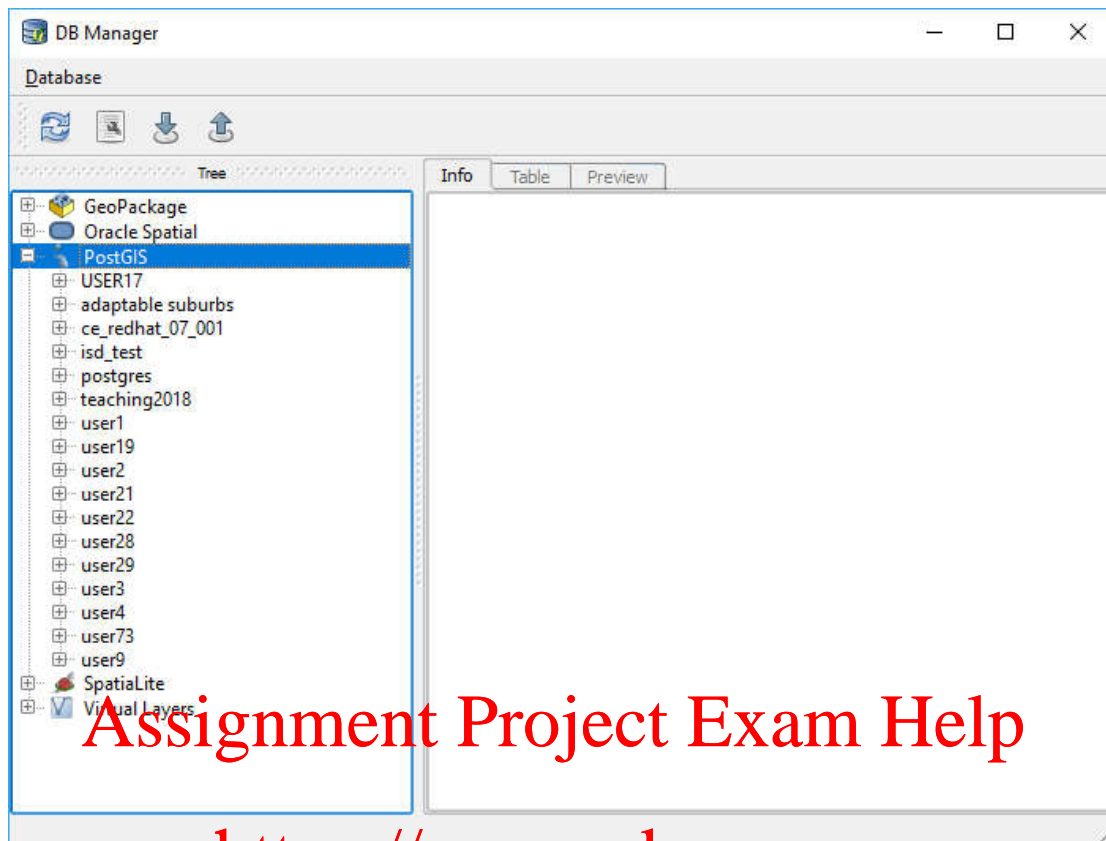
1. Make sure you are connected to PostGIS by opening a layer from your database.
2. Open the DATABASE > DB MANAGER > DB MANAGER tool.
3. Select POSTGIS and then select your connection from the list and double click – you should see the PUBLIC schema and any other schemas you have created.

Add WeChat powcoder

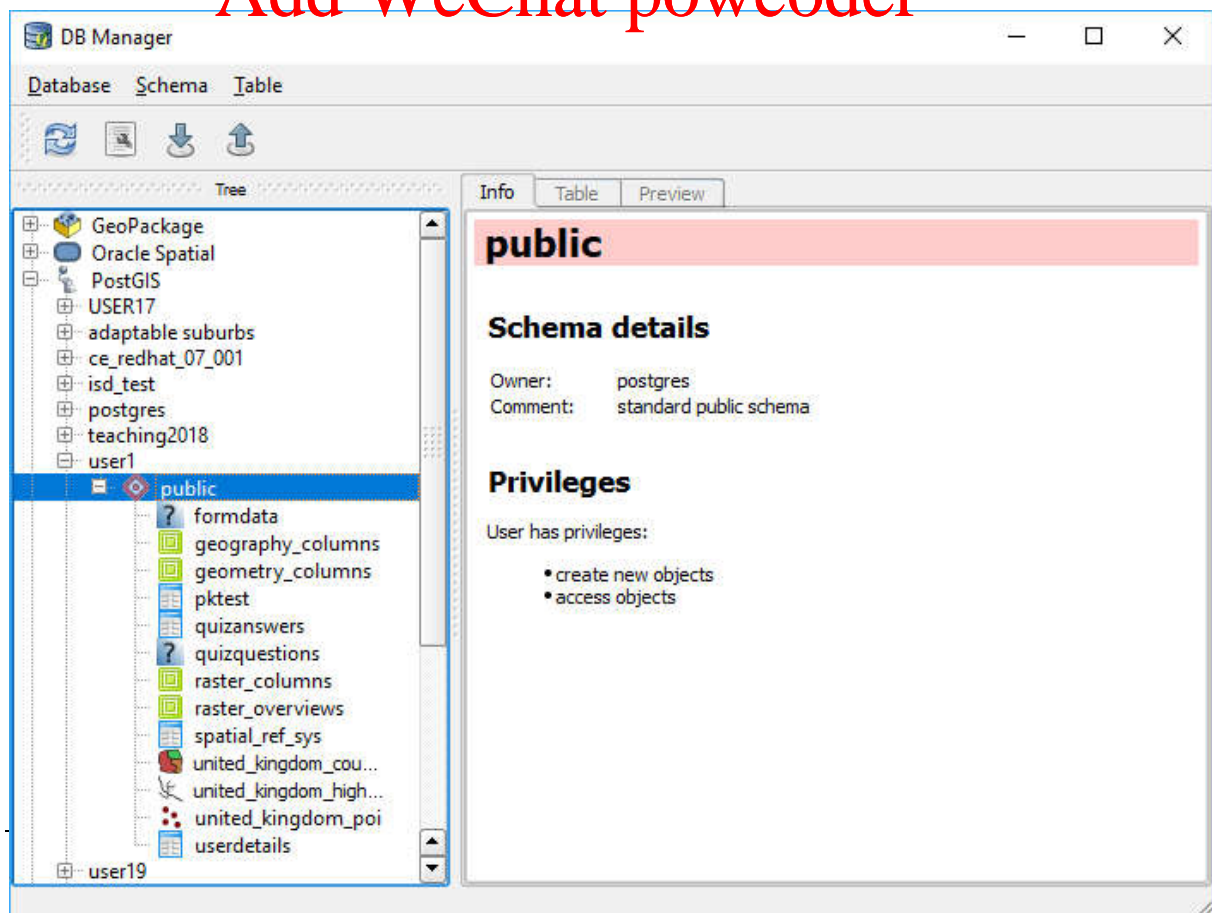


Add WeChat powcoder

4. Click on a schema and you will see your tables



5. To import data select the DOWN ARROW (Import Layer File) option



-
6. Select the shapefile you need, and set the other details as follows (NB – be careful about the SRID for the COUNTIES data as it is different):

Import vector layer

Input ...

☐ Import only selected features Update options

Output table

Schema

Table

Options

☐ Primary key

☐ Geometry column

☐ Source SRID ☐ Target SRID

☐ Encoding

☐ Replace destination table (if exists)

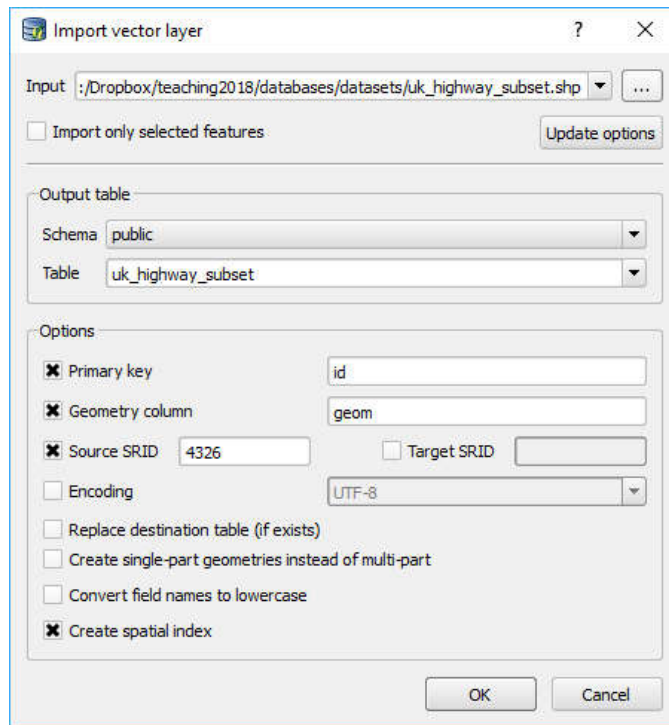
☐ Create shrink-wrap geometries instead of multi-part

☐ Convert field names to lowercase

☐ Create spatial index

OK Cancel

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder



Assignment Project Exam Help

7. Click OK and wait for the import to run – this may take a while!

<https://powcoder.com>

8. Repeat the above for the uk_poi_subset (SRID 4326) and the uk_counties_subset (SRID 27700)

Add WeChat powcoder

Part 2- Data Management: Spatial Data

Exercise 1 – Spatial Data DDL and DML

The following exercise will guide you through the creation of spatial tables from scratch in PostgreSQL/PostGIS and will show you how to insert simple data into the tables using the Well-Known-Text (WKT) method of representing data.

Note that for this exercise we are assuming the data is in British National Grid, using the Coordinate Reference System 27700.

1. Continue to use the practical3.txt file you created above for all the SQL in this part of the practical
2. In pgAdmin create a new schema called practical3
3. Type the following SQL into the SQL Editor Pane:

```
CREATE TABLE practical3.spatial_table_points(id serial, name  
varchar);
```

<https://powcoder.com>

Execute the query by pressing F5, or clicking on the Execute Query Icon.

Add WeChat powcoder

4. If you refresh the pgAdmin window database view, you will see that a new empty table was added to the database. We now need to spatially enable the table by adding a geometry column.
5. Empty the SQL Editor Pane, then write the following query into your text file and copy/paste into the editor:

```
SELECT  
AddGeometryColumn('practical3','spatial_table_points','the_geom'  
, '27700', 'POINT', 2);
```

6. Run the query, then refresh the table view to see the geometry column added to the table. In this case, we instructed the database to add to the table "*spatial_table*" a "*the_geom*" column which will hold spatial objects in "27700" SRID, aka British National Grid.

Note: The column will only hold "*Point*" objects, recorded in "2" dimensions (PostGIS allows to record objects in 3 dimensions if needed!). This is a CONSTRAINT

-
7. To add a point object to our table, we need to run the following query:

```
INSERT INTO practical3.spatial_table_points (name,the_geom) values
('Jeremy Bentham',st_GeomFromText('POINT(529440 182165)', 27700));
```

Here we instruct PostGIS to insert into our table a new value (or row), with a 'id' of 1, a 'name' value of 'Jeremy Bentham' , and add a point geometry to our 'the_geom' column. The GeometryFromText function returns a specified ST_Geometry value from a Well-Known Text representation (WKT). In this case, a Point with the specified British National Grid coordinates.

To check if the query executed, go to the table in the Admin Console, right-click and click View Data -> View Top 100 Rows. One row with the Jeremy Bentham entry should pop up.

Alternatively, you can run the following query in the Query Editor:

```
Select * from practical3.spatial_table_points;
```

8. Repeat the task above for the following points:

Assignment Project Exam Help		
id	Name	Coordinates
2	Euston Square	529448 - 182361
3	UCL Science Library	529639 - 182146

<https://powcoder.com>

9. To insert line data, you need to first create a table that can hold such data, as follows:

```
CREATE TABLE practical3.spatial_table_lines(id serial, name
varchar);
```

```
SELECT
AddGeometryColumn('practical3','spatial_table_lines','the_geom'
,'27700','LINESTRING',2);
```

10. You can then insert lines using text similar to the following:

```
INSERT INTO practical3.spatial_table_lines values
('1','University Street 1',st_geomFromText('LINESTRING(529370
182097, 529461 182160)', 27700));
```

(Note the use of the ST_GeomFromText option, and the term LINESTRING in this case.)

11. Similarly, for polygons, use:

```
CREATE TABLE practical3.spatial_table_polygons(id serial
PRIMARY KEY, name varchar);
```

```
SELECT
AddGeometryColumn('practical3','spatial_table_polygons','the_geom'
,'27700','POLYGON',2);
```

12. Add data to the polygon table as follows:

```
INSERT INTO practical3.spatial_table_polygons values ('1','DMS
Watson Library',st_GeomFromText ('POLYGON((529611 182164,
529635 182131, 529653 182145, 529629 182177, 529611 182164))',
27700));
```

13. Create a new table in the practical3 schema, called *polygons_no_crs* with the same columns as the previous polygon table.

14. Add a geometry column to this table, but this time put the SRID as 0 (i.e. we don't have a coordinate reference system).

15. Write the INSERT statements to insert the three polygons we listed the coordinates on the slides in class.

16. Once you have completed this exercise, open the tables you have created in QGIS to view the data you have inserted.

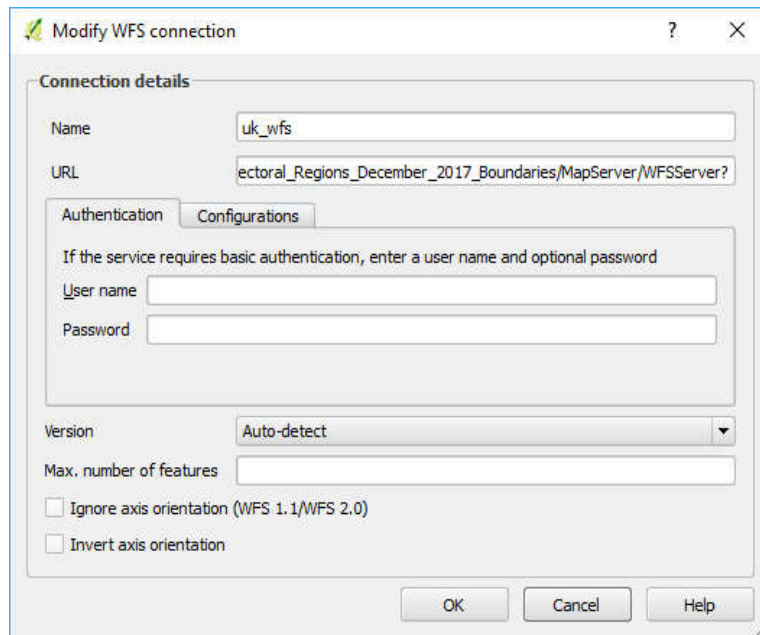
NB: When you open the polygons_no_crs data you will be asked to assign a coordinate reference system. The easiest thing to do here is to assign British National Grid – 27700 – and then your data will appear off the south west of the UK!

You can test this out by adding a map of the UK:

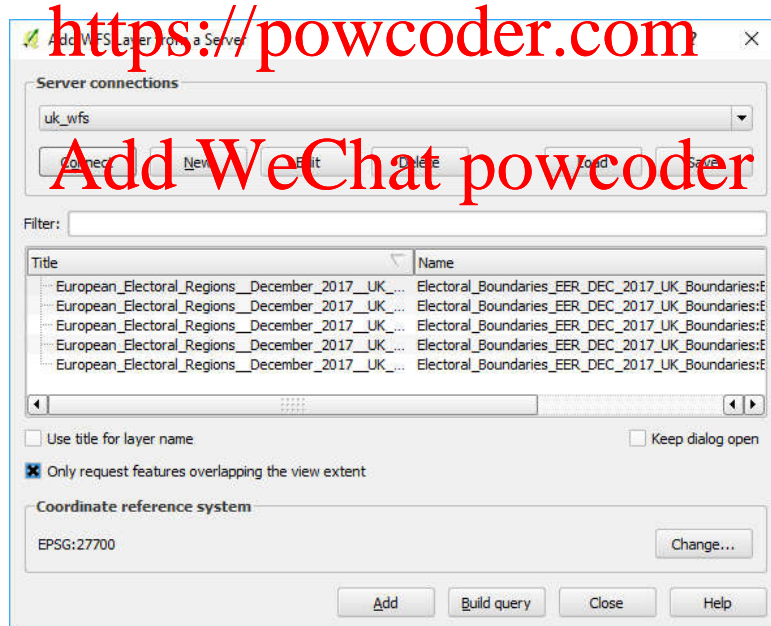
1. Click LAYER > ADD LAYER > ADD WFS¹ LAYER
2. Click NEW
3. Call the connection UK_WFS and use the URL:
https://ons-inspire.esriuk.com/arcgis/services/Electoral_Boundaries/European_Electoral_Regions_December_2017_Boundaries/MapServer/WFSServer?

¹ The more correct thing to do would be to define your own coordinate system – but this is not something we will cover on this module and you are not expected to do this for your assignment.

² A WFS is a Web Feature Service – this is a standard way of sharing map data over the internet. This is one service of 1000s available worldwide.



4. Click OK
5. Click CONNECT to connect to your new connection
6. Select the first layer from the resulting list and click OK



7. Now zoom to the *polygons_no_crs* layer by right-clicking on the layer and selecting ZOOM TO LAYER
8. Very slowly, zoom out (use the mouse wheel to make this easier) – your polygons will disappear as they get smaller, but the islands off the south west coast of the UK will appear
9. If you want to see the polygons, edit the style to change the *outline width* to around 5.0