Spatial Data Science & Engineering

Assignment 2

Maximum points Possible – 10

The required task is to run spatial SQL queries on a PostGIS database.

Setting Up Working Environment:

- Install PostgreSQL version 14.5
- Add the tool PostGIS on PostgreSQL
- Install psycopg2 latest version
- Install Geopandas latest or a recent version
- Install Sqlalchemy
- Install GeoAlchemy2

Problem Description:

Given a Python project template and two data files – 1) a shape file containing Chicago Airbnb information 2) a CSV file containing latitude and longitude representing locations of various abandoned vehicles in Chicago. Assume that SRID is 4326 for both files. The Python project template creates a database on PostGIS named 'space_assignment2' for you. You will find a Python file Assignment2.py with two empty methods: 'load_shape_data' and 'explore_spatial_sql'.

The first method *load_shape_data* has 2 parameters:

engine: a sqlalchemy connection engine for the database

input_path: path to the shape file containing Chicago Airbnb information

The second method *explore_spatial_sql* has 10 parameters:

connection: psycopg2 connection object for the database

input_path: path to the CSV file containing abandoned vehicle locations

output_path1, output_path2, ….., output_path8

You need to complete above two methods with the below tasks:

Tasks:

1. In the method ‘load_shape_data’, load the given Airbnb shape file from input_path into a table named ‘shape_data’. The table has not been created for you. The schema of the table should be similar to the schema of the shape file. The name of the geometry type column should be ‘geometry’.

The rest of the tasks should be completed in the method ‘explore_spatial_sql’.

2. Load the latitudes and longitudes of the CSV file from path input_path into a table named ‘point_data’. The table has not been created for you. The table should have three columns
    1. Name – latitude, type – REAL
    2. Name – longitude, type – REAL
    3. Name – geom, type – geometry.

    The column geom should store the corresponding geometry version of latitude and longitude.

3. For each geometry in table shape_data, find the number of points from the table point_data that are inside the corresponding geometry. Boundary points are not considered. Write the number of points in the ascending order in output_path1.
4. For each geometry in table shape_data, find the number of adjacent geometries. Two geometries are adjacent if they share at least 1 point together. Write the numbers in the ascending order in output_path2.
5. Find the Hausdorff distance between each geometry in the table shape_data and the point having lat = 41.84469250346108 and lon = -87.60914087617012. Assume that SRID is 4326 for the point. Write the distances in the ascending order in output_path3.
6. Find the maximum Distance between each geometry in the table shape_data and the point having lat = 41.84469250346108 and lon = -87.60914087617012. Assume that SRID is 4326 for the point. Write the distances in the ascending order in output_path4.
7. Find the length of the boundary of the convex hull of each geometry in the table shape_data. Write the perimeters in the ascending order in output_path5.
8. Find the area covered by the outer boundary of each geometry in the table shape_data. Exclude a geometry if it is not a polygon (Exclude multipolygons also). Write the area values in the ascending order in output_path6.
9. Find the point from the minimum bounding rectangle of each geometry in the table shape_data that is nearest_point from the point having lat = 41.84469250346108 and lon = -87.60914087617012. Assume that SRID is 4326 for the point. Write the text representations of the points in the ascending order in output_path7.
10. Find the area of the shared portion between each geometry in the table shape_data and the following polygon geometry. Assume that SRID is 4326 for the polygon. Write the area values in the ascending order in output_path8.

‘POLYGON((-87.69227959522789 41.85766547551493,-87.69227959522789 41.88908028505862,-87.63450859376373 41.88908028505862,-87.63450859376373 41.85766547551493,-87.69227959522789 41.85766547551493))’

Output Format:

Sample outputs are provided for queries 3-10. Follow the sample output files.

Grading Criteria:

There are 10 queries/tasks. Each query is responsible for 1 point. First two queries will be graded based on whether the table schemas are correct or not. Last 8 queries will be graded based on output correctness and usage of correct ST functions in your queries. You question should be answered through SQL queries. You can use Python coding to write the SQL query outputs into files. A sample method is given in Assignment2.py file for writing query outputs into files. You can either call that method or use alternative writing strategy. Don’t import anything from Tester.py into Assignemnt2.py. It will generate module not found error, and you will not get any points.

Submission Instructions:

Submit only Assignment2.py file. Tester.py file is for your testing only. Run Tester.py to test the output.