

# Adding Marina and Boat Ramp data to QGIS and Postgres

By Kody Miller

For CS540 – Professor Lehr

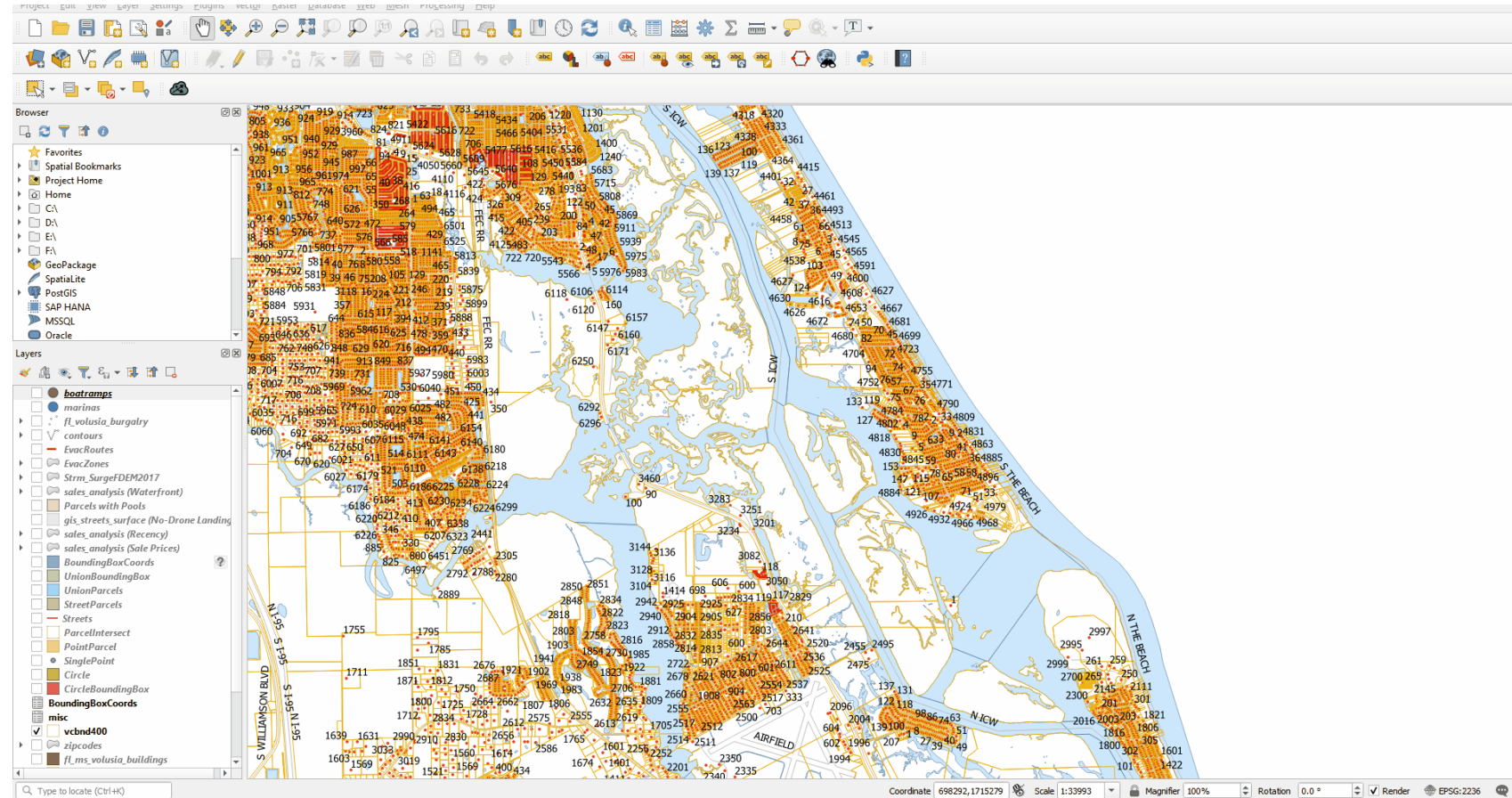
Spring 2021

# First, download the required files:

- <https://github.com/Kodyth/CS540>
  - Read the readme to learn more about each of the files in the repository.
  - The files needed to get the data in Postgres and QGIS are:
    - boatramps.zip
    - marinas.zip
    - Marina\_and\_boatramps\_distance.csv

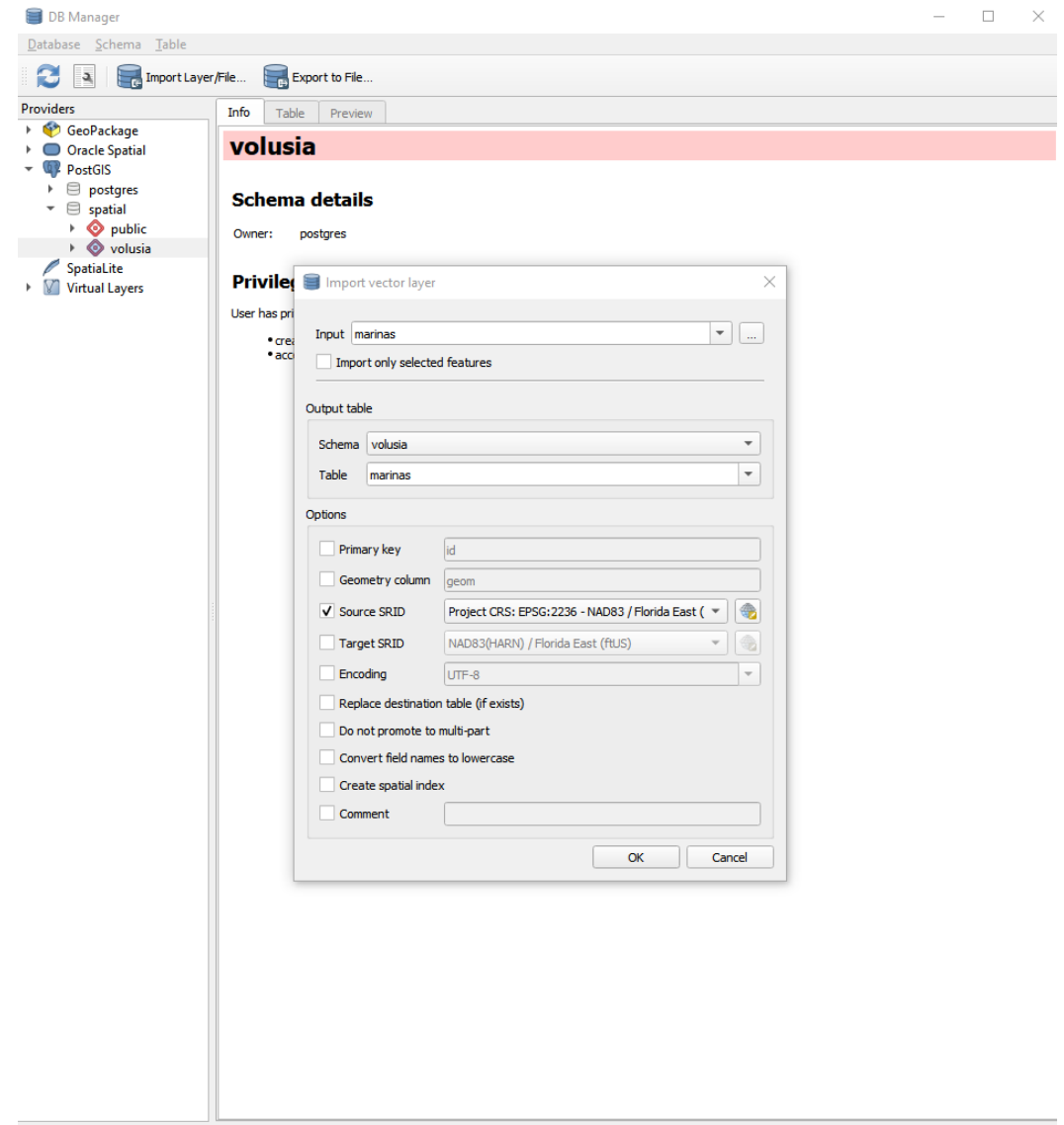
# Create new layers in QGIS to visualize the location of each Marina/Boat ramp

- Extract the .zip files into a known location such as C:\temp\cs540\qgis
- In QGIS, add two new layers for the marinas and boatramps
  - Select layer -> add layer -> new vector layer -> from file
    - Select marinas.shp to create the marinas layer
    - Select BoatRamps.shp to create the boatramp layer
      - Edit the name of the boatramps layer to remove capitals letters: (BoatRamps -> boatramps)



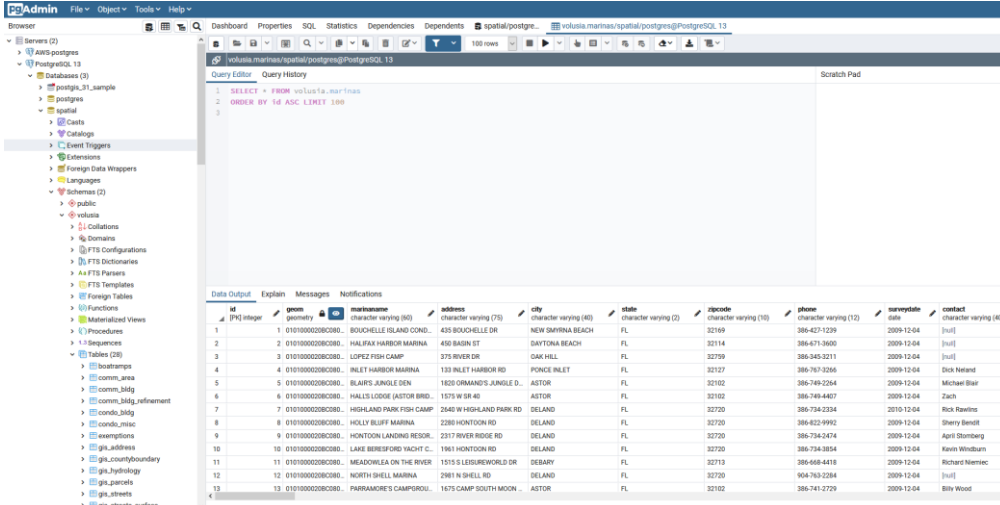
# Import the marinas and boatramps layers into Postgres

- Select DB Manager -> Postgis -> spatial -> volusia
- Next select import layer/file
  - For input, select the marinas layer (repeat this process to add the boatramps layer second)
  - For schema select volusia
  - For table, select marinas
  - Make sure the Source SRID is the correct one for the project (EPSG:2236)
  - Click done and QGIS will import the layer into Postgres as a new table



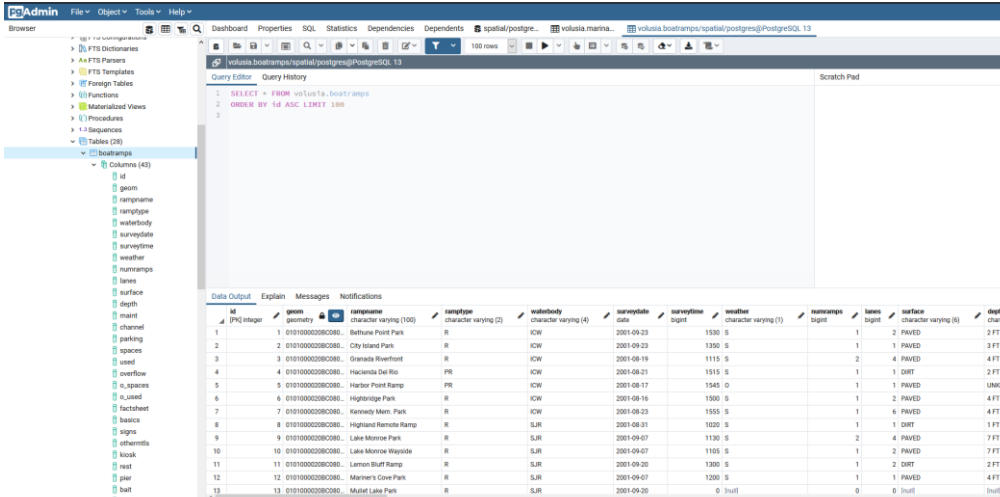
# Verify tables have been added successfully

- Check in PGAdmin that the tables have been created and contain data
  - Find each table by clicking databases -> spatial -> schema -> volusia -> tables -> marinas and boatramps
- It is recommended that you change the column labeled “name” to “marinaname” in the marinas table to prevent issues when querying the table
  - To do this, simply right click on the name column, select properties, change the name, and click save



PGAdmin interface showing the 'marinas' table in the 'volusia' schema. The table contains 13 rows of data. The columns are: id, geom, marinaname, address, city, state, zipcode, phone, surveydate, and contact.

id	geom	marinaname	address	city	state	zipcode	phone	surveydate	contact
1	01010002020C00...	BOUCHELLE ISLAND COND.	435 BOUCHELLE DR	NEW SMYRNA BEACH	FL	32149	386-427-1239	2009-12-04	[null]
2	01010002020C00...	HALPAK HARBOR MARINA	400 BAKER ST	DAYTONA BEACH	FL	32114	386-471-3600	2009-12-04	[null]
3	01010002020C00...	LOPEZ FISH CAMP	375 RIVER DR	DADE HILL	FL	32759	386-345-3211	2009-12-04	[null]
4	01010002020C00...	INLET HARBOR MARINA	132 INLET HARBOR RD	PONCE INLET	FL	32127	386-767-3266	2009-12-04	Dick Netland
5	01010002020C00...	BLAIRS JUNGLE DEN	1820 ORMANDS JUNGLE D.	ASTOR	FL	32102	386-749-2264	2009-12-04	Michael Blair
6	01010002020C00...	HALLS LODGE (ASTOR BRG.)	1575 W DR DR	ASTOR	FL	32102	386-749-4467	2009-12-04	Zach
7	01010002020C00...	HIGHLAND PARK FISH CAMP	2640 W HIGHLAND PARK RD	DELAND	FL	32720	386-734-2334	2010-12-04	Rick Rawlins
8	01010002020C00...	HOLLY BLUFF MARINA	2280 HONTON RD	DELAND	FL	32720	386-823-9992	2009-12-04	Berry Smith
9	01010002020C00...	HONTON LANDING RESOR.	2317 RIVER RIDGE RD	DELAND	FL	32720	386-734-2474	2009-12-04	April Stenberg
10	01010002020C00...	LAKE BEEFORD YACHT C.	1961 HONTON RD	DELAND	FL	32720	386-734-3854	2009-12-04	Kevin Windburn
11	01010002020C00...	MEADOWLA ON THE RIVER	1315 S LESUREWORLD DR	DEBARY	FL	32713	386-668-4418	2009-12-04	Richard Nemec
12	01010002020C00...	NORTH SHELL MARINA	2961 N SHELL RD	DELAND	FL	32720	904-763-2284	2009-12-04	[null]
13	01010002020C00...	PARADISE CAMPROU.	1675 CAMP SOUTH MOON	ASTOR	FL	32102	386-741-2729	2009-12-04	Billy Wood

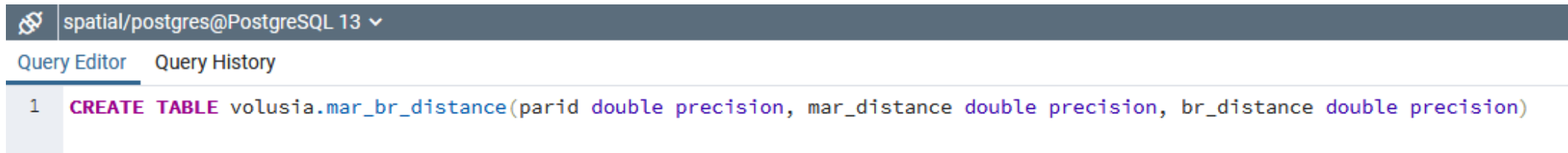


PGAdmin interface showing the 'boatramps' table in the 'volusia' schema. The table contains 13 rows of data. The columns are: id, geom, rampname, ramptype, waterbody, surveydate, weather, rampsize, lanes, surface, and depth.

id	geom	rampname	ramptype	waterbody	surveydate	weather	rampsize	lanes	surface	depth
1	01010002020C00...	Bethune Point Park	R	ICW	2001-09-23	1530 S	1	2	PAVED	2 FT
2	01010002020C00...	City Island Park	R	ICW	2001-09-23	1350 S	1	1	PAVED	3 FT
3	01010002020C00...	Granada Riverfront	R	ICW	2001-08-19	1115 S	2	4	PAVED	4 FT
4	01010002020C00...	Palmdale De Rio	PR	ICW	2001-08-21	1815 S	1	1	DIRT	2 FT 4
5	01010002020C00...	Harbor Point Ramp	R	ICW	2001-08-17	1345 S	1	1	PAVED	UNKN
6	01010002020C00...	Hightbridge Park	R	ICW	2001-08-16	1300 S	1	2	PAVED	4 FT
7	01010002020C00...	Kennedy Mem. Park	R	ICW	2001-08-23	1555 S	1	6	PAVED	4 FT 4
8	01010002020C00...	Highland Remote Ramp	R	SLR	2001-08-31	1620 S	1	1	DIRT	1 FT
9	01010002020C00...	Lake Monroe Park	R	SLR	2001-09-07	1130 S	2	4	PAVED	7 FT
10	01010002020C00...	Lake Monroe Wayside	R	SLR	2001-09-07	1165 S	1	2	PAVED	7 FT 4
11	01010002020C00...	Lemon Bluff Ramp	R	SLR	2001-09-20	1380 S	1	2	DIRT	2 FT
12	01010002020C00...	Mariner's Cove Park	R	SLR	2001-09-07	1200 S	1	1	PAVED	4 FT 1
13	01010002020C00...	Mullet Lake Park	R	SLR	2001-09-20	0	0	0	PAVED	[null]

# Create a new table in PGAdmin to import the distance data

- Create a new table using the below command in PGAdmin using the query tool; this table will be used to import the data into the other tables

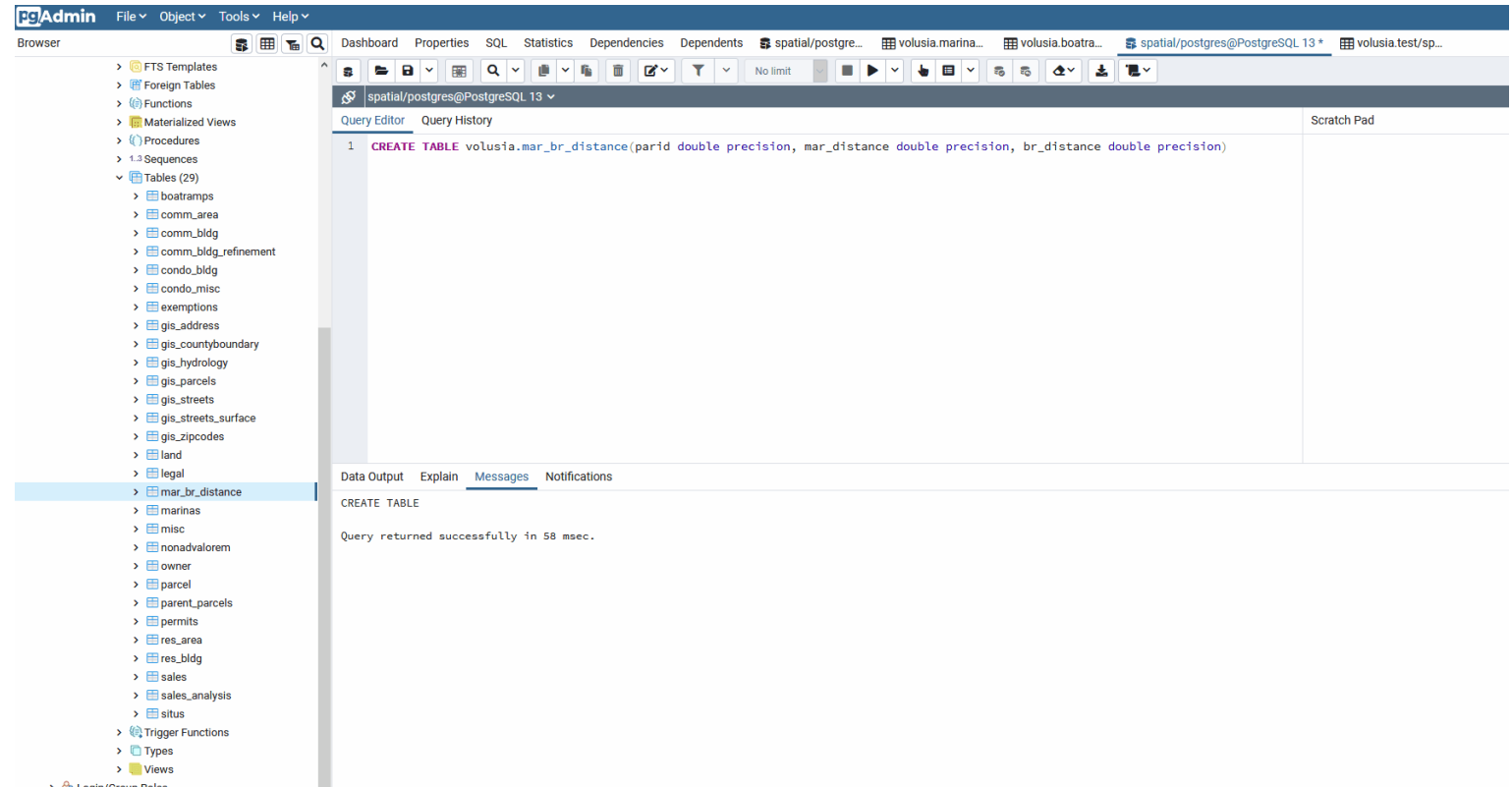
A screenshot of the PGAdmin interface. The top bar shows 'spatial/postgres@PostgreSQL 13' with a dropdown arrow. Below the bar are two tabs: 'Query Editor' (active) and 'Query History'. The main area displays a SQL command on a single line, numbered '1' on the left. The command is: `CREATE TABLE volusia.mar_br_distance(parid double precision, mar_distance double precision, br_distance double precision)`. The text is color-coded: 'CREATE' is purple, 'TABLE' is pink, and the rest is blue.

```
1 CREATE TABLE volusia.mar_br_distance(parid double precision, mar_distance double precision, br_distance double precision)
```

- The distance data was obtained by using a KNN query in Postgres that identified the closest marinas and boat ramps to a given parcel
  - A python script was used to iteratively loop through all the parcels in Volusia country – it is available to view in the repository a:  
**“update\_marina\_and\_boatramp\_distances.py”**

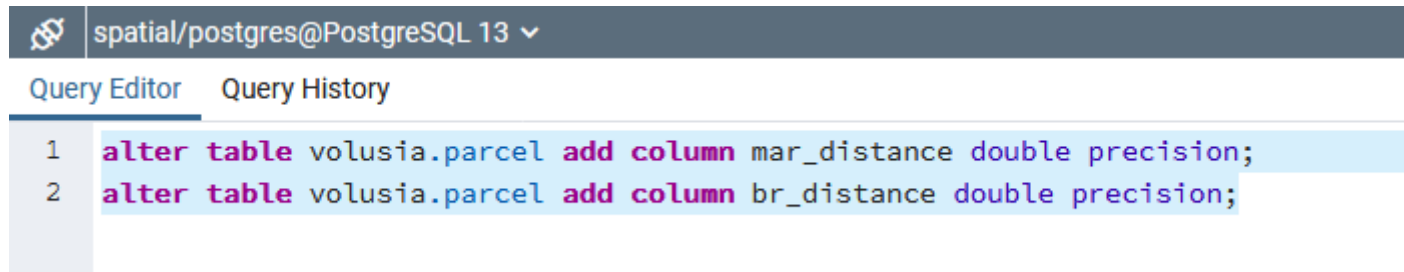
# Import the data into the newly created table

- Right click on the new table and select Import/Export
  - Make sure Import is selected
  - Choose the “marina\_and\_boatramp\_distance s.csv” file for the filename
  - The format is csv
  - Under Miscellaneous, select YES for header
  - Select OK and the data will be imported into the table
    - Verify this is the case by viewing at least the first 100 rows of the table



# Prepare the desired table for data insertion

- You can now begin the processing of merging the data into any other table such as the volusia.parcel table.
- First alter the desired table to add the new columns for the distance data:
  - mar\_distance
  - br\_distance



```
spatial/postgres@PostgreSQL 13 ▾  
Query Editor  Query History  
1 alter table volusia.parcel add column mar_distance double precision;  
2 alter table volusia.parcel add column br_distance double precision;
```



# Merge the data into the desired tables

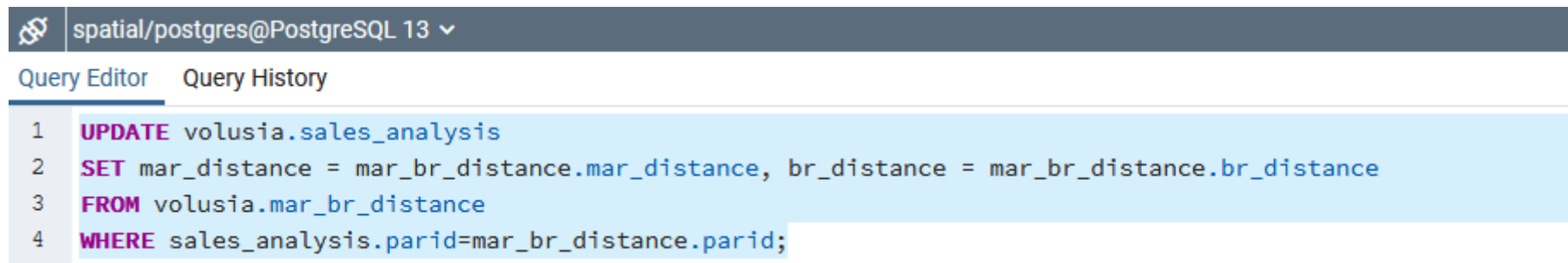
- You can now merge the data into the volusia.parcel table by using the below command:

```
6  
7 UPDATE volusia.parcel  
8 SET mar_distance = mar_br_distance.mar_distance, br_distance = mar_br_distance.br_distance  
9 FROM volusia.mar_br_distance  
10 WHERE parcel.parid=mar_br_distance.parid;
```

- After doing that, once again check that the table has been updated correctly

# Wrapping up

- At this point, the volusia.parcel table should now contain the distance data a parcel is to the closest Marina and Boat Ramp in miles.
- You can also add the data to the volusia.sales\_analysis table by adding the necessary columns (mar\_distance, br\_distance) and running the below query



The screenshot shows a PostgreSQL Query Editor window. The title bar indicates the connection is to 'spatial/postgres@PostgreSQL 13'. Below the title bar are two tabs: 'Query Editor' (which is active) and 'Query History'. The main area of the 'Query Editor' tab contains an SQL query with four lines, numbered 1 through 4 on the left. The query is an UPDATE statement that copies distance data from the 'mar\_br\_distance' table into the 'sales\_analysis' table.

```
1 UPDATE volusia.sales_analysis
2 SET mar_distance = mar_br_distance.mar_distance, br_distance = mar_br_distance.br_distance
3 FROM volusia.mar_br_distance
4 WHERE sales_analysis.parid=mar_br_distance.parid;
```

- To verify all the distances have been placed correctly, you can measure the distance a parcel is from the nearest marina or boat ramp in QGIS using the distance tool