Lab 02. Spatial Data

Please execute the SQL script provided below to create multiple geometric features required for completing Lab02. After running the SQL script, use the Geometry Viewer to verify the generated geometries. The resulting geometries should resemble those depicted in Figure 1. Ensure to replace ch03 with your designated schema name before executing the script.

After executing the SQL queries using the specified functions for each Task, provide the following outputs:

* A screenshot of the Geometry Viewer displaying the results.
* SQL statements used to retrieve the geometry properties.

CREATE TABLE ch03.boston\_geometry (name varchar, geom geometry);

INSERT INTO ch03.boston\_geometry VALUES

('Point', ST\_GeomFromText('POINT( -71.1052 42.3508)', 4326)),

('Linestring', ST\_GeomFromText('LINESTRING(

-71.1173 42.3513 ,

-71.1033 42.3496 ,

-71.0971 42.3490 ,

-71.0887 42.3491

)', 4326)),

('Polygon', ST\_GeomFromText('POLYGON((

-71.0724 42.3554,

-71.0632 42.3577,

-71.0622 42.3566,

-71.0646 42.3525,

-71.0706 42.3519,

-71.0724 42.3554

))', 4326)),

('PolygonWithHole', ST\_GeomFromText('POLYGON((

-71.1145 42.3473,

-71.1108 42.3468,

-71.1105 42.3454,

-71.1136 42.3446,

-71.1145 42.3473

), (

-71.1120 42.3466,

-71.1113 42.3461,

-71.1117 42.3459,

-71.1121 42.3465,

-71.1120 42.3466

))', 4326));

A map of a city

Description automatically generated

**Figure 1 Geometry of boston\_geometry**

Task 1. Utilize the following PostGIS functions to retrieve key properties of the geometries stored in the database:

* ST\_GeometryType() – Determines the type of geometry (e.g., Point, LineString, Polygon).
* ST\_NDims() – Returns the number of dimensions of the geometry (2D or 3D).
* ST\_SRID() – Retrieves the Spatial Reference Identifier (SRID) of the geometry.

Task 1 Output:

A screenshot of a computer

Description automatically generated

SELECT name, ST\_GeometryType(geom), ST\_NDims(geom), ST\_SRID(geom)

FROM ch03.boston\_geometry;

Task 2. Write an SQL query to select all geometries from the database and convert them to Well-Known Text (WKT) format. (Hint: Use the function ST\_AsText() to convert geometries to WKT format.)

Task 2 Output:

A screenshot of a computer

Description automatically generated

SELECT name, ST\_AsText(geom) FROM ch03.boston\_geometry;

Task 3. Write an SQL query using the structure SELECT … FROM … WHERE … to retrieve the coordinates of the geometry from the boston\_geometry table where the name is 'Point'.

Task 3 Output:

A screenshot of a computer

Description automatically generated

SELECT ST\_AsText(geom)

FROM ch03.boston\_geometry

WHERE name = 'Point';

Task 4. Write an SQL query to retrieve the length of the Linestring geometry from the boston\_geometry table using the ST\_Length() function.

Task 4 Output:

A screenshot of a computer

Description automatically generated

SELECT ST\_Length(geom)

FROM ch03.boston\_geometry

WHERE name = 'Linestring';

Task 5. Write an SQL query to calculate the area of all polygon geometries in the boston\_geometry table where the name includes the character pattern 'Polygon%'.

Task 5 Output:

A screenshot of a computer

Description automatically generated

SELECT name, ST\_Area(geom)

FROM ch03.boston\_geometry

WHERE name LIKE 'Polygon%';