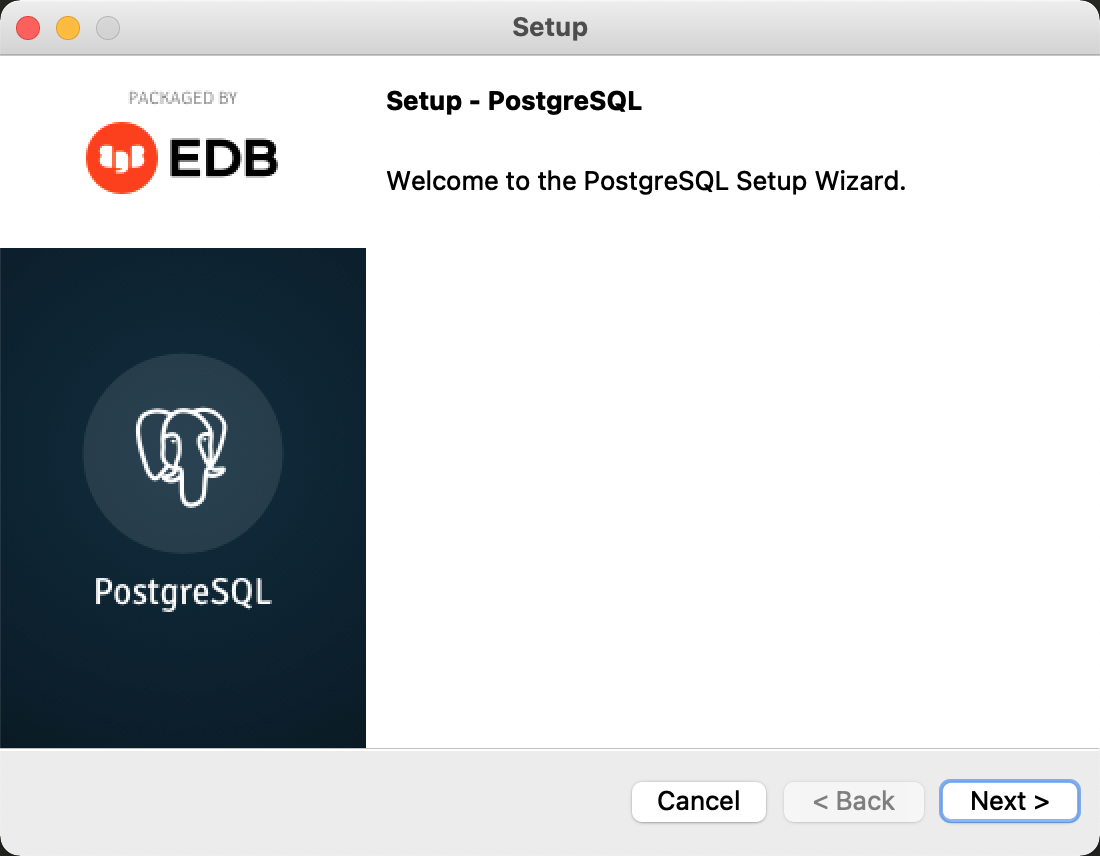
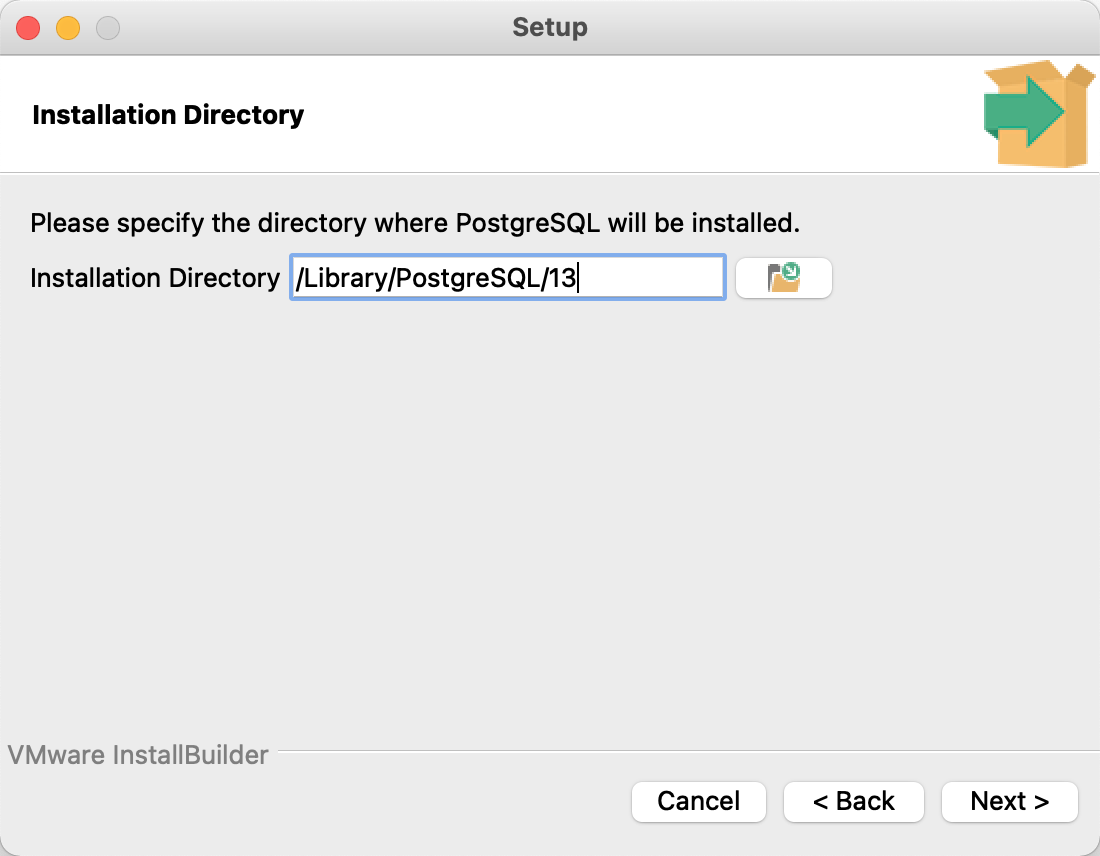
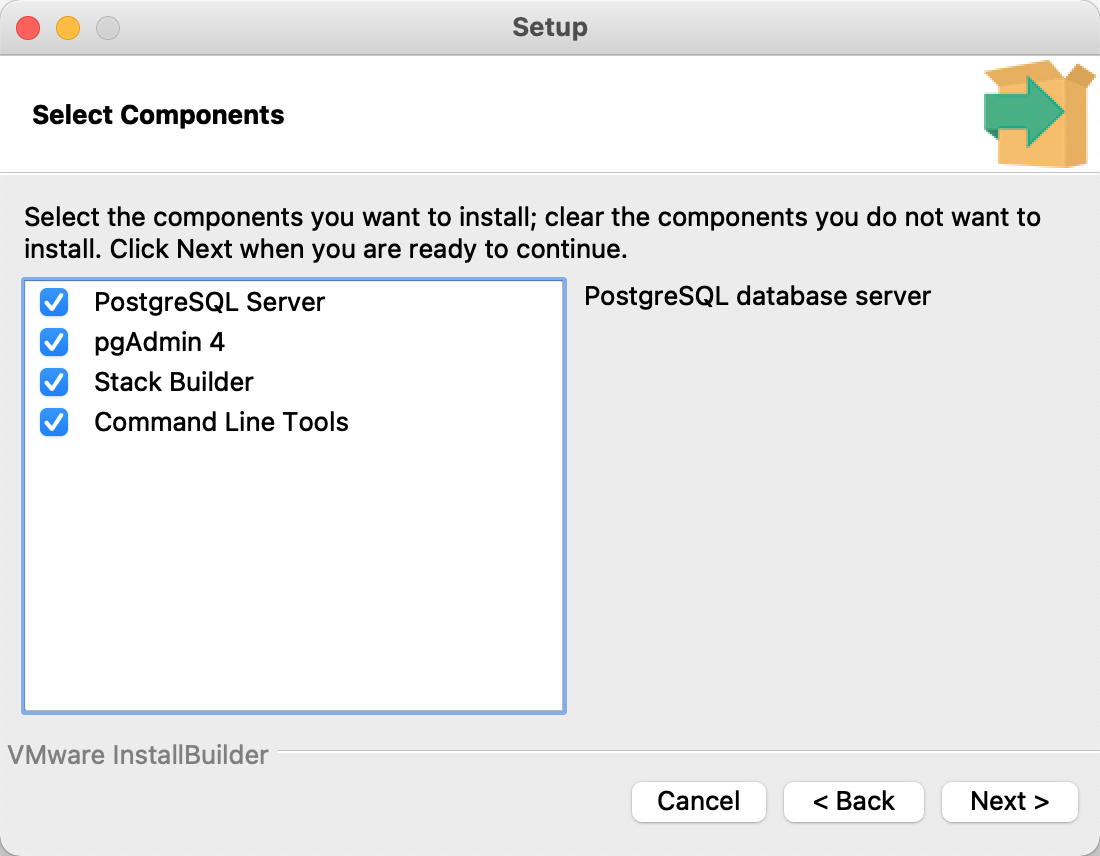
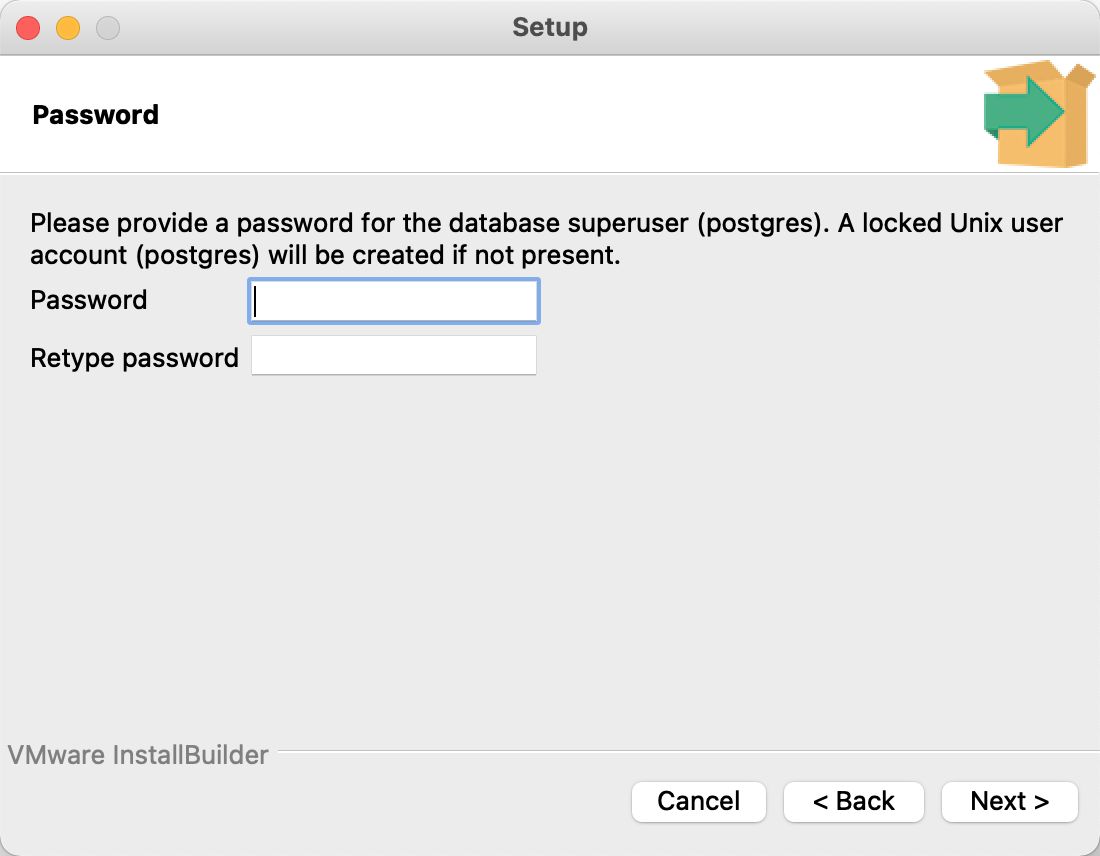
**1. Configure the Postgres/PostGIS**

Download the Postgres/PostGIS, <https://www.postgresql.org/download/>

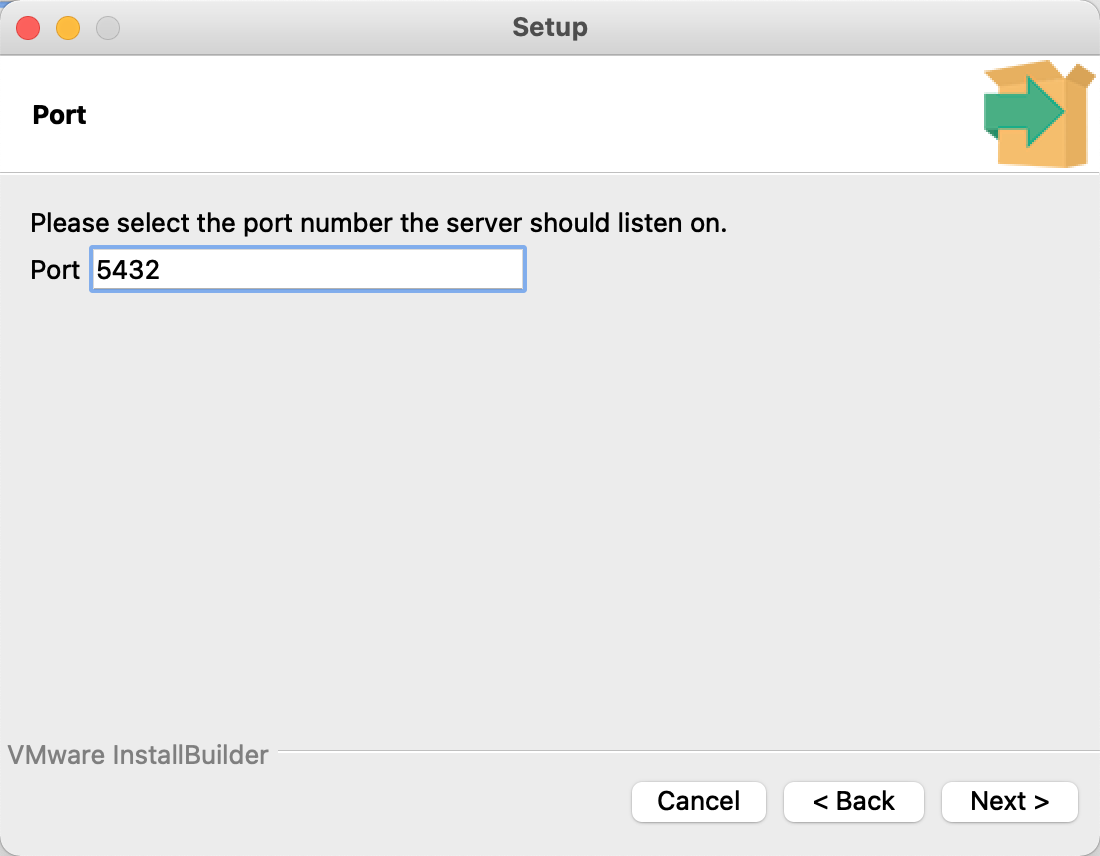


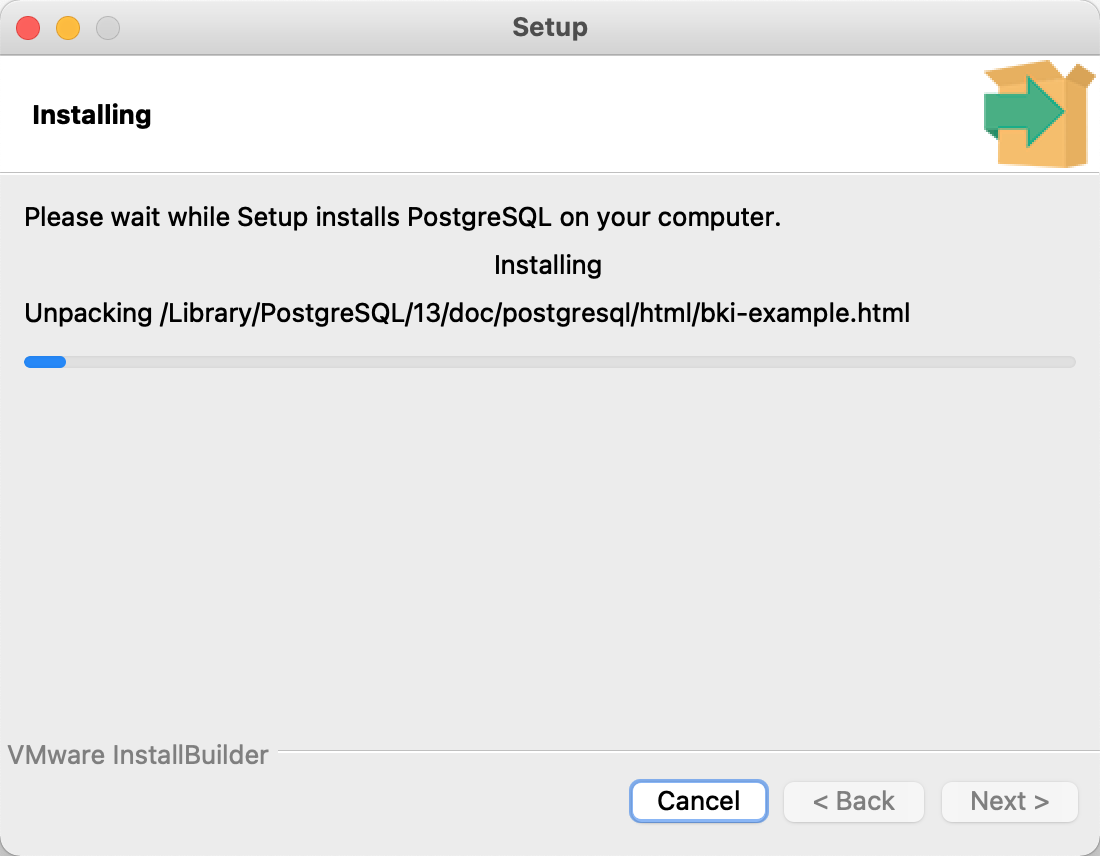


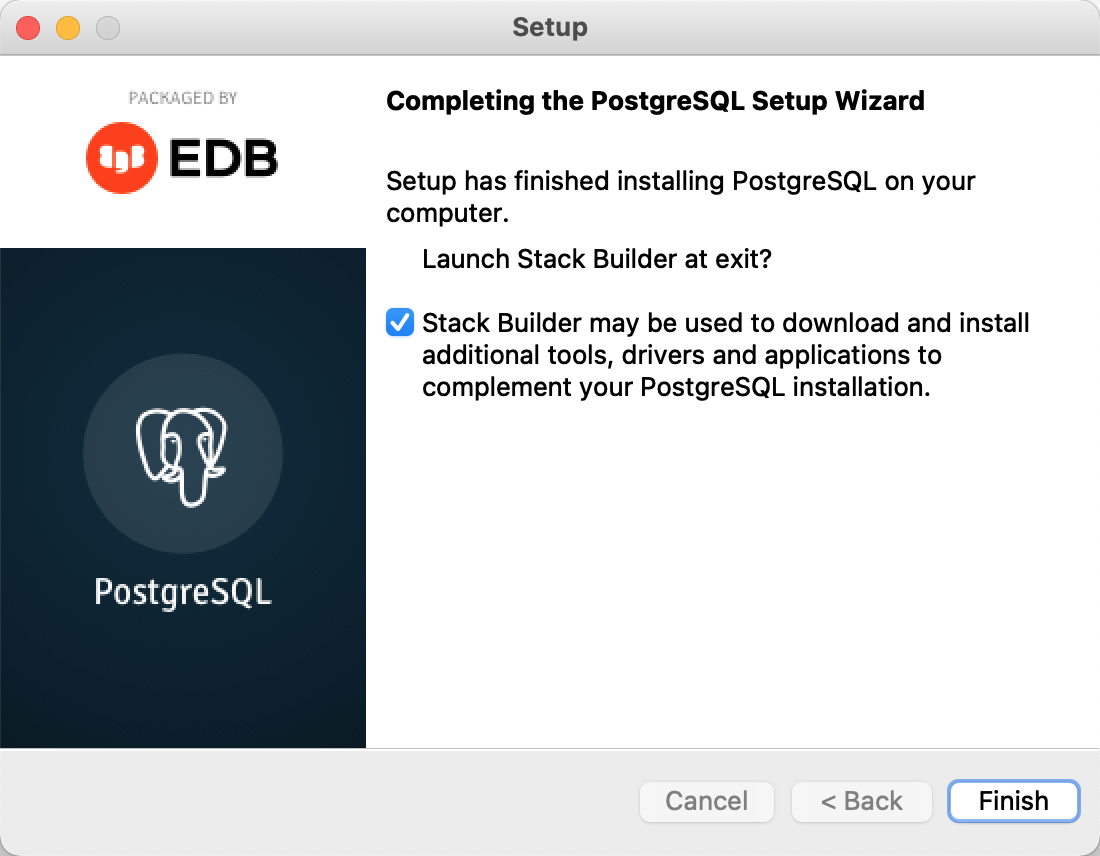


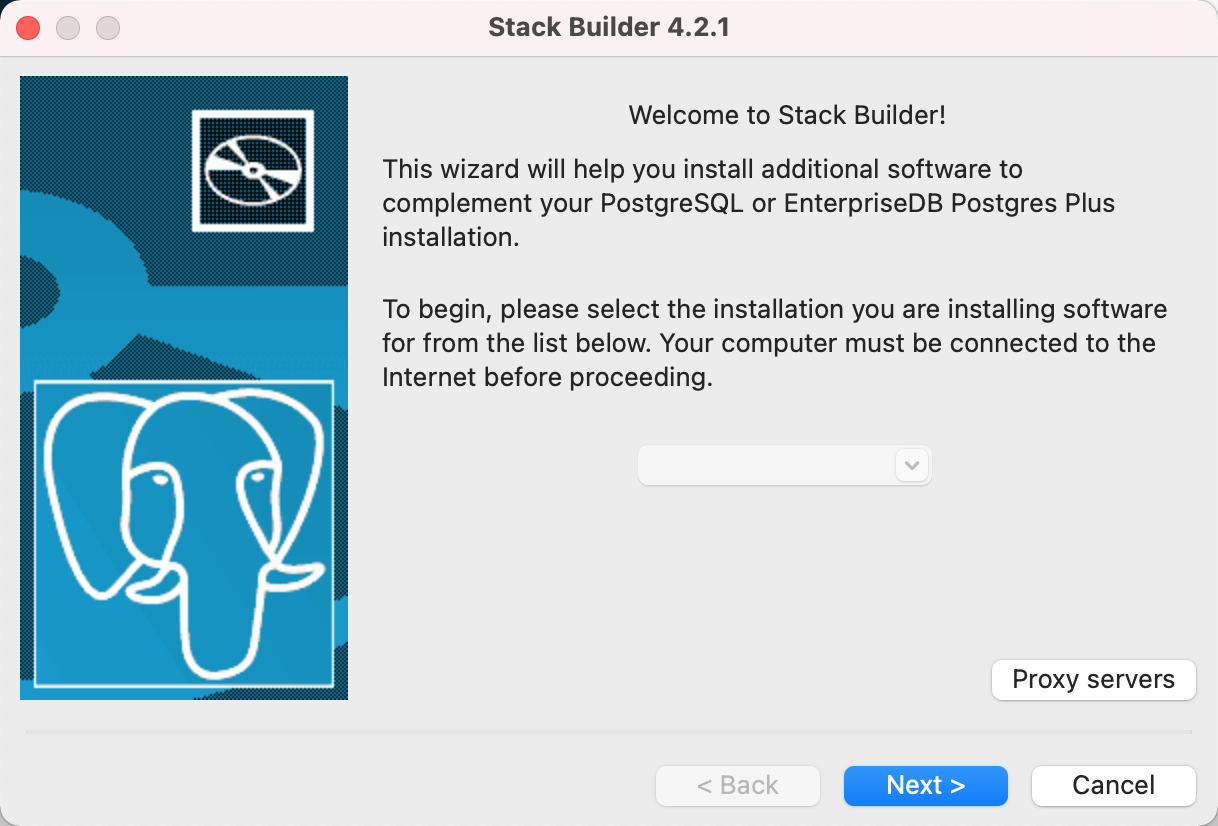


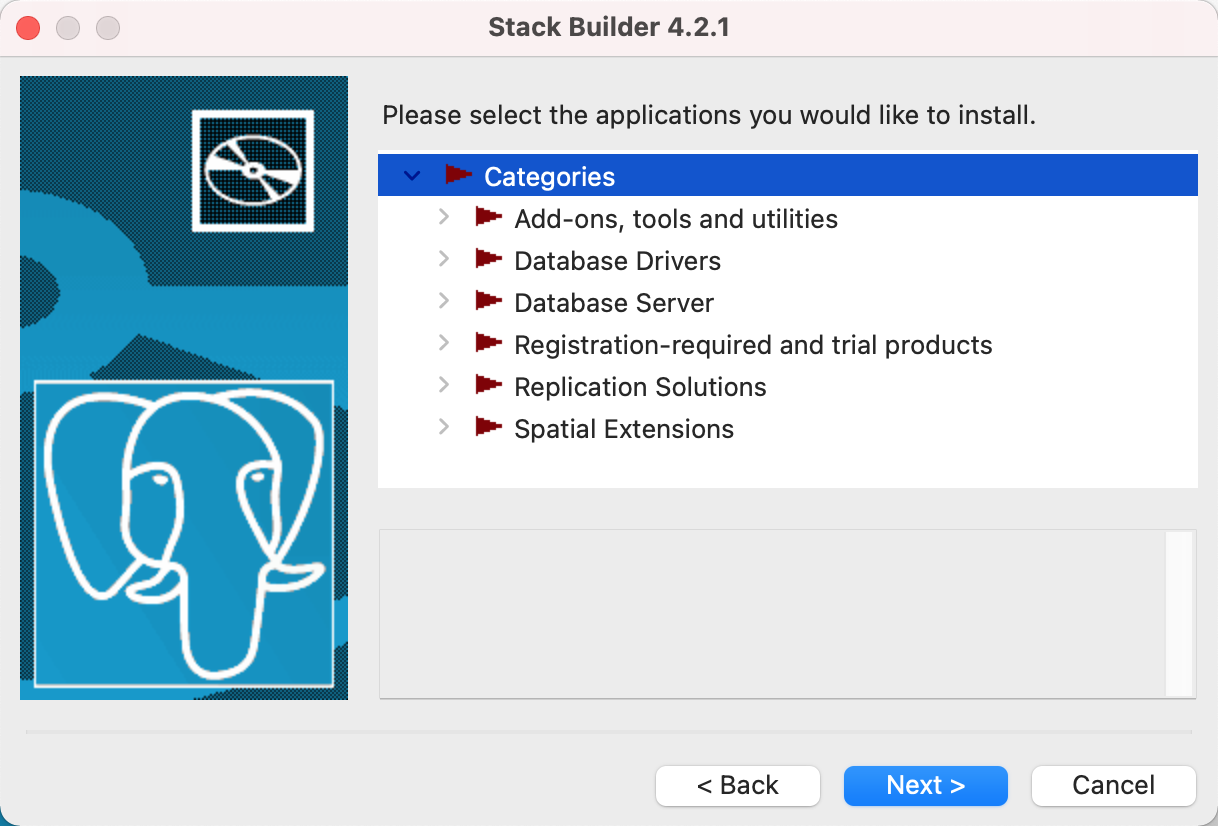
Password: 1234

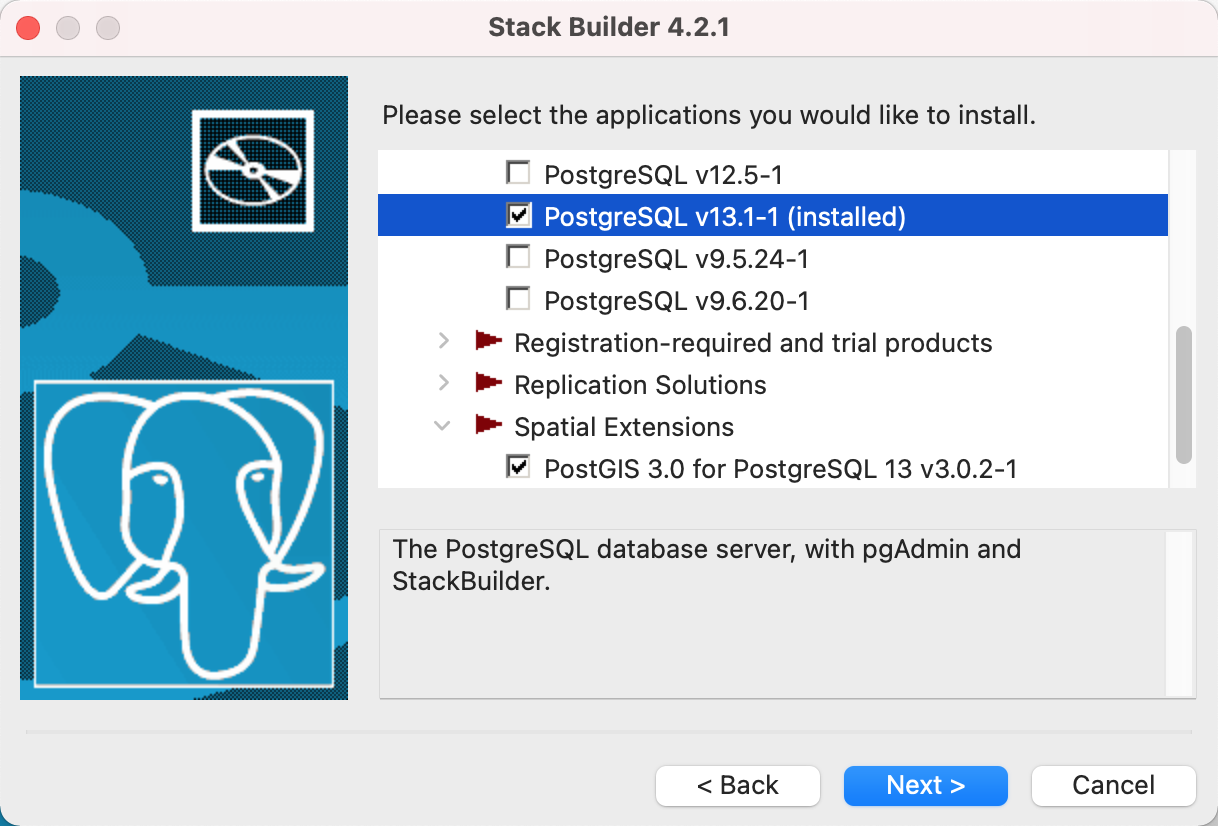






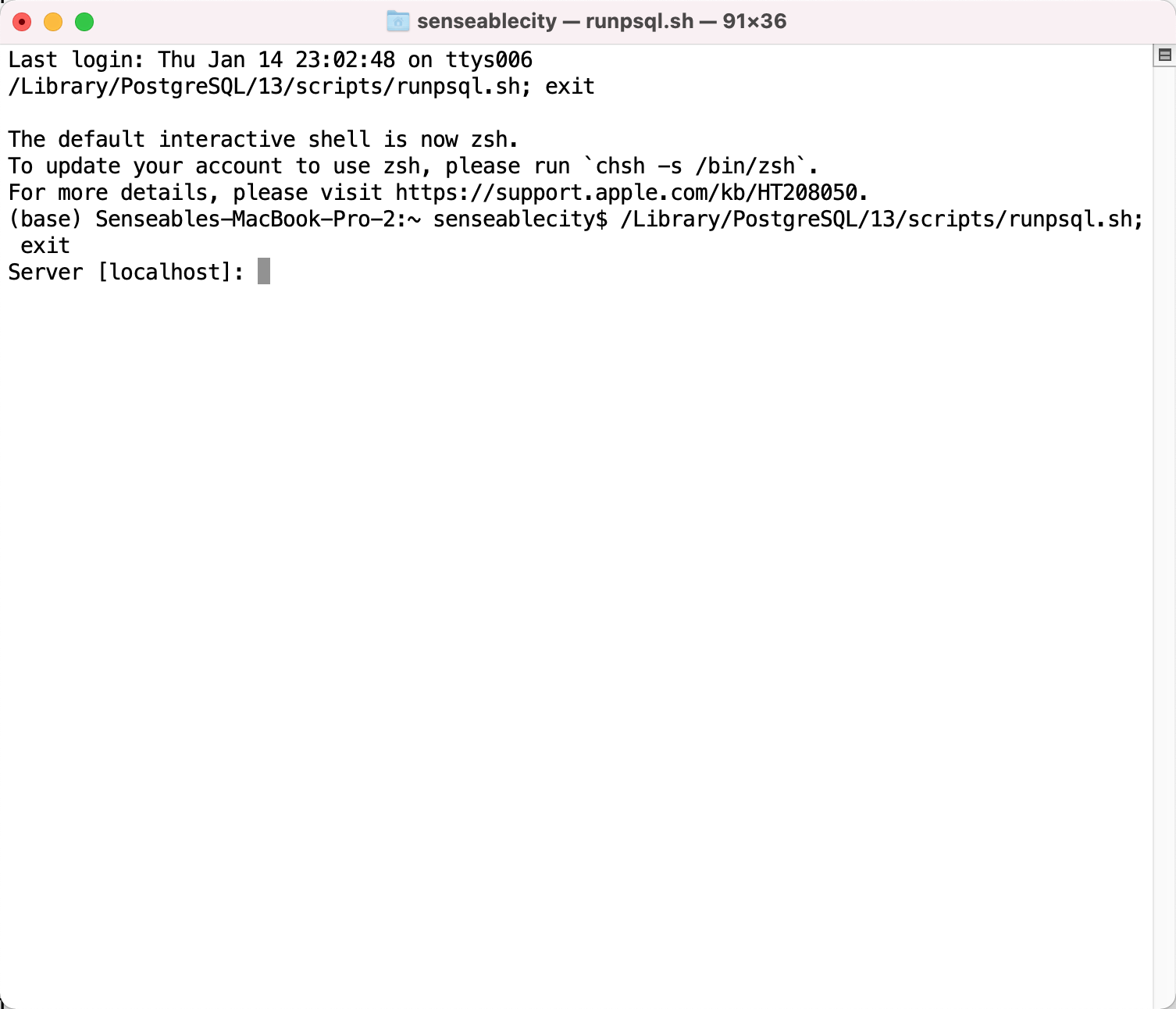




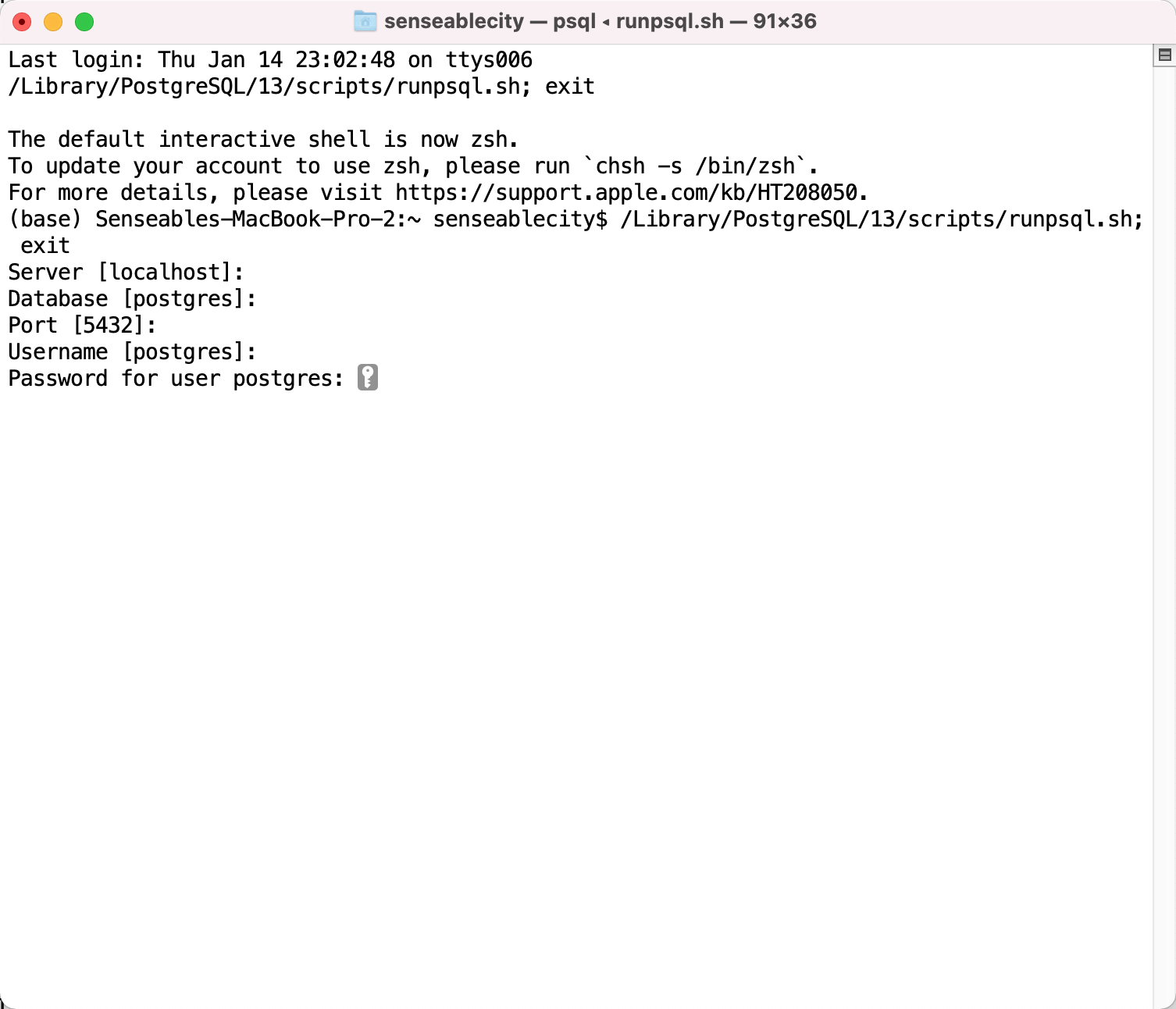


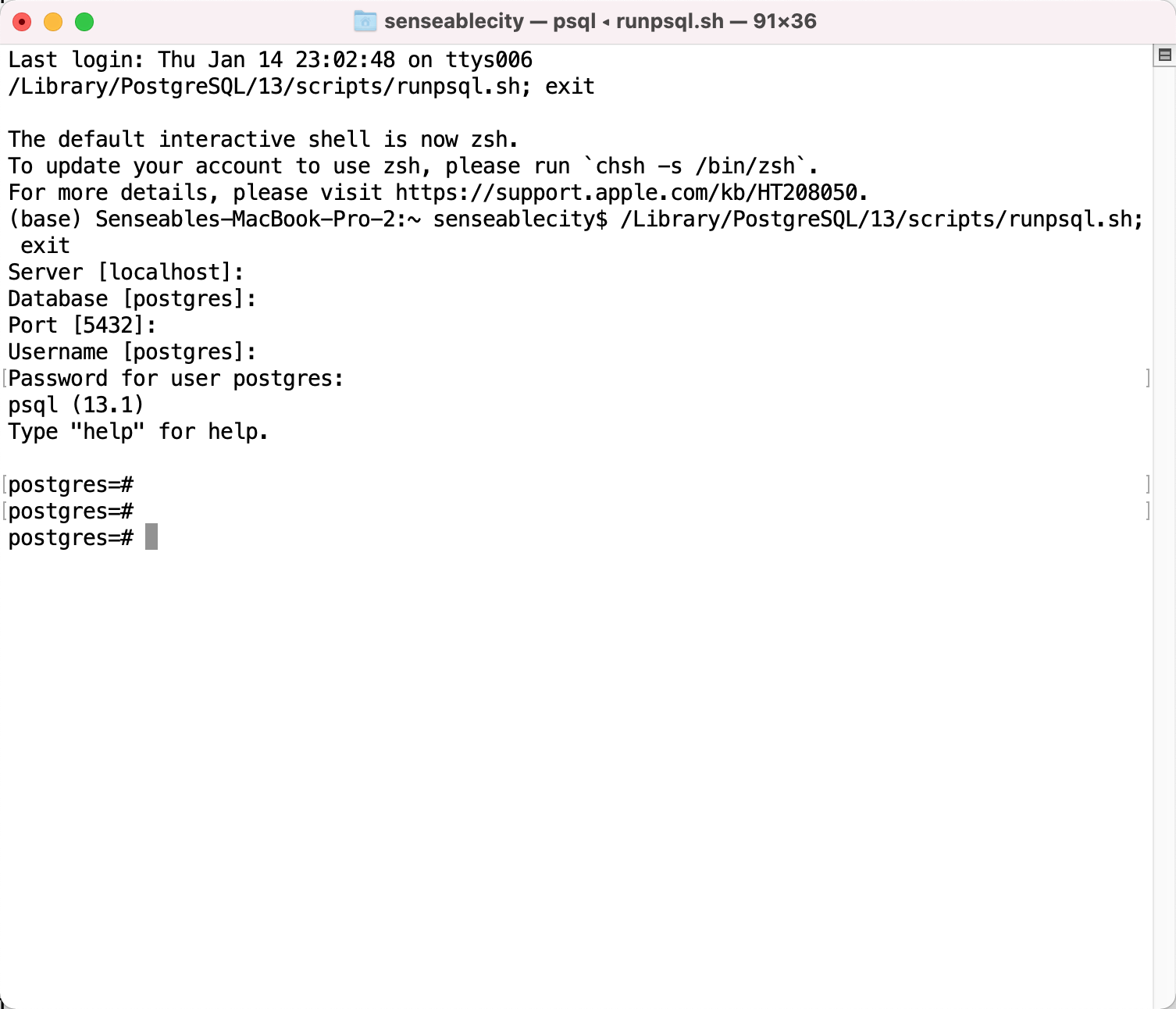
**2. Be familiar with the database**

Open PSQL terminal,



Typein your password,





2.3.1 Create tables and db opearations

List all databases,

\l

List all tables,

\c dbname

\dt

Create a PostGIS table for the Runkeeper metadata,

CREATE EXTENSION postgis;

CREATE TABLE IF NOT EXISTS boston

(tripid VARCHAR(40) PRIMARY KEY NOT NULL,

userid VARCHAR(40),

rtype VARCHAR(15),

age VARCHAR(5),

gender VARCHAR(1),

distance REAL,

startT VARCHAR(20),

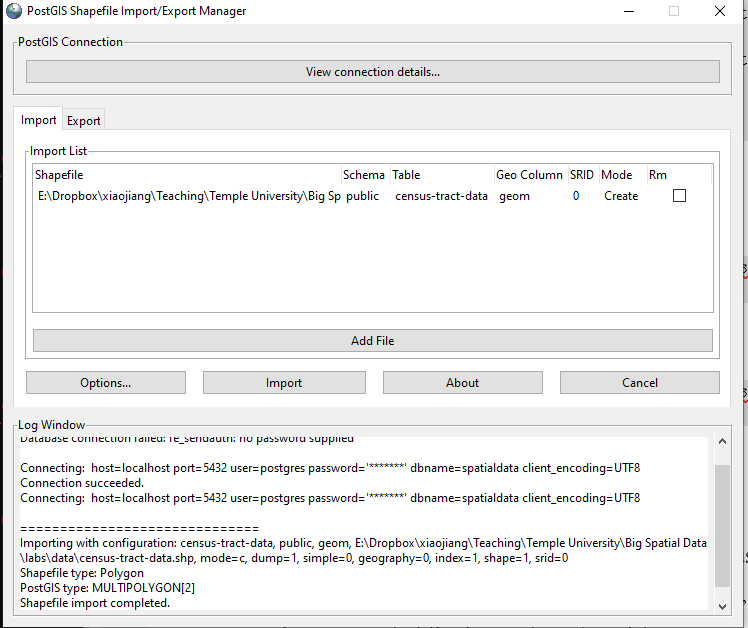
endT VARCHAR(20),

geom geometry(LineString, 3857));

**3. Convert the shapefile into database,**

Download the tool for convert the shapefile to spatial data base table,

<http://download.osgeo.org/postgis/windows/>



**4. Write SQL to query the table**

Enable PostGIS.

Create extension of PostGIS

CREATE EXTENSION IF NOT EXISTS PostGIS CASCADE;

Then you can check the version of the installed PostGIS,

SELECT PostGIS\_version();

Create a database,

create DATABASE runners;

CREATE EXTENSION IF NOT EXISTS PostGIS CASCADE;

SELECT PostGIS\_version();

CREATE TABLE IF NOT EXISTS TEST

(id VARCHAR(30) PRIMARY KEY NOT NULL,

year VARCHAR(10),

month VARCHAR(2),

city VARCHAR(20));

INSERT INTO TEST (id, year, month, city) VALUES('1', '2015', '07', 'Philadelphia');

INSERT INTO TEST (id, year, month, city) VALUES('2', '2016', '07', 'Boston');

INSERT INTO TEST (id, year, month, city) VALUES('3', '2017', '07', 'New York City');

INSERT INTO TEST (id, year, month, city) VALUES('4', '2018', '07', 'Houston');

SELECT \* FROM TEST;

Add the geom field to the table,

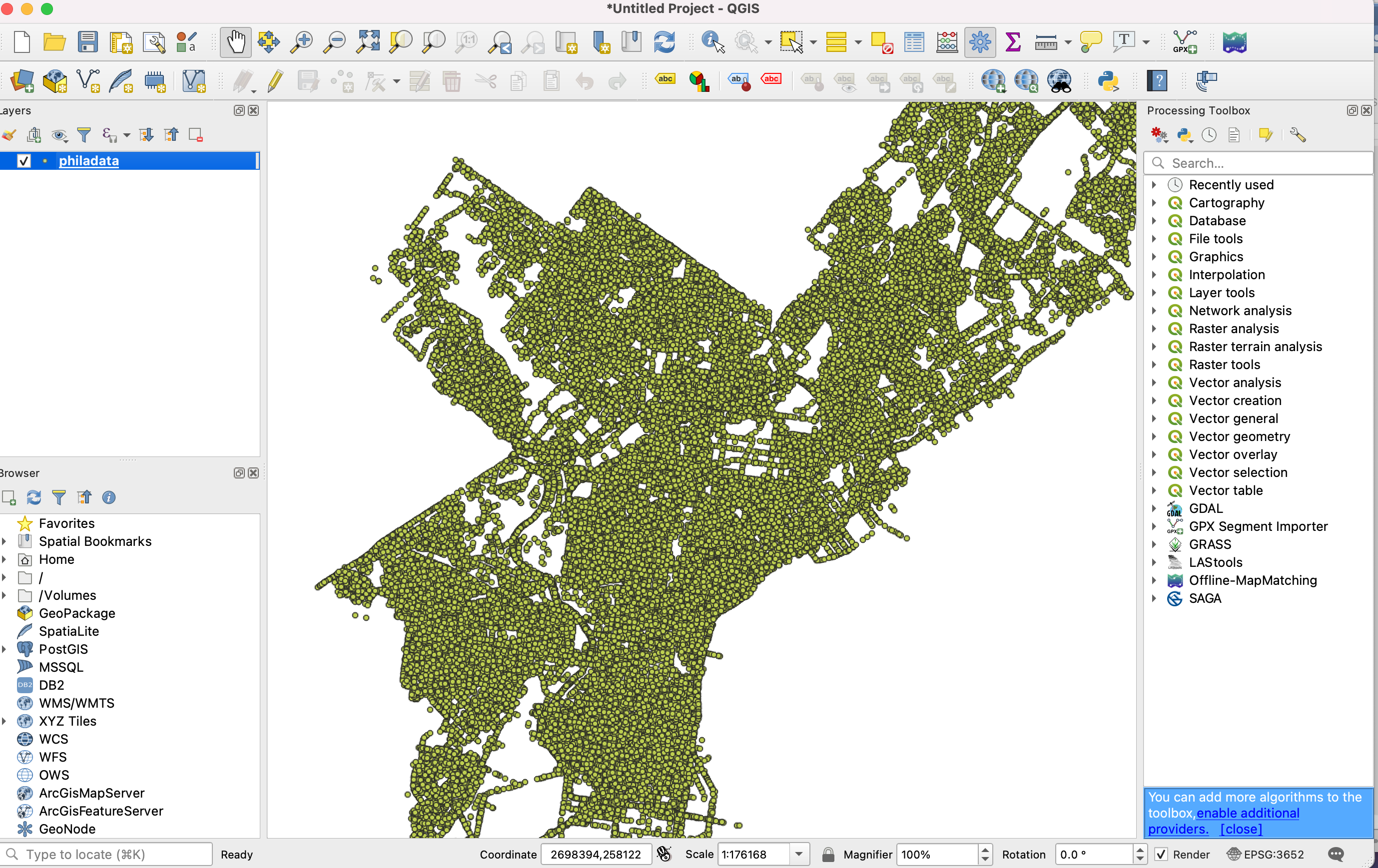
ALTER TABLE gvi\_phily\_yale ADD COLUMN geom geometry(Point, 4326);

UPDATE gvi\_phily\_yale SET geom = ST\_SetSRID(ST\_MakePoint(lon, lat), 4326);

Save the PostGIS database into a shapefile,

pgsql2shp -u postgres -h localhost -P 5424796 -f philadata geospatial "SELECT \* FROM gvi\_phily\_yale;"

Here is your results look like,



You can save the shapefile as a spatial data base using the above tool,

It is usually way more efficient and flexible to use database to manage the trajectory data. Therefore, different from the previous version code, here we can use the database to manage the huge amount of Runkeeper trajectory data.

**5. Access the database table using Python**

You can use Python to access the database and manipulate the table,

You need to install psycopg2 module first by using,

pip install psycopg2

Then you can open your Jupyter Notebook and access the database table,

import psycopg2

conn = psycopg2.connect(host="localhost",database="spatialdata", user="postgres", password="5424796")

You can also use Python to call the SQL statement like the following statement,

import psycopg2

# user = 'xiaojiang@urbanenv'

# host = 'urbanenv.postgres.database.azure.com'

# port = "5432"

# db\_filename = "panodb"

# tb\_name = "millionTreeNYC"

# conn = psycopg2.connect(database = db\_filename, user = user, host = host, port = port, password="@LIxiao5424796")

# cursor = conn.cursor()

conn = psycopg2.connect(host="localhost",database="build2park", \

user="postgres", password="5424796")

cur = conn.cursor()

Create table for save the Runkeeper data in Boston,

import psycopg2

# user = 'xiaojiang@urbanenv'

# host = 'urbanenv.postgres.database.azure.com'

# port = "5432"

# db\_filename = "panodb"

# tb\_name = "millionTreeNYC"

# conn = psycopg2.connect(database = db\_filename, user = user, host = host, port = port, password="@LIxiao5424796")

# cursor = conn.cursor()

conn = psycopg2.connect(host="localhost",database="runners", \

user="postgres", password="5424796")

cur = conn.cursor()

cur.execute('''CREATE TABLE IF NOT EXISTS boston

(tripid VARCHAR(40) PRIMARY KEY NOT NULL,

userid VARCHAR(40),

rtype VARCHAR(15),

age VARCHAR(5),

gender VARCHAR(1),

distance REAL,

startT VARCHAR(20),

endT VARCHAR(20));

''')

conn.commit()

2.3.2 Insert the metadata of the runners to the table,

import psycopg2

# connect to the postgres database in localhost

conn = psycopg2.connect(host="localhost",database="runners", \

user="postgres", password="5424796")

cursor = conn.cursor()

tripid = '92a44192cd98aa28949c02c3e14e8d37'

userid = '628fccca952b3eca9343f0441f8908ea'

gender = 'M'

age = '25-34'

triptype = 'Running'

startT = '2013-08-10 08:34:22'

endtime = '2013-08-10 09:03:57'

distance = 4182.2831187888505

query\_statement = "INSERT INTO %s (tripid, userid, rtype, age, gender, distance, startT, endT) \

VALUES ('%s', '%s', '%s', '%s', '%s', %f, '%s', '%s')"\

%('boston', tripid, userid, triptype, age, gender, float(distance), startT, endtime)

print('Query statement', query\_statement)

cursor.execute(query\_statement)

conn.commit()

**Insert the geom column value from the LineString,**

line\_lst = [(-7910721.245572232, 5215245.223501071), (-7910687.070488571, 5215244.319580916), (-7910687.070488571, 5215244.319580916), (-7910658.350059941, 5215243.566314173), (-7910643.989845624, 5215242.5117408475), (-7910626.957963548, 5215241.306514313), (-7910601.688439128, 5215237.690835631), (-7910574.415163898, 5215231.363401256), (-7910527.327019311, 5215213.285040632), (-7910527.327019311, 5215213.285040632), (-7910485.470890775, 5215189.331265961), (-7910485.470890775, 5215189.331265961), (-7910442.835525784, 5215160.8579900395), (-7910410.107595506, 5215131.782193456), (-7910376.823067753, 5215088.695944117), (-7910359.90250513, 5215053.443704142), (-7910346.5441662315, 5215012.165607111), (-7910342.536664541, 5214980.529231224), (-7910342.536664541, 5214980.529231224), (-7910341.312150148, 5214953.7137192115), (-7910342.870622999, 5214927.500876394), (-7910347.100763646, 5214902.041345674), (-7910357.564795745, 5214860.161319831), (-7910363.798687228, 5214843.740784746), (-7910373.928760876, 5214821.445057893), (-7910381.498486238, 5214806.380407321), (-7910394.188908193, 5214784.687352609), (-7910402.426550486, 5214771.731801981), (-7910402.426550486, 5214771.731801981), (-7910433.373368923, 5214728.496546225), (-7910482.576583813, 5214662.363557921)]

epsgcode = '3857'

linstring = 'LINESTRING('

for coord in line\_lst:

lon = coord[0]

lat = coord[1]

linstring = linstring + str(lon) + ' ' + str(lat) + ', '

linstring = linstring[:-2] + ')'

geom\_wkb = "ST\_GeomFromText('%s', %s)"%(linstring, epsgcode)

geom\_wkb

You geom\_wkb would look like this,

"ST\_GeomFromText('LINESTRING(-7910721.245572232 5215245.223501071, -7910687.070488571 5215244.319580916, -7910687.070488571 5215244.319580916, -7910658.350059941 5215243.566314173, -7910643.989845624 5215242.5117408475, -7910626.957963548 5215241.306514313, -7910601.688439128 5215237.690835631, -7910574.415163898 5215231.363401256, -7910527.327019311 5215213.285040632, -7910527.327019311 5215213.285040632, -7910485.470890775 5215189.331265961, -7910485.470890775 5215189.331265961, -7910442.835525784 5215160.8579900395, -7910410.107595506 5215131.782193456, -7910376.823067753 5215088.695944117, -7910359.90250513 5215053.443704142, -7910346.5441662315 5215012.165607111, -7910342.536664541 5214980.529231224, -7910342.536664541 5214980.529231224, -7910341.312150148 5214953.7137192115, -7910342.870622999 5214927.500876394, -7910347.100763646 5214902.041345674, -7910357.564795745 5214860.161319831, -7910363.798687228 5214843.740784746, -7910373.928760876 5214821.445057893, -7910381.498486238 5214806.380407321, -7910394.188908193 5214784.687352609, -7910402.426550486 5214771.731801981, -7910402.426550486 5214771.731801981, -7910433.373368923 5214728.496546225, -7910482.576583813 5214662.363557921)', 3857)"

QUINUH Ngyuen

Steven Mooney

You can then insert the geom\_wkb into the PostGIS table.

import psycopg2

# connect to the postgres database in localhost

conn = psycopg2.connect(host="localhost",database="runners", \

user="postgres", password="5424796")

cursor = conn.cursor()

tripid = '92a44192cd98aa28949c02c3e14e8d37'

userid = '628fccca952b3eca9343f0441f8908ea'

gender = 'M'

age = '25-34'

triptype = 'Running'

startT = '2013-08-10 08:34:22'

endtime = '2013-08-10 09:03:57'

distance = 4182.2831187888505

geom = "ST\_GeomFromText('LINESTRING(-74.036486 40.730121, -74.03125 40.72882, -74.011123 40.725958)', 3857)"

query\_statement = "INSERT INTO %s (tripid, userid, rtype, age, gender, distance, startT, endT, geom) \

VALUES ('%s', '%s', '%s', '%s', '%s', %f, '%s', '%s', %s)"\

%('boston', tripid, userid, triptype, age, gender, float(distance), startT, endtime, geom\_wkb)

# query\_statement = "INSERT INTO boston (tripid, userid, rtype, age, gender, distance, startT, endT, geom) VALUES ('000f4091cb692b8e6ef321b25473d1fa', '628fccca952b3eca9343f0441f8908ea', 'Running', '25-34', 'M', 4182.283119, '2013-08-10 08:34:22', '2013-08-10 09:03:57', ST\_GeomFromText('LINESTRING(-74.036486 40.730121, -74.03125 40.72882, -74.011123 40.725958)', 3857))"

print('Query statement', query\_statement)

cursor.execute(query\_statement)

conn.commit()

2.3.3 Convert the PostGIS db into a shapefile,

You can then export the PostGIS table into a shapefile locally

pgsql2shp -u postgres -h localhost -P 5424796 -f runner runners "SELECT \* FROM boston;"

The result is a runner.shp shapefile.

Save the result of trajectories that captured in July only, because July is the hottest season one year,

Example create shapefile that record the trajectories in July,

pgsql2shp -u postgres -h localhost -P 5424796 -f july\_trj runners "select \* from boston where SUBSTRING(startt, 6, 2)='07';"

Create shapefile for records that in July of 2015 from the database,

pgsql2shp -u postgres -h localhost -P 5424796 -f july\_trj2015 runners "select \* from boston where SUBSTRING(startt, 6, 2)='07' and left(startt, 4)='2015';"

Create shapefile for records that in July and August of 2015 from the database,

pgsql2shp -u postgres -h localhost -P 5424796 -f July\_aug\_trj2015 runners "select \* from boston where SUBSTRING(startt, 6, 2)='07' or SUBSTRING(startt, 6, 2)='08' and left(startt, 4)='2015';"