***SGD LAB EXP – 8***

**Name** : Aditi Chhajed; **Reg. No.** : 221081009

**Branch** : IT ; **Course Instructor** : Prof. Vedashree Awati

Name : Aditi Chhajed

Reg.No : 22108100

***Aim:***

Spatial Measurement Functions.

***Theory:***

***1. ST\_Area(geometry)***

* *Calculates the area of a polygonal geometry in the spatial reference system of the geometry.*
* *Example: `SELECT ST\_Area(geom) FROM parks WHERE name = 'Yellowstone National Park';`*

***2. ST\_Centroid(geometry)***

* *Returns the centroid (geometric center) of a geometry, which is the average location of all points in the geometry.*
* *Example: `SELECT ST\_Centroid(geom) FROM cities WHERE name = 'Los Angeles';`*

***3. ST\_Distance(geometry, geometry)***

* *Calculates the minimum distance between two geometries.*
* *Example: `SELECT ST\_Distance(geom1, geom2) FROM roads, landmarks WHERE roads.name = 'Route 66' AND landmarks.name = 'Grand Canyon';`*

***4. ST\_Distance\_Spheroid(geometry, geometry, spheroid)***

* *Computes the distance between two geometries on the surface of a spheroid (ellipsoid), which accounts for the Earth's curvature.*
* *Example: `SELECT ST\_Distance\_Spheroid(geom1, geom2, 'SPHEROID["WGS 84", 6378137, 298.257223563]') FROM cities WHERE name = 'New York' AND name = 'Los Angeles';`*

***5. ST\_Distance\_Sphere(geometry, geometry)***

* *Calculates the minimum distance between two geometries on the Earth's surface using a spherical model, considering the curvature of the Earth.*
* *Example: `SELECT ST\_Distance\_Sphere(geom1, geom2) FROM cities WHERE name = 'New York' AND name = 'Los Angeles';`*

***6. ST\_Length(geometry)***

* *Computes the total length of a linestring geometry.*
* *Example: `SELECT ST\_Length(geom) FROM roads WHERE name = 'Route 66';`*

***7. ST\_Length\_Spheroid(geometry, spheroid)***

* *Measures the length of a linestring geometry on a spheroid, accounting for the Earth's curvature.*
* *Example: `SELECT ST\_Length\_Spheroid(geom, 'SPHEROID["GRS 1980",6378137,298.257222101]') FROM roads WHERE name = 'Route 66';`*

***8. ST\_Length3D(geometry)***

* *Calculates the 3D length of a linestring geometry, considering the Z-dimension (height).*
* *Example: `SELECT ST\_Length3D(geom) FROM mountain\_trails WHERE name = 'Everest Base Camp Trail';`*

***9. ST\_Length3D\_Spheroid(geometry, spheroid)***

* *Computes the 3D length of a linestring geometry on the surface of a spheroid, incorporating both the geometry's Z-dimension and the Earth's curvature.*
* *Example: `SELECT ST\_Length3D\_Spheroid(geom, 'SPHEROID["GRS 1980",6378137,298.257223563]') FROM mountain\_trails WHERE name = 'Everest Base Camp Trail';`*

***10. ST\_Perimeter(geometry)***

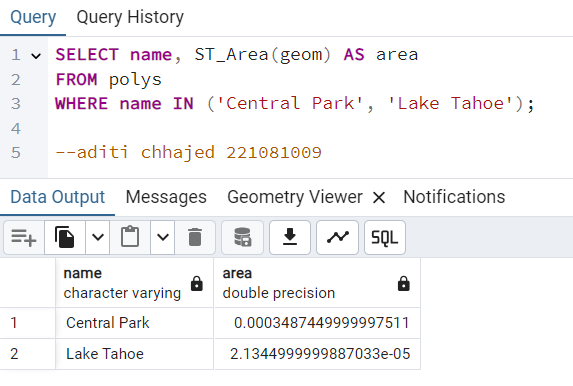
* *Calculates the perimeter of a polygonal geometry, which is the total length of the boundary.*
* *Example: `SELECT ST\_Perimeter(geom) FROM parks WHERE name = 'Grand Canyon';`*

***11. ST\_Perimeter3D(geometry)***

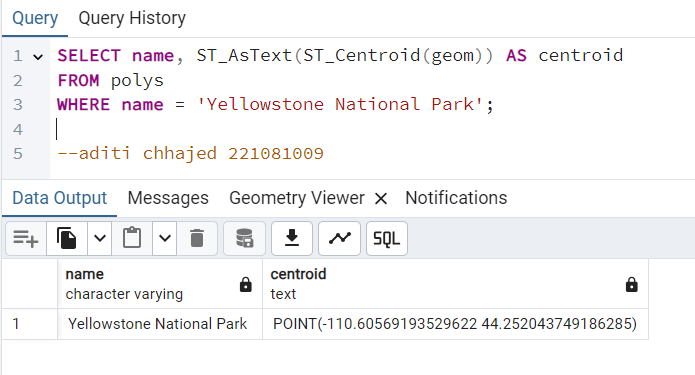
* *Computes the perimeter of a 3D polygon geometry, accounting for the Z-dimension.*
* *Example: `SELECT ST\_Perimeter3D(geom) FROM 3d\_parks WHERE name = 'Grand Canyon';`*

***Implementation:***

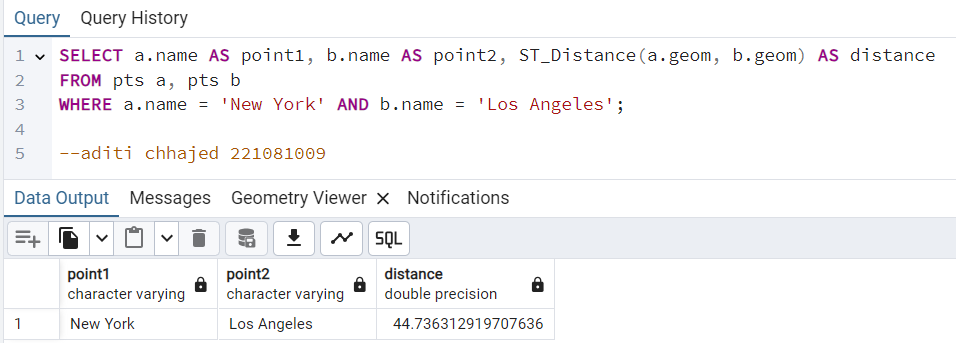
1. ***ST\_Area():***

******

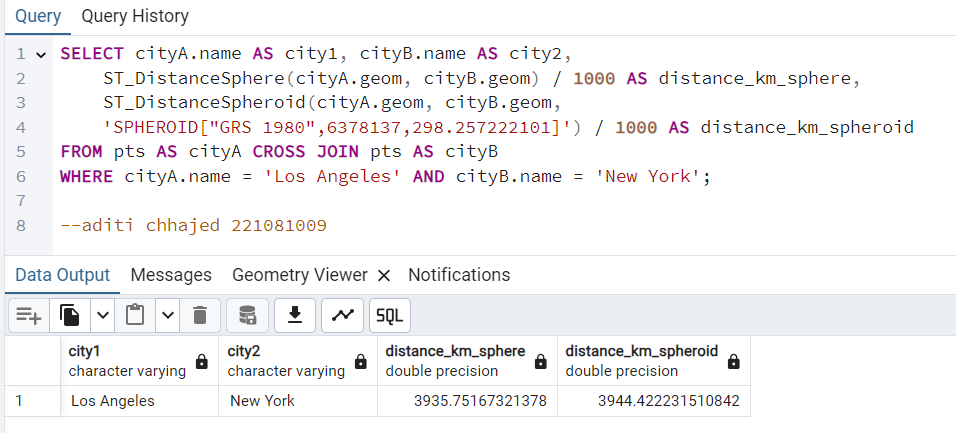
1. ***ST\_Centroid():***

******

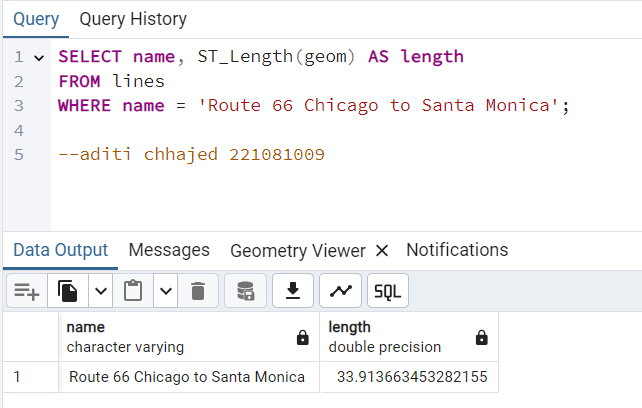
1. ***ST\_Distance():***

******

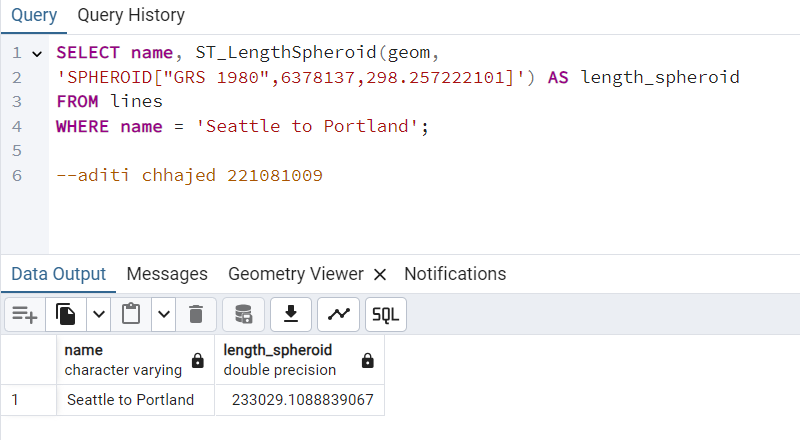
1. ***ST\_Distance\_Spheroid() and ST\_Distance\_Sphere():***

******

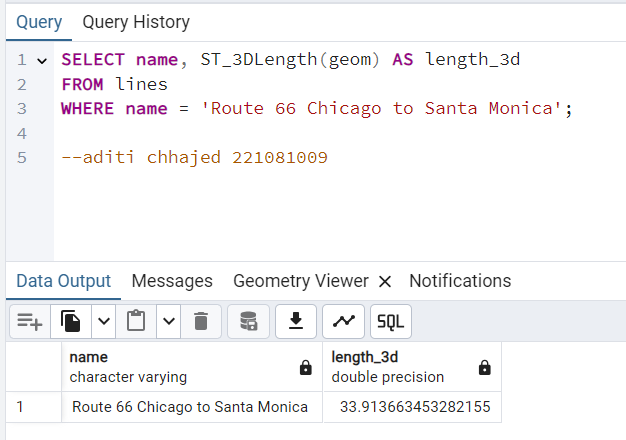
1. ***ST\_Length():***

******

1. ***ST\_LengthSpheroid():***

******

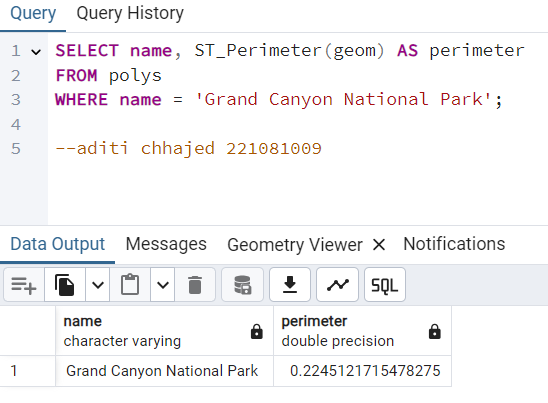
1. ***ST\_3DLength():***

******

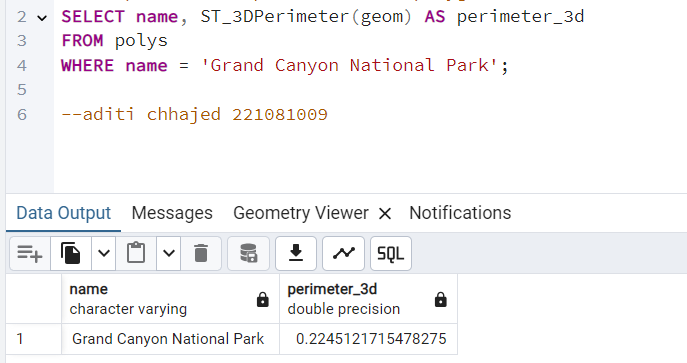
1. ***ST\_Length3D\_Spheroid():***

* *PostGIS offers several spatial functions for distance and length calculations, but not every theoretical function (like ST\_Length3D\_Spheroid()) is supported. Instead, ST\_DistanceSpheroid() and transformations to geography types are typically used for high-accuracy calculations involving Earth's curvature.*

1. ***ST\_Perimeter():***

******

1. ***ST\_3DPerimeter():***

******

***Conclusion:***

*In summary, using PostGIS spatial relationship functions allowed me to analyze and understand how different geographic features relate to one another.*

*By using functions like `ST\_Contains()` and `ST\_Within()`, I can determine whether a specific geometry lies entirely within another.*

*With functions like `ST\_Intersects()` or `ST\_Overlaps()`, I can identify shared spaces or overlapping areas.*

*These tools helped me explore boundaries, distances, adjacency, and coverage between geographic entities.*

*Whether I need to check if a point is inside a polygon, find features within a specific distance, or identify touching boundaries, these functions provide me with powerful ways to analyze and interact with spatial data for insightful results and meaningful spatial relationships.*