Lab Setup
Spatial Databases

This note describes how to install the following software used in the spatial databases module:

- PostgreSQL/PostGIS (go to postgis, installation notes will get postgresql and then you enable postgis after installation)
- gpAdmin 4 (graphical user interface for connecting to postgis)
- OpenGIS

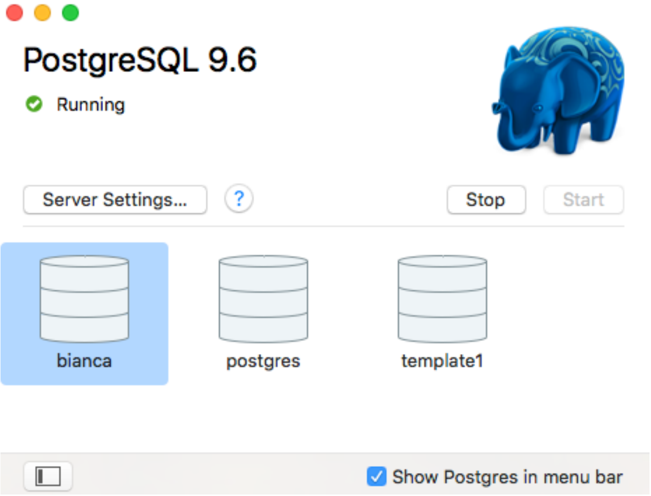
Additionally, this file shows you how to load data into PostGIS and how to run a spatial query in QGis.

Installation Notes

PostgreSQL and PostGIS

- Download from http://postgis.net/install/
- Follow download and install instructions from the website
    - PostGIS needs to be enabled, just downloading is not enough
- Download pgadmin 4 https://www.pgadmin.org/download/
    - Follow download and install instructions

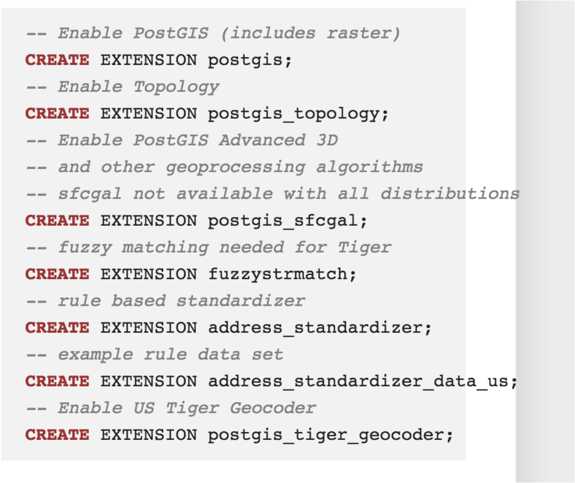
After installation:



Test that you can access the postgreSQL database. Quick way is using the terminal window. Remember that for CLI tools the executable needs to be added to your PATH variable (see documentation in links above). Create a table add some values, drop the table. This is to ensure that you have access and everything is working fine. If not, something went wrong during your installation.
Have a look around postgreSQL. Log on via psql. Run a `\dn` and `\dt`. What do you get?
`\list` gets you a list of all databases installed.

This has both postgres and postgis included. You need to enable postgis by running the following statements

Lab Setup
Spatial Databases



```
-- Enable PostGIS (includes raster)
CREATE EXTENSION postgis;
-- Enable Topology
CREATE EXTENSION postgis_topology;
-- Enable PostGIS Advanced 3D
-- and other geoprocessing algorithms
-- sfcgal not available with all distributions
CREATE EXTENSION postgis_sfcgal;
-- fuzzy matching needed for Tiger
CREATE EXTENSION fuzzystrmatch;
-- rule based standardizer
CREATE EXTENSION address_standardizer;
-- example rule data set
CREATE EXTENSION address_standardizer_data_us;
-- Enable US Tiger Geocoder
CREATE EXTENSION postgis_tiger_geocoder;
```

They can be found at http://postgis.net/install/
(They won't all run, if you get an error like:
create extension postgis_sfcgal;
ERROR: could not open extension control file "/Applications/Postgres.app/Contents/Versions/9.6/share/postgresql/extension/postgis_sfcgal.control": No such file or directory
Don’t worry about it for the moment.

- In your terminal, log out and on again.
- At your psql prompt type `\dn` to get a list of schemas, there should now be a tiger, tiger_data and topology schema, in addition to previous ones.
- At your psql prompt type `\dt` to get a list of tables, there should be a lot more associated with the `tiger`_schema.

Now try to set up a table with a geometry type (example from http://postgis.net/install/):
-- Create table with spatial column

```
CREATE TABLE mytable (id SERIAL PRIMARY KEY, geom GEOMETRY(Point,
26910), name VARCHAR(128));
```

-- Add a spatial index

```
CREATE INDEX mytable_gix ON mytable USING GIST(geom);
```

-- Add a point
-- 26910 is the so called “projection”, or the reference system. Can you find out which one it is? What do the numbers mean?

```
INSERT INTO mytable (geom) VALUES (ST_GeomFromText('POINT(0 0)', 26910)
);
```

-- Query for nearby points , also try a normal select * from…. And see how it will come up

```
SELECT id, name FROM mytable WHERE ST_DWithin(geom,
ST_GeomFromText('POINT(0 0)', 26910), 1000 );
```

<u>Now upload a database table from a file: *(alternatively, you can go through pgAdmin 4, see below)*</u>

- File provided in Webcourses: county.sql

```
psql -d your_db_name -f Your_path_to_the_file/county.sql
...
bianca=# \dt
                  List of relations
 Schema |            Name            | Type  | Owner
```

```
--------+-------------------------+-------+--------
 public | county                  | table | bianca
 public | helloworld              | table | bianca
 public | spatial_ref_sys         | table | bianca
 public | testspatial             | table | bianca


bianca=# \dt county
        List of relations
 Schema |  Name  | Type  | Owner
--------+--------+-------+--------
 public | county | table | bianca
(1 row)
```

Investigate the database table:

```
bianca=# select name from county where ST_Contains(geom,
ST_GeomFromText('POINT(-97.0 35.0)'));
     name
--------------
 POTTAWATOMIE
(1 row)

bianca=# select distinct name from county;
     name
--------------
 MAJOR
 PAWNEE
 HUGHES
 PITTSBURG
 MCCURTAIN
 CREEK
 GARFIELD
 LE FLORE
 TULSA
 WOODS
 SEQUOYAH
 ROGERS
 BRYAN
 HASKELL
 BEAVER
 TEXAS
 CHEROKEE
 GRADY
 CLEVELAND
 KAY
 OKMULGEE
bianca=# \d county
                                   Table "public.county"
   Column   |          Type          |                      Modifiers
------------+------------------------+----------------------------------
--------------------
 gid        | integer                | not null default
nextval('county_gid_seq'::regclass)
 name       | character varying(20)  |
  state      | double precision        |
  county     | double precision        |
  stateplane | character varying(2)    |
  geom       | geometry(MultiPolygon)  |
Indexes:
    "county_pkey" PRIMARY KEY, btree (gid)

bianca=#
```

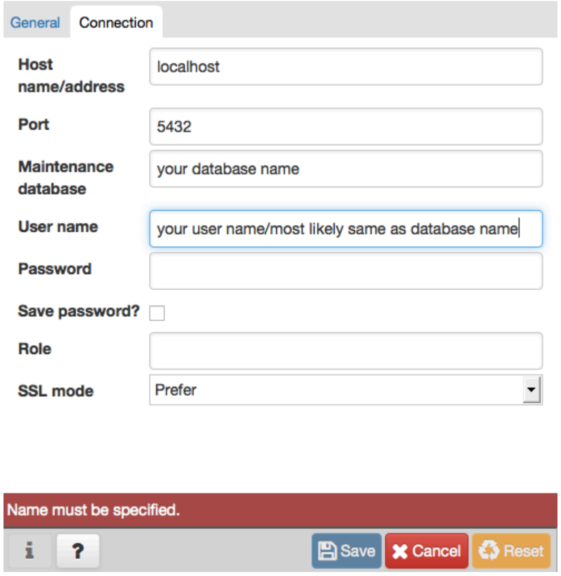
Note **geom** is the column where the geometry is stored. Geometry is generally only meaningful when rendered and viewed graphically. It is not a good idea to query raw geometry in a text interface. However we can run some queries in a text window:

```
bianca=# select st_perimeter(geom) from county where name = 'WOODS';
   st_perimeter
------------------
 2.83399677915026
(1 row)

bianca=# select sum(st_area(geom))/1000000 as "sq kms" from county where
name='WOODS';
        sq kms
----------------------
 3.37232713540013e-07
(1 row)
```

pgAdmin 4

- Download pgAdmin 4, link given in the lab folder
- Connect to the database
- Click add new server:



Host name/address: localhost

Port: 5432

Maintenance database: your database name

User name: your user name/most likely same as database name

Password:

Save password? □

Role:

SSL mode: Prefer

Name must be specified.

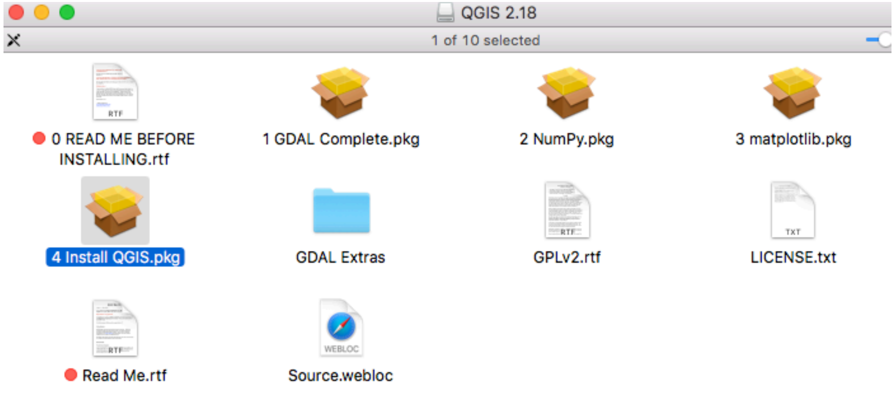
You shouldn't have to include a password. This depends on your postgreSQL installation process. Now read the documentation, see Webcourses folder.

Lab Setup
Spatial Databases

It is not strictly necessary to use pgAdmin 4. If you are comfortable with the CLI psql, use that.

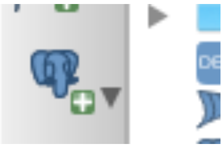
Install QGIS

- Installation link to QGIS is available in the Webcourses folder.
- Download the installer and follow the instructions.
- You may have several dependencies that need installing first (see the README that comes with the installer:

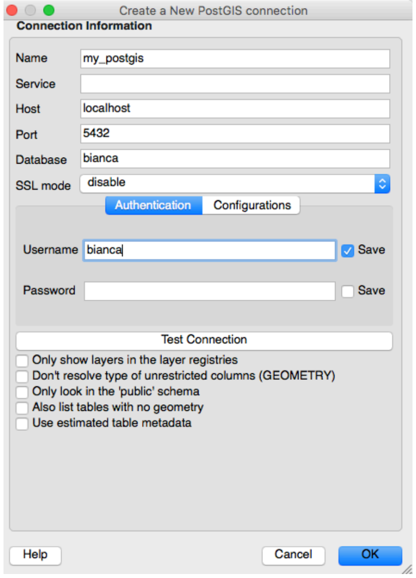


Connect PostGIS to QGIS

There are several ways of doing this. The easiest is to start QGIS, create an empty, new project, and on the left panel, use the button: Add PostGIS layers….



This lets you connect to your database.



Create a New PostGIS connection

Connection Information

Name: my_postgis
Service:
Host: localhost
Port: 5432
Database: bianca
SSL mode: disable

Authentication | Configurations

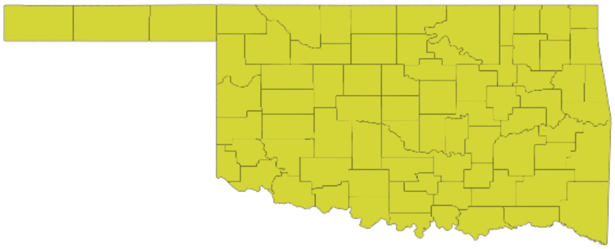
Username: bianca ☑ Save
Password: ☐ Save

Test Connection

☐ 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

Add the county table. Choose all defaults.

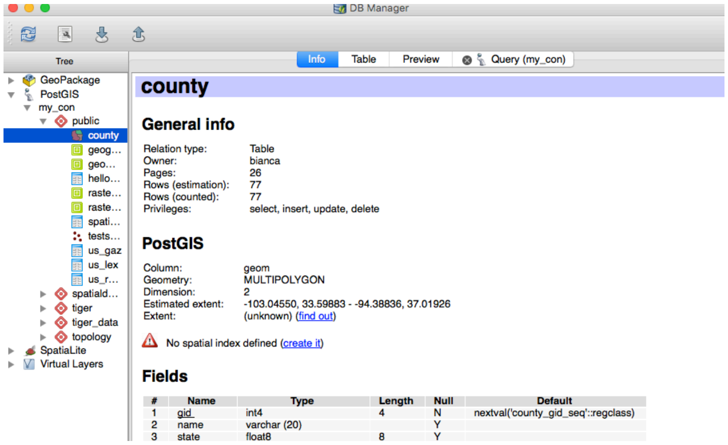
You should see something like the following:



You can manipulate the colours by double clicking on the left pane and in the style tab choosing your colour.

Now let's run a query.

Go to *Databases – DB Manager – DB Manager*.



county

General info

Relation type: Table
Owner: bianca
Pages: 26
Rows (estimation): 77
Rows (counted): 77
Privileges: select, insert, update, delete

PostGIS

Column: geom
Geometry: MULTIPOLYGON
Dimension: 2
Estimated extent: -103.04550, 33.59883 - -94.38836, 37.01926
Extent: (unknown) (find out)

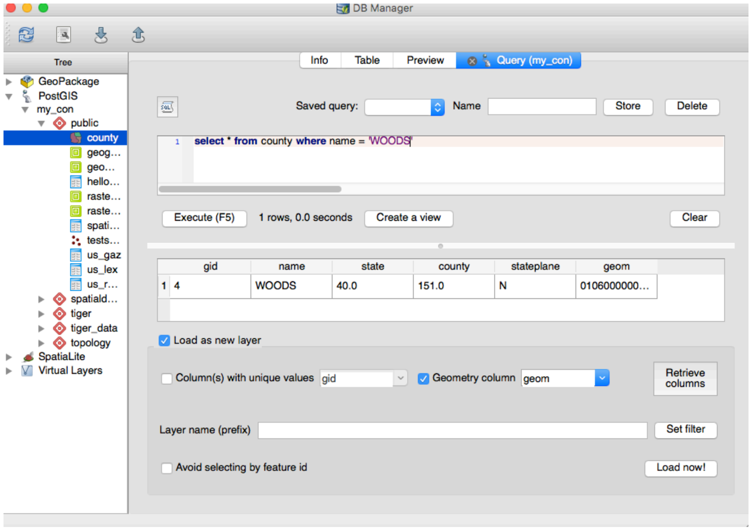
⚠ No spatial index defined (create it)

Fields

#	Name	Type	Length	Null	Default
1	gid	int4	4	N	nextval('county_gid_seq'::regclass)
2	name	varchar (20)		Y	
3	state	float8	8	Y	

Your database should come up here, as you have added it earlier already. Go to the query tab.

Insert a query as seen here:



```
select * from county where name = 'WOODS'
```

1 rows, 0.0 seconds

	gid	name	state	county	stateplane	geom
1	4	WOODS	40.0	151.0	N	0106000000…

☑ Load as new layer

□ Column(s) with unique values: gid

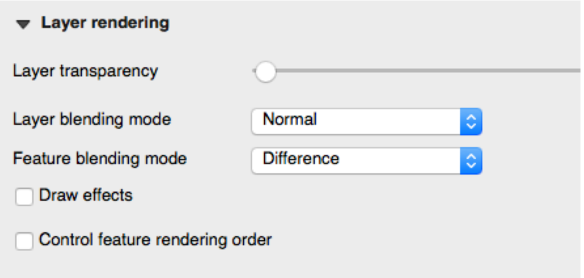
☑ Geometry column: geom

Layer name (prefix):

□ Avoid selecting by feature id

- Click Execute (F5).
- Check Loas as new layer
- Click Load Now!

Your query will come up as a new layer on the bottom left of QGIS.
To get both, your original map from the counties, and the query result showing as in the figure below, you need to change the layer settings. Several options are possible. This is just an example:



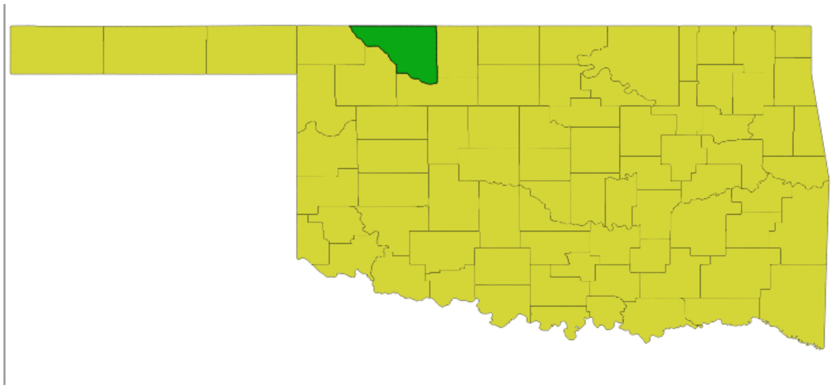
Layer transparency

Layer blending mode: Normal

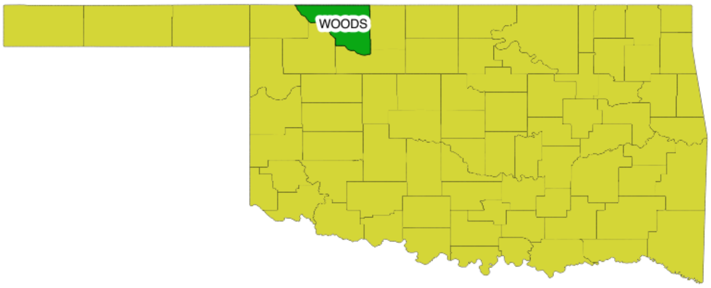
Feature blending mode: Difference

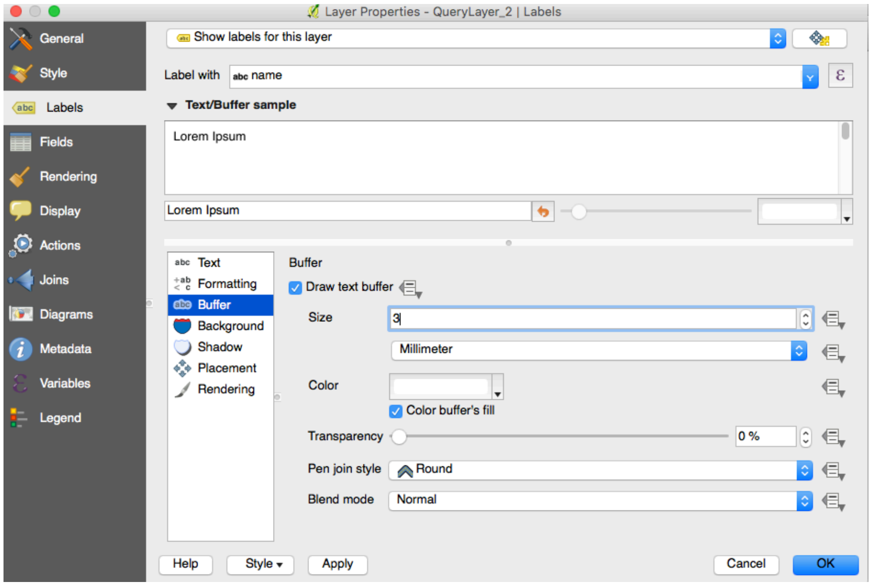
□ Draw effects

□ Control feature rendering order



You can show a label for your result set (or any of the counties). Experiment with these settings.





Show labels for this layer

Label with: name

Text/Buffer sample

Lorem Ipsum

Lorem Ipsum

Buffer

☑ Draw text buffer

Size: 3

Millimeter

Color:

☑ Color buffer's fill

Transparency: 0 %

Pen join style: Round

Blend mode: Normal