Spatial Data Science Bootcamp

Set Up Your Environment

- Virtual machine environments (Linux-based)
- Spatial databases (PostgreSQL/PostGIS)

Wrangle Data

- Modern data standards and formats (GeoJSON)
- Open mapping tools (GDAL, QGIS)

Analyze Data

- Python (i.e. PySAL, NumPy, Pandas, Jupyter Notebook)
- R and R Studio (i.e. sp, raster, rgdal, maptools)

Visualize and Publish Data

- Web mapping and data visualizations (CARTO, Leaflet, D3)
- Data Visualization in Python

Day 1: Open Data Formats/ Tools and Spatial Databases

Morning Agenda:

1. Presentation: Open Data Standard Formats and Tools
2. Hands-on Tutorial #1: Intro to VM, GeoJSON, and GitHub
3. Hands-on Tutorial #2: Open Data Tools: GDAL/OGR

Afternoon Agenda:

1. Presentation: Intro to Spatial Databases
2. Hands-on Tutorial #3: Manage PostgreSQL/PostGIS databases using pgAdmin
3. Hands-on Tutorial #4: Write Spatial Queries and use PostgreSQL/PostGIS databases in QGIS

Intro to Spatial Databases: PostgreSQL/PostGIS

Jenny Palomino

May 29, 2017

Berkeley UNIVERSITY OF CALIFORNIA
College of Natural Resources
INTERNATIONAL AND EXECUTIVE PROGRAMS

GEOSPATIAL INNOVATION FACILITY
Cutting-Edge Mapping Technology at UC Berkeley

PostgreSQL and PostGIS

- PostgreSQL database developed in 1980s at UCB by Computer Science professor Michael Stonebraker

- PostGIS is the Spatial Database Engine for PostgreSQL: adds support for spatial data types and operations on those data

- PostGIS is installed as an extension to PostgreSQL and forms the spatial database backend to many online web mapping applications, like CARTO

Getting Started with PostgreSQL/PostGIS

Install PostgreSQL and PostGIS on Windows, Mac, or Linux

Optional Install: pgAdmin (Graphical User Interface)

Enable PostGIS in individual databases (see exercise slides):

1. Connect to the database with psql or pgAdmin
2. Run the following SQL statements (PostgreSQL documentation):
   - Create extension adminpack;
   - Create extension postgis;

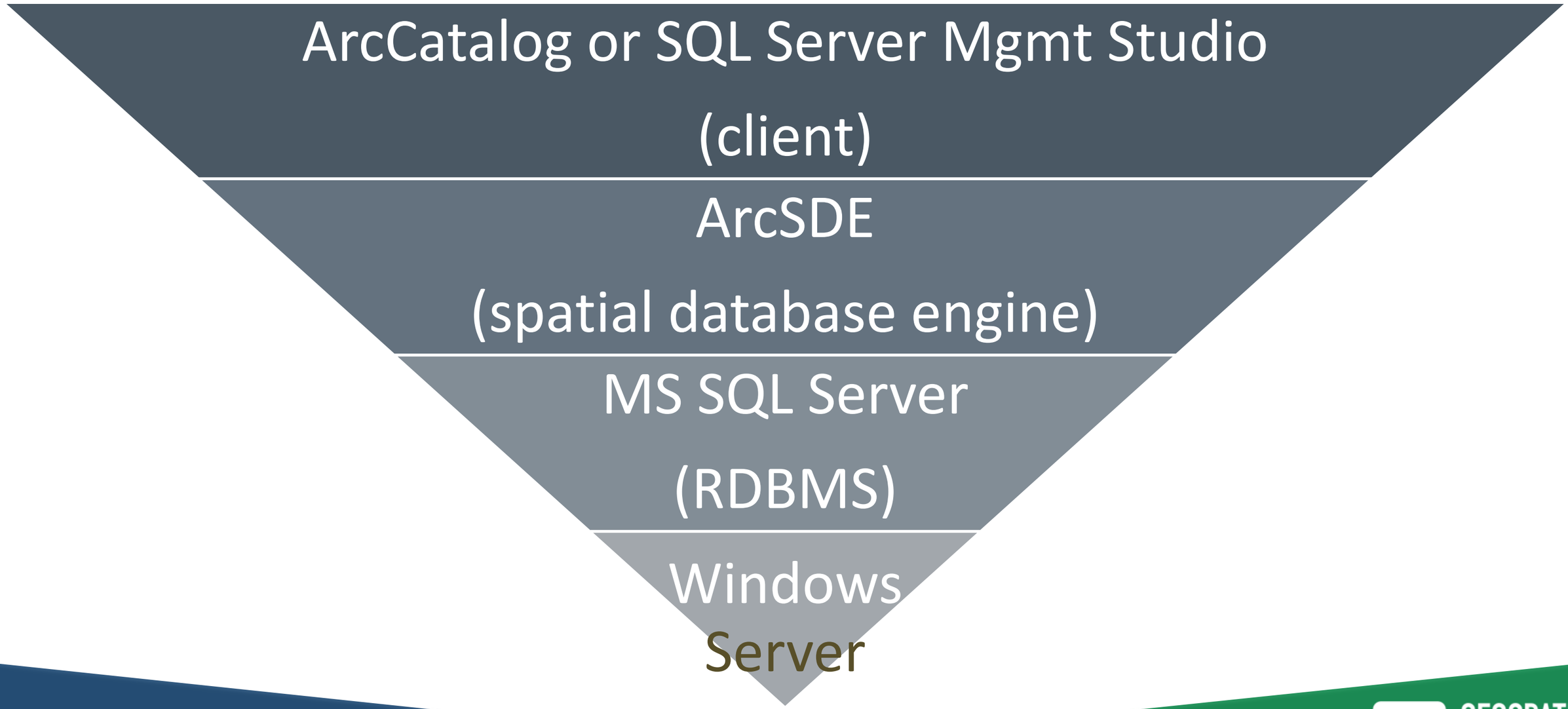
Accessing PostgreSQL/PostGIS

Once installed, PostgreSQL runs as a service on the operating system

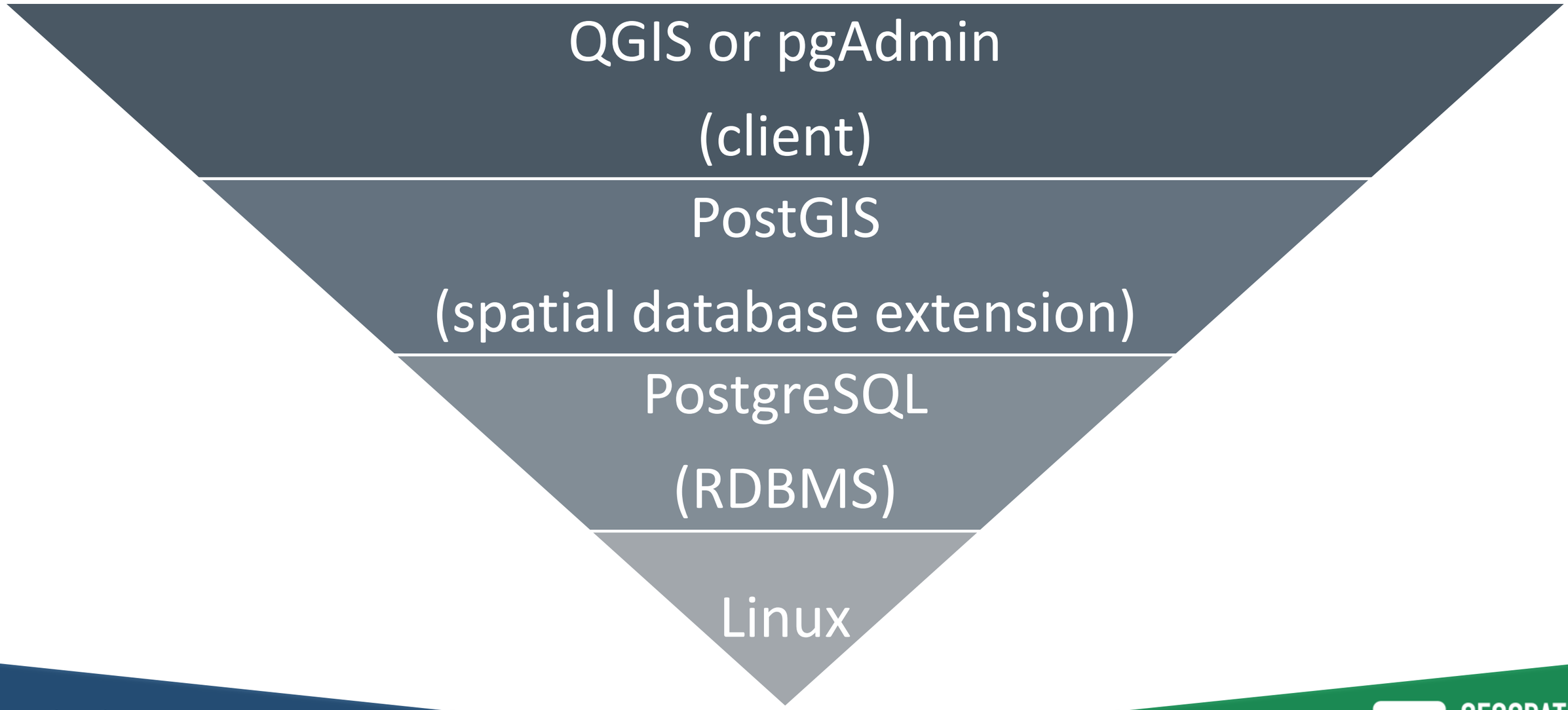
- Referred to as a postgres ‘instance’ (there can be multiple, as needed)
- Accessed via the terminal (command line) or GUI like pgAdmin
- PostGIS is the spatial database extension tool for PostgreSQL and thus, is installed on top on PostgreSQL

PostGIS is conceptually similar to ArcSDE, which is ESRI’s spatial database engine that runs on top of RDBMS (i.e., MS SQL Server)

Basic Architecture: ESRI and MS SQL Server



Basic Architecture: PostgreSQL/PostGIS



Useful Tools to Load PostgreSQL/PostGIS

shp2pgsql

raster2pgsql

geojson2pgsql

csv2psql (or COPY FROM statement: see provided SQL file called Part1_ManagingData_SelectbyAttributes.sql)

What is Structured Query Language (SQL)?

Standardized syntax used to create, read, update and delete (or CRUD) rows in database tables

Basic Structure:

```
select * from table_name
where column_name = 'value'
order by column_name;
```

Learn SQL for free at code academy and w3schools

Examples of SQL Queries

```
select * from table_name
where column_name = 'value'
order by column_name;
```

* indicates all columns in table will be returned

Consider a database called NPS with a table called parks.

The table called parks contains multiple columns of data such as:
unit_name, unit_type, and visitation_2017.

Examples of SQL Queries

```
select * from parks
where unit_name = 'Yosemite';

select unit_name, visitation2017 from parks
where unit_type = 'National Recreation Area';

select count(*) from parks;

select distinct (unit_type) from parks
order by unit_type;

select unit_type, count(*) from parks
group by unit_type;
```

Relational Database Management System (RDBMS)

Hypothetical Relational Database Model

PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Street, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sept-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1952	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution

http://www.ibm.com/developerworks/library/x-matters8/

Fundamentals of RDBMS

Primary key: unique ID for row within a table

Foreign key: ID connecting one row to rows in other tables

Constraints: can be applied to columns, tables, or databases

- Ex: all values for primary key column need to be unique

Joins: how rows in one table are matched to rows in other tables

- Types: one-to-one, one-to-many, many-to-many

Fundamentals of RDBMS - Joins

Hypothetical Relational Database Model

PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Street, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sept-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1952	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution

http://www.ibm.com/developerworks/library/x-matters8/

Rise of NoSQL Database Options

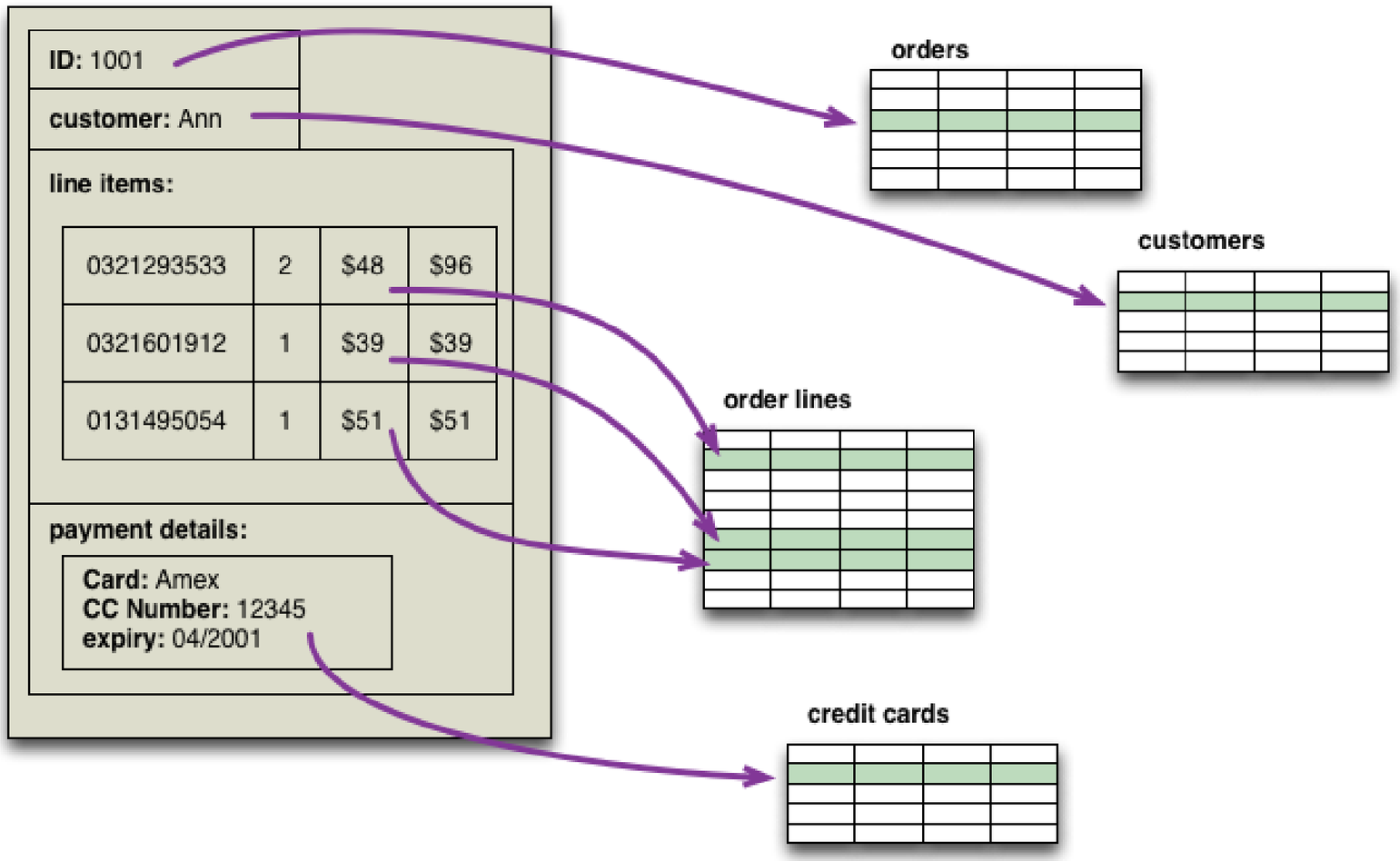
Instead of relational structure, data is stored as hierarchical documents (including a key and a value for each data item)

GeoJSON, XML, etc are used to structure and access documents via APIs

Design driven by “big” data, schema flexibility, the cloud, and distributed processing

http://www.mongodb.com/nosql-explained

NoSQL Document vs Relational Tables



http://martinfowler.com/bliki/AggregateOrientedDatabase.html

Other Popular Spatial Database Options

Proprietary: Oracle and Microsoft SQL Server
(Both with or without ESRI ArcGIS Server/ArcSDE)

Open Source: MongoDB (NoSQL), SpatiaLite, MySQL

Cloud: SAP HANA, Google Cloud Datastore and Google Cloud BigTable (Both NoSQL)

Hands-on Tutorial #3 – Manage PostgreSQL/PostGIS databases using pgAdmin

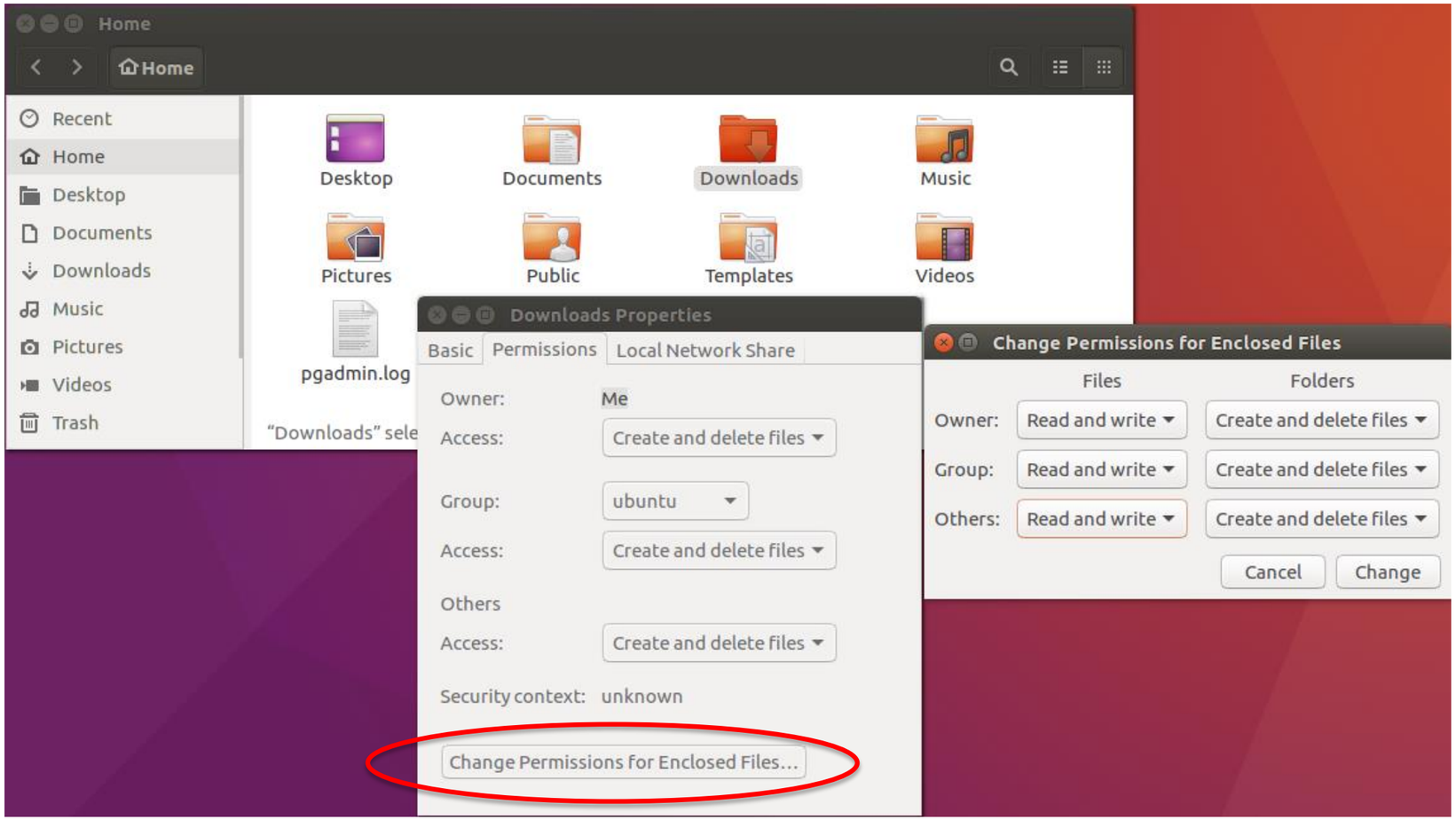
Tasks:

1. Create a new user for PostgreSQL via terminal
2. Connect to PostgreSQL as new user via pgAdmin
3. Create a new database
4. Enable the PostGIS extension in the new database
5. Learn fundamentals of querying data with SQL (including loading and exporting CSV files using COPY statement)

Necessary items: terminal, pgAdmin, SQL file

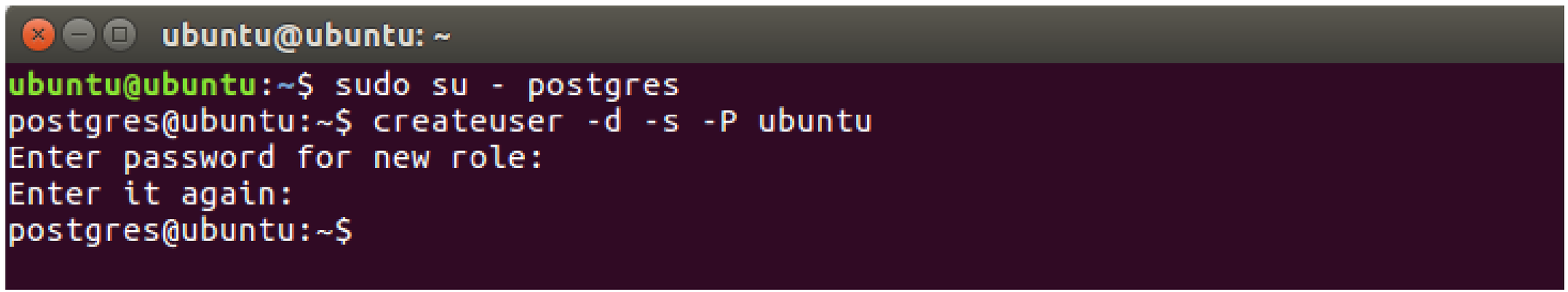
1. Part1_ManagingData_SelectbyAttributes.sql

Preparation for Exercise: Set Folder Permissions on Linux



Right-click on the Downloads folder and select the Permissions tab. Click Change Permissions for Enclosed Files, and change all permissions to Read and Write and Create and delete files.

Task #1: Create a new user for PostgreSQL using terminal



In the terminal, type these two commands, hitting enter after each one:

sudo su - postgres

createuser - d - s -P ubuntu

You will be prompted to assign a password to the new user called ubuntu. For this exercise, use something simple like bootcamp2017.

With these commands, you are switching to the operating system user called postgres and executing the createuser command to create a new PostgreSQL user called ubuntu. Parameter P prompts you to assign a password, -d allows the user to create new databases, and –s creates the user as a “super” user with virtually unlimited access within PostgreSQL.

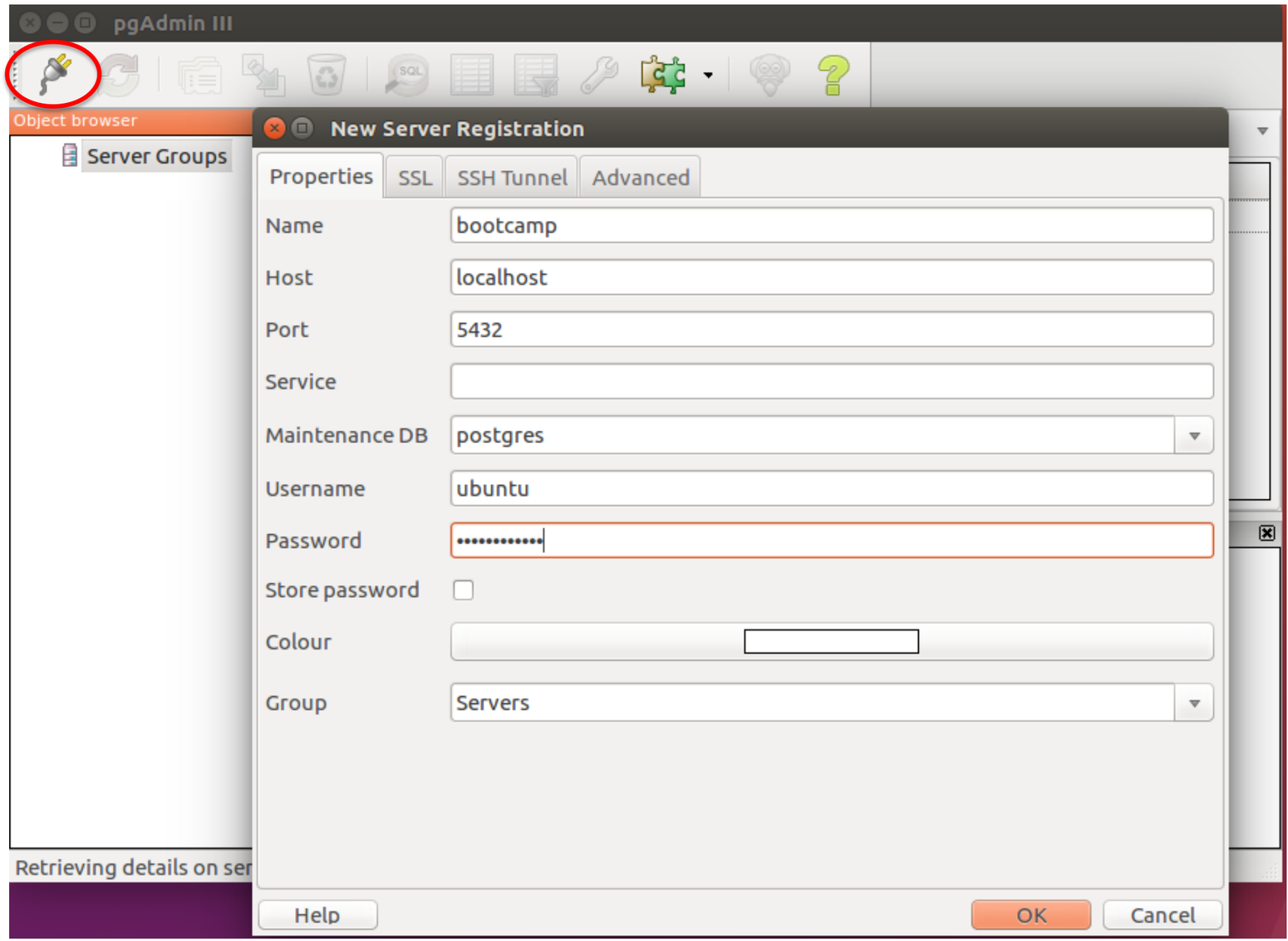
Task #2: Connect to PostgreSQL via pgAdmin

Click on the Search Computer icon and type pgAdmin. Double-click the pgAdmin icon to open.

Click the icon circled in red to add a connection to a server.

Fill out the window as shown, with ubuntu as the UserName and your assigned password.

Click OK.

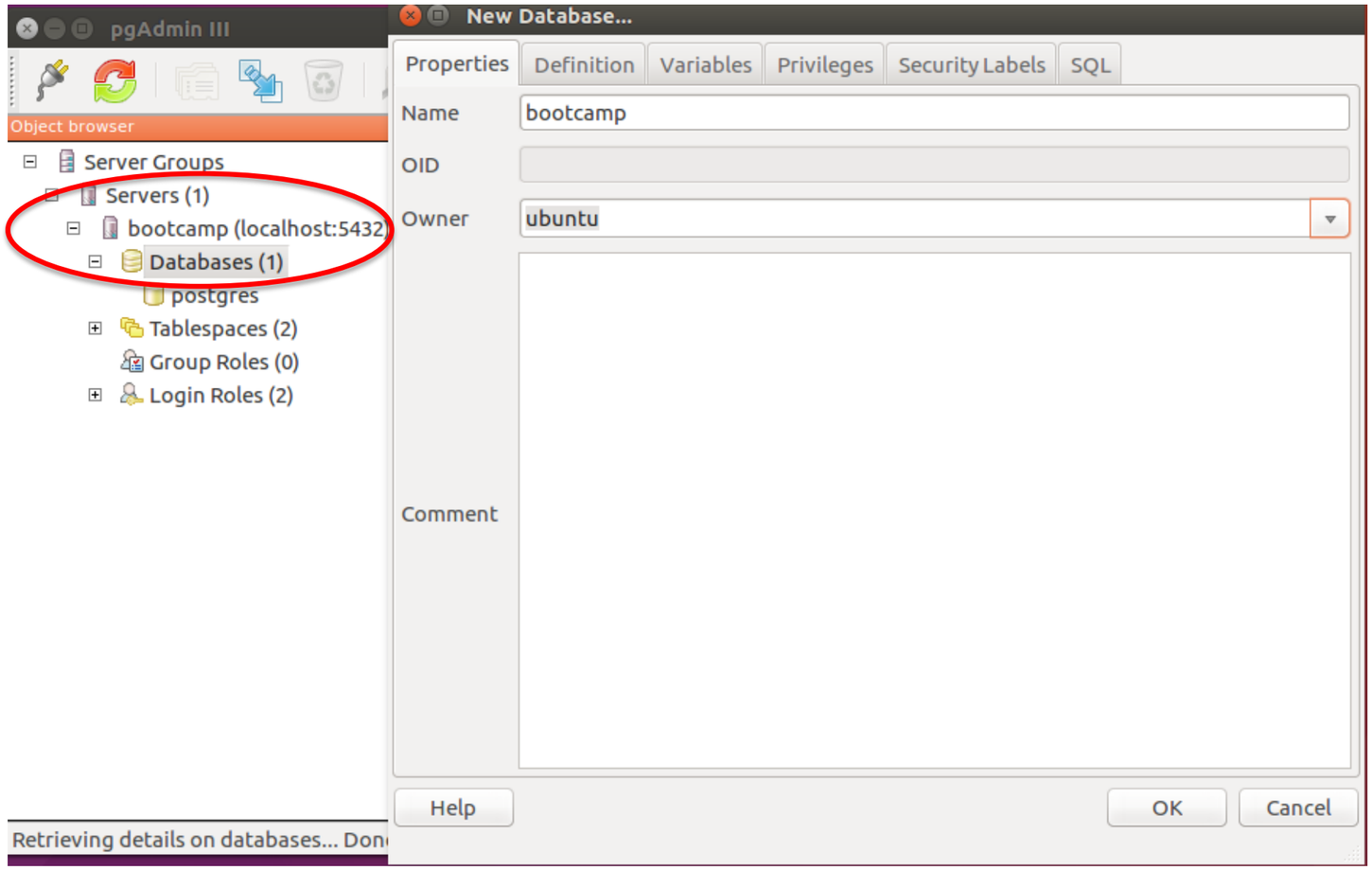


Task #3: Create a new database

Click on the + next to bootcamp server connection to expand the content list. You will see the default database called postgres.

Right on Databases and select New Database. Create a new database called bootcamp with ubuntu as the owner.

Click OK.

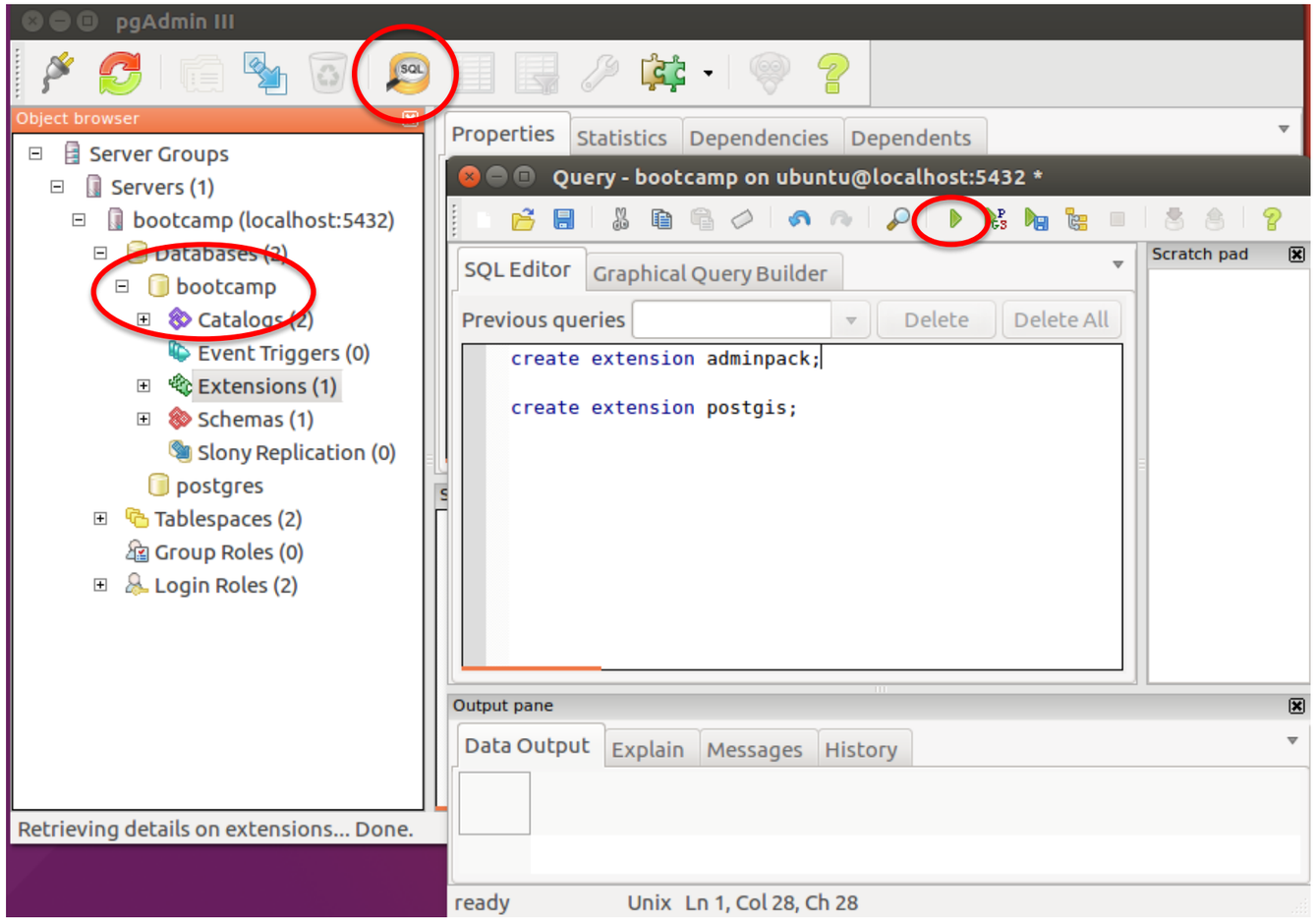


Task #4: Enable PostGIS extension in new database

Double-click on the bootcamp database to activate it. Click on the SQL Window icon, circled.

Type the two commands:
create extension adminpack;
create extension postgis;

Execute them by clicking on the Execute Query icon, circled.

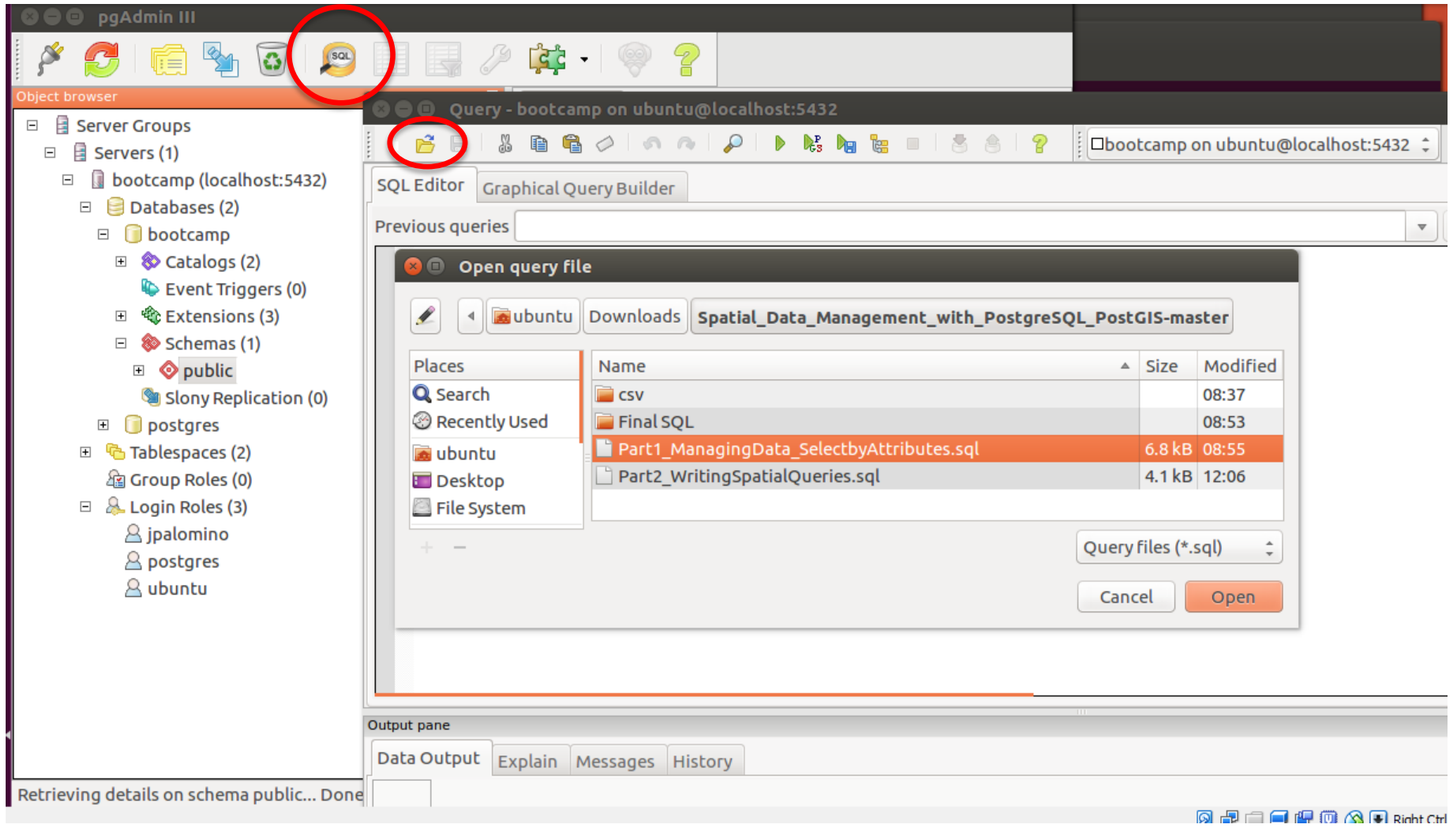


Task #5: Fundamentals of querying data with SQL

Click on the SQL Window icon again to open a new window.
Click on the Open file icon.

Navigate to the folder downloaded from github and select the sql file called Part1_ManagingData _SelectbyAttributes.sql.

Click open. Follow the exercise in the file.



Hands-on Tutorial #4 – Write Spatial Queries and use PostgreSQL/PostGIS databases in QGIS

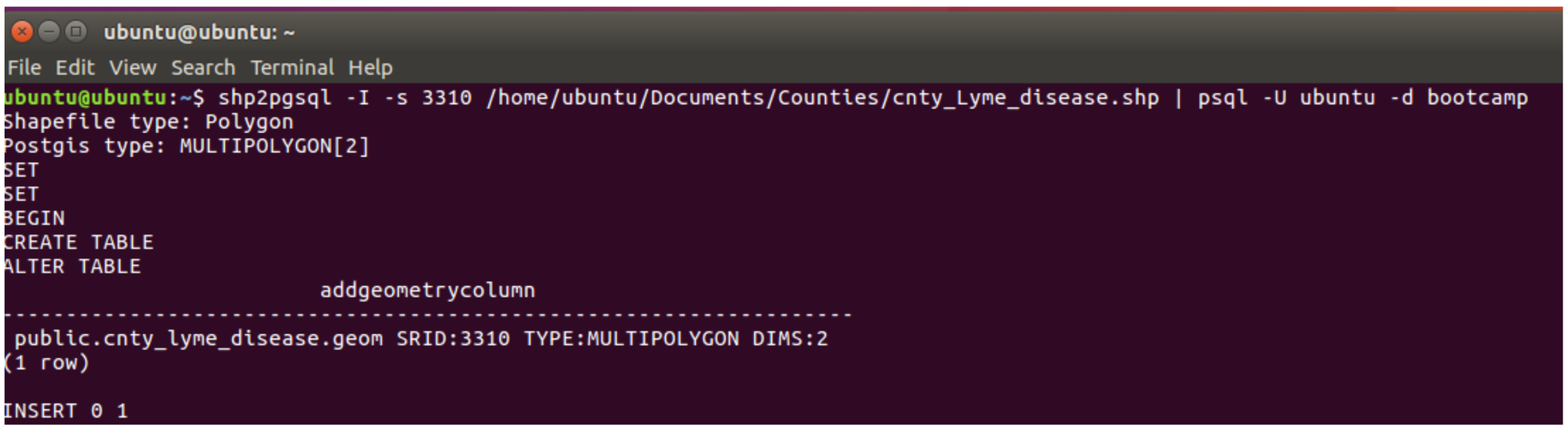
Tasks:

1. Load shapefiles using shp2pgsql command in terminal
2. Load rasters using raster2pgsql command in terminal
3. Expand SQL querying to include spatial functions
4. Connect to PostgreSQL via QGIS to query and visualize data
5. Create a new layer from SQL query in QGIS

Necessary items: terminal, pgAdmin, SQL file, QGIS

1. Part2_WritingSpatialQueries.sql

Task #1: Load shapefiles into PostgreSQL



In the terminal, type this command as one line and hit enter:

shp2pgsql -I -s 3310 /home/ubuntu/Documents/Counties/cnty_Lyme_disease.shp | pgsql –U ubuntu –d bootcamp

With this command, you are creating an object named cnty_Lyme_disease in the database named bootcamp. Parameter –I creates a spatial index, -s is the SRID (3310 for California Albers), -U is the user and –d is the database.

Task #1: Practice

```
ubuntu@ubuntu: ~
File  Edit  View  Search  Terminal  Help
ubuntu@ubuntu:~$ shp2pgsql -I -s 3310 /home/ubuntu/Documents/Counties/cnty_Lyme_disease.shp | psql -U ubuntu -d bootcamp
Shapefile type: Polygon
Postgis type: MULTIPOLYGON[2]
SET
SET
BEGIN
CREATE TABLE
ALTER TABLE
                          addgeometrycolumn
------------------------------------------------------------------
 public.cnty_lyme_disease.geom SRID:3310 TYPE:MULTIPOLYGON DIMS:2
(1 row)

INSERT 0 1
INSERT 0 1
INSERT 0 1
```

Repeat the command from the previous slide to load the following shapefiles:

1. CDPH_ticks_albers.shp from /home/ubuntu/Documents/Tick_Locations
    * This is a point dataset of known tick locations; same SRID
2. EPA_eco13_albers.shp from /home/ubuntu/Documents/Ecoregions
    * This is a polygon dataset of the EPA ecoregions for California; same SRID

More information on using shp2pgsql can be found here, here, and here.

Task #2: Load rasters into PostgreSQL

```
ubuntu@ubuntu:~$ raster2pgsql -s 3310 -t 128x128 /home/ubuntu/Documents/Land_Cover/NLCD2011_LC_albers.tif nlcd2011 | psql -U ubuntu
-d bootcamp
```

In the terminal, type this command as one line and hit enter:

raster2pgsql -s 3310 –t 128x128 /home/ubuntu/Documents/Land_Cover/NLCD2011_LC_albers.tif nlcd2011 | psql -U ubuntu -d bootcamp

With this command, you are creating an object named nlcd2011 in the database named bootcamp. Parameter -s is the SRID (3310 for California Albers), -t is the tile size to split the raster into table rows, -U is the user and –d is the database.

Task #2: Practice

```
ubuntu@ubuntu:~$ raster2pgsql -s 3310 -t 128x128 /home/ubuntu/Documents/Land_Cover/NLCD2011_LC_albers.tif nlcd2011 | psql -U ubuntu
-d bootcamp
```

Repeat the command from the previous slide to load the following rasters:

1. PRISM_vpdmin_30yr_normal_subset.tif
2. PRISM_vpdmax_30yr_normal_subset.tif
    * both in /home/ubuntu/Documents/Climate
    * These are rasters of min and max Vapor Pressure Deficit (the difference between the amount of moisture in the air and how much moisture the air can hold when it is saturated), which is thought to have a relationship with tick presence
    * SRID is 4269, which is unprojected NAD83

More information on using raster2pgsql can be found here and here.

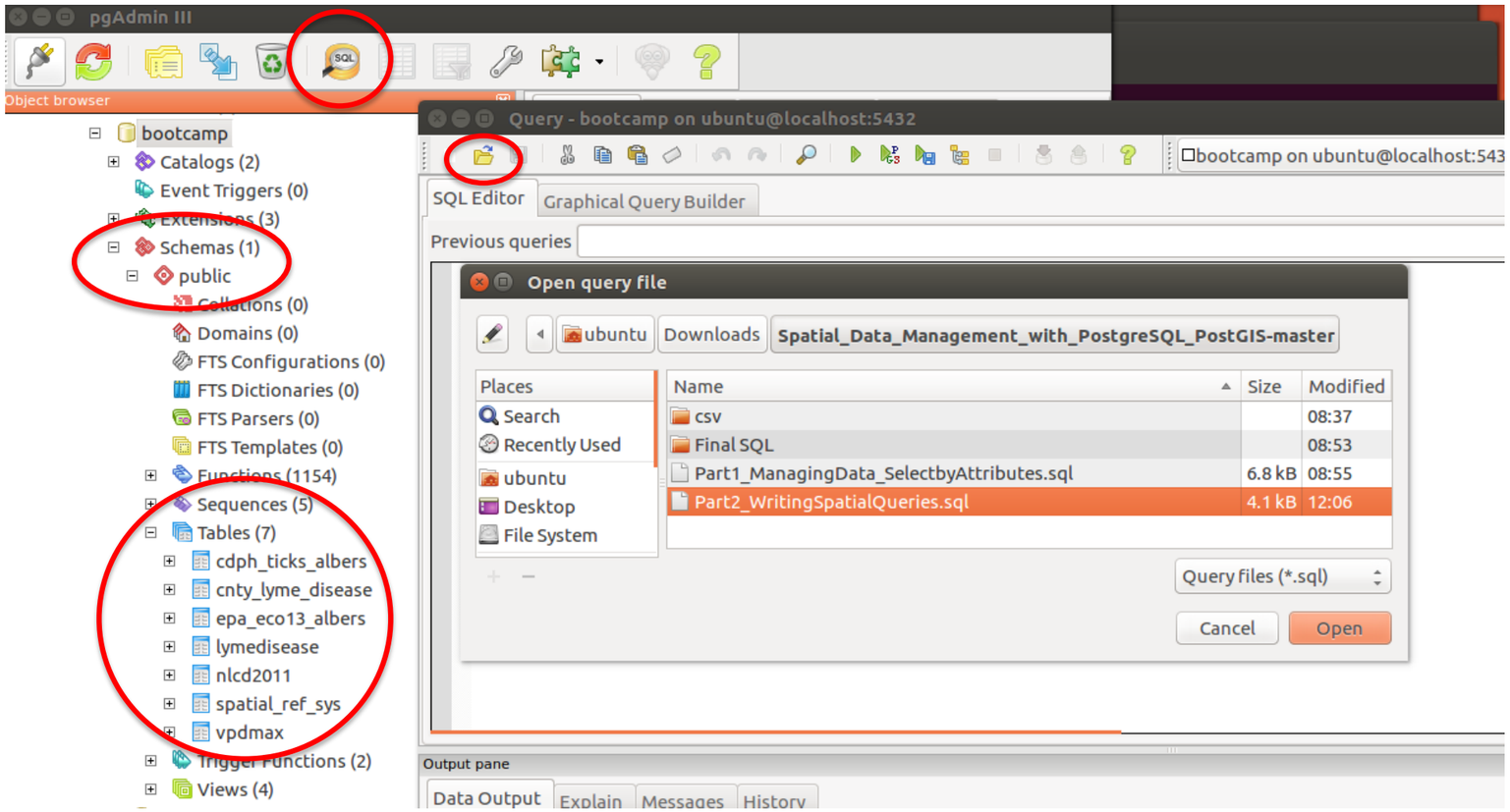
Task #3: Expand SQL Queries with Spatial Functions

Expand Schemas > public > Tables
to see the list of new objects.

Click on the SQL Window icon
again to open a new window.
Click on the Open file icon.

Navigate to the folder
downloaded from github and
select the sql file called
Part2_WritingSpatialQueries.sql.

Click open. Follow the exercise in the file.

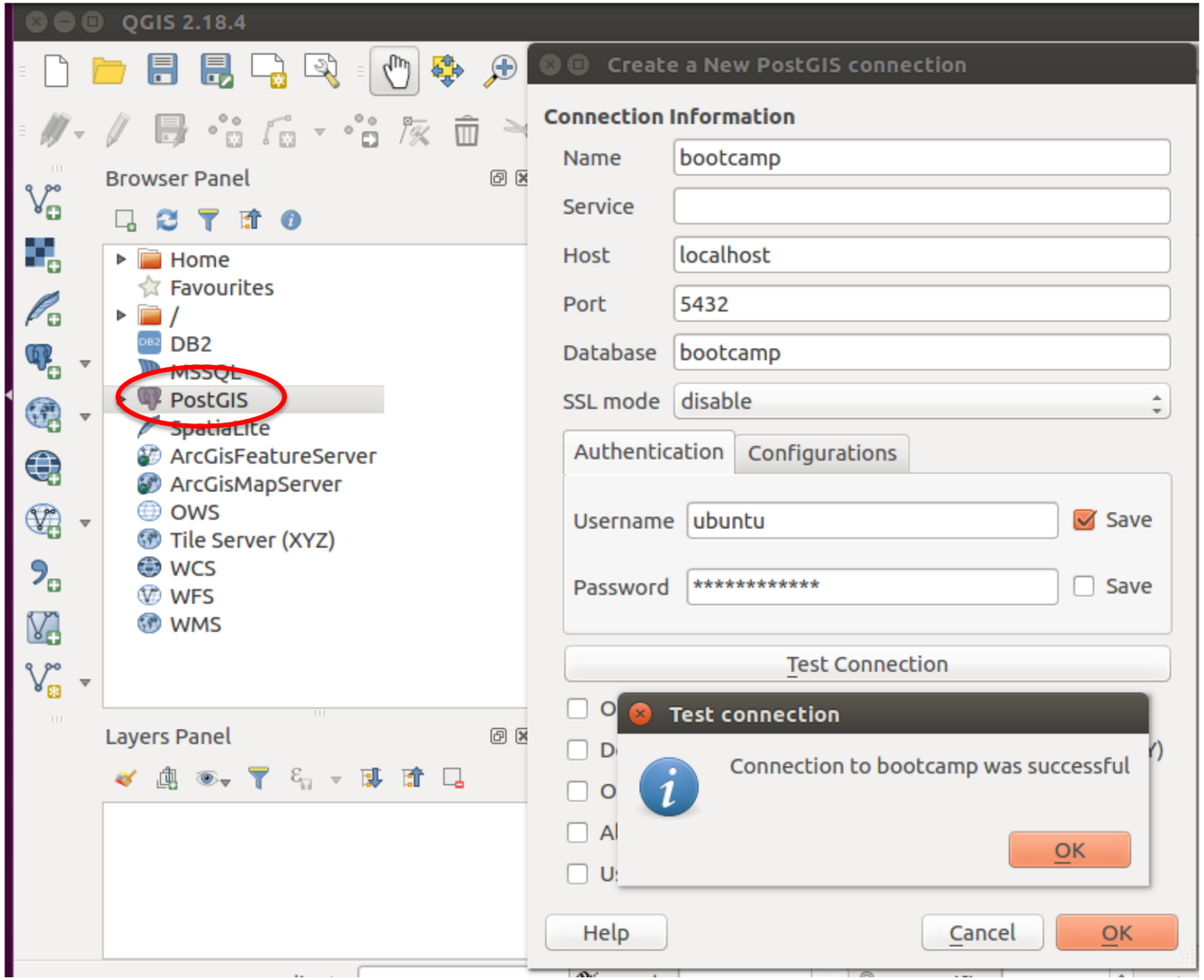


Task #4: Connect to PostgreSQL via QGIS

Click on the Search Computer icon and type QGIS. Double-click the QGIS icon to open it.

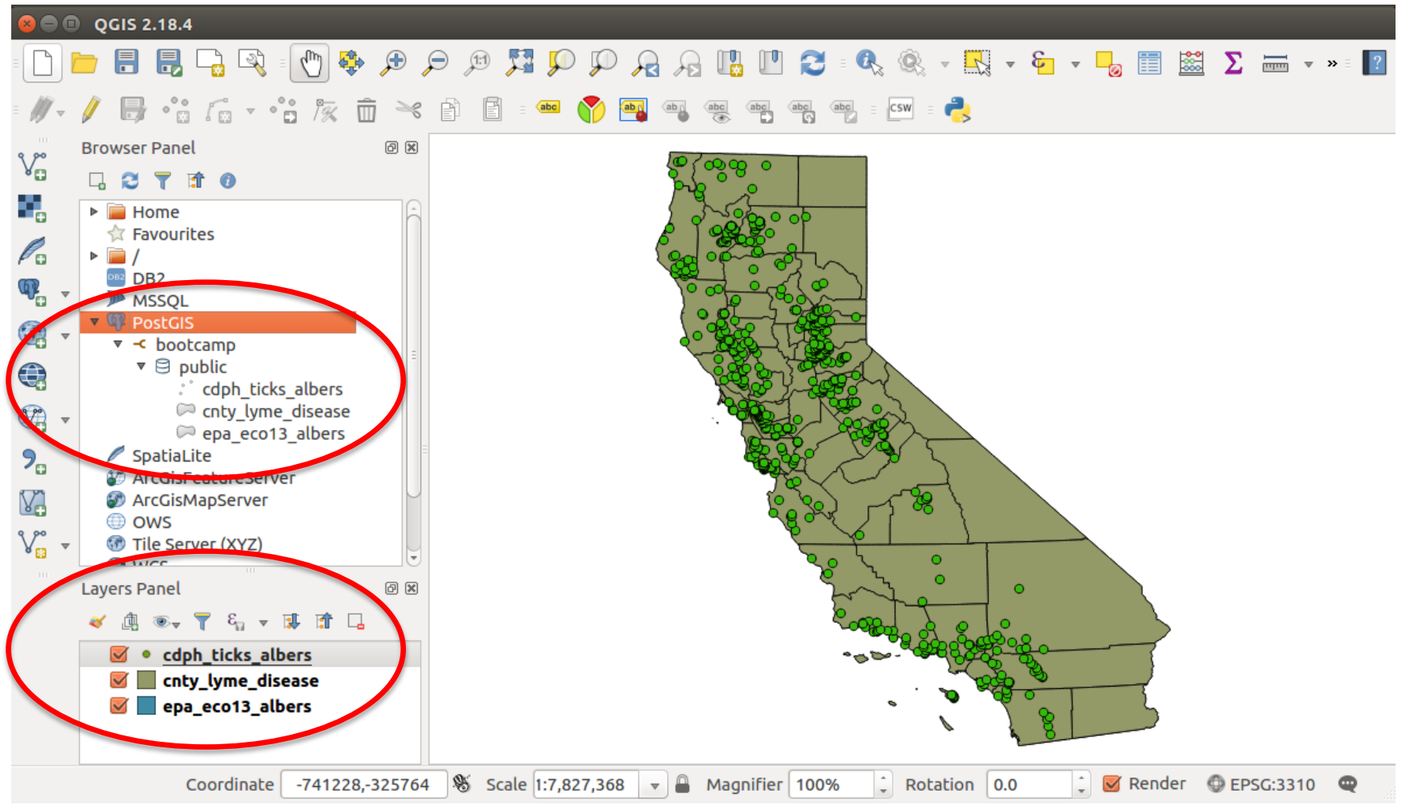
In the QGIS Browser menu on the left, right-click on PostGIS and select New Connection.

Fill out the window as shown and click Test Connection.



Task #4: Connect to PostgreSQL via QGIS (continued)

After successful message from Test Connection, expand PostGIS > bootcamp > public and drag **vector features** from the menu to the map.



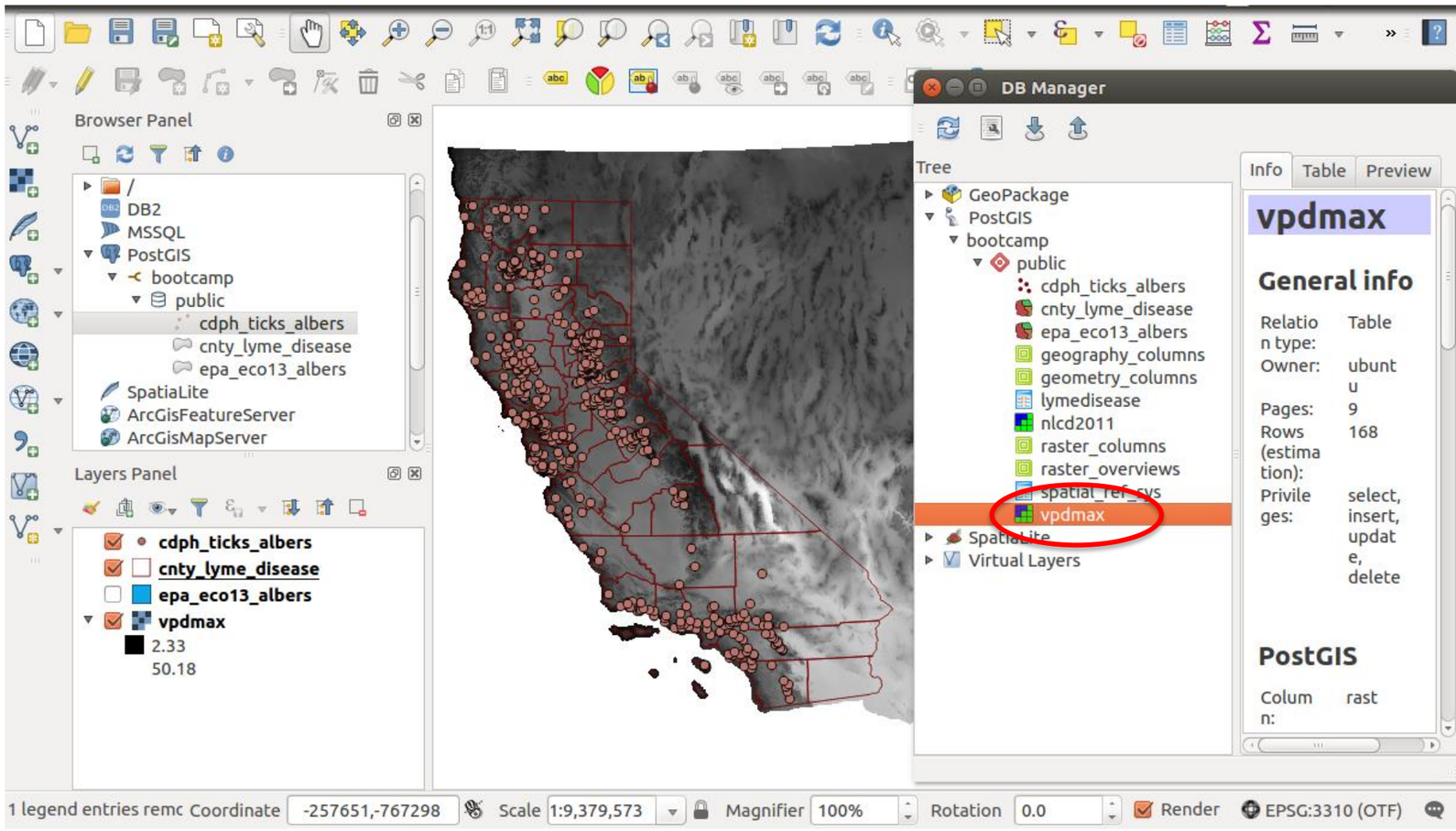
Right-click on the layer name in the Layers Panel on the bottom left, and select Properties to change the symbology or Open Attribute Table to see the table.

Task #4: continued for rasters

Hold your cursor over the words QGIS Desktop in the top left corner of the ubuntu desktop.

On the new menu that appears, select Database > DB Manager.

Expand the PostGIS connection and right-click on the raster name (ex: vpdmax) and select Add to Canvas.

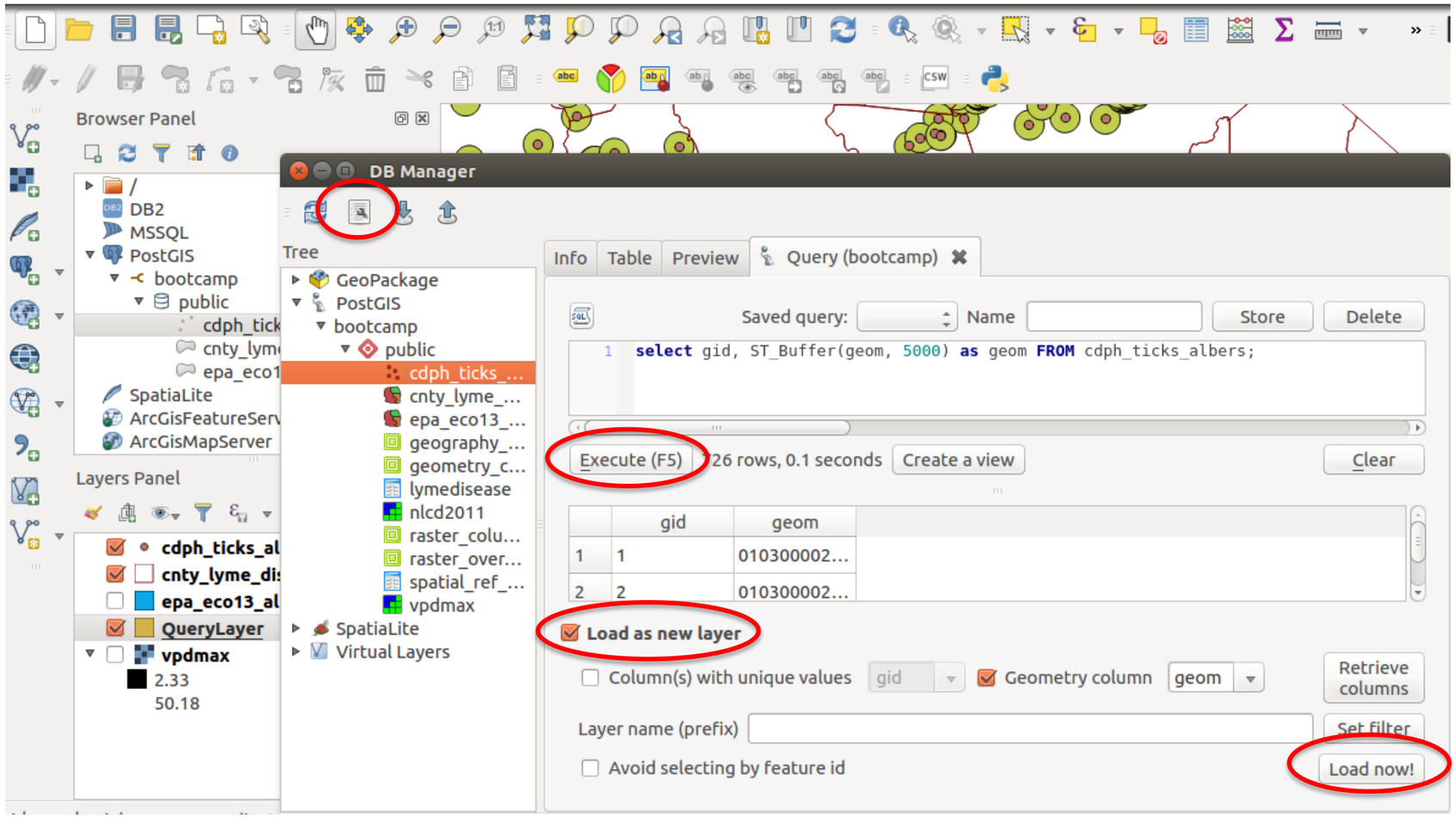


Task #5: Create a new layer from query in QGIS

In the DB Manager, click on the SQL window icon (circled).

Copy the first query from the last section of Part2_WritingSpatial Queries.sql.

Click Execute and then Load Now! so that you can add the new layer to the map. Repeat with the other queries.



Bonus Activity: test your new skills with other data

Be sure to check the projections so you can identify the SRIDs.

Note: SFPD incidents are in WGS84

Additional Resources for Working with Rasters

http://postgis.net/docs/using_raster_dataman.html

http://postgis.net/docs/manual-2.1/RT_reference.html#Raster_Management_Functions

http://www.postgis.net/docs/RT_ST_SummaryStats.html

Additional Resources for Spatial Queries

http://www.postgis.us/foss4gna2015

https://2015.foss4g-na.org/session/advanced-spatial-analysis-postgis.html

https://2015.foss4g-na.org/session/magical-postgis-three-brief-movements.html

https://2015.foss4g-na.org/session/postgis-feature-frenzy.html

Followup: Open PostgreSQL to non-localhost users

Work with the network administrator to get an externally facing IP for the server that has PostgreSQL

Modify pg_hba.conf to allow user connections to databases:
http://www.postgresql.org/docs/9.4/static/auth-pg-hba-conf.html

Modify postgresql.conf to allow external connections to PostgreSQL:
http://www.postgresql.org/docs/9.4/static/runtime-config-connection.html

Spatial Data Science Bootcamp

Set Up Your Environment

- Virtual machine environments (Linux-based)
- Spatial databases (PostgreSQL/PostGIS)

Wrangle Data

- Modern data standards and formats (GeoJSON)
- Open mapping tools (GDAL, QGIS)

Analyze Data

- Python (i.e. PySAL, NumPy, Pandas, Jupyter Notebook)
- R and R Studio (i.e. sp, raster, rgdal, maptools)

Visualize and Publish Data

- Web mapping and data visualizations (CARTO, Leaflet, D3)
- Data Visualization in Python

Spatial Data Science Bootcamp

Set Up Your Environment

- Virtual machine environments (Linux-based)
- Spatial databases (PostgreSQL/PostGIS)

Wrangle Data

- Modern data standards and formats (GeoJSON)
- Open mapping tools (GDAL, QGIS)

Analyze Data

- Python (i.e. PySAL, NumPy, Pandas, Jupyter Notebook)
- R and R Studio (i.e. sp, raster, rgdal, maptools)

Visualize and Publish Data

- Web mapping and data visualizations (CARTO, Leaflet, D3)
- Data Visualization in Python

Day 2: Spatial Data Analysis with Python and R

Morning Agenda:

1. Presentation: Fundamentals of Scripting (Python and R)
2. Presentation: Introduction to Spatial Autocorrelation
3. Hands-on Tutorial #1: Spatial Autocorrelation in Python

Afternoon Agenda:

1. Hands-on Tutorial #2: Spatial Data Processing with R
2. Hands-on Tutorial #3: Regression and Modeling with R