***SGD LAB EXP – 7***

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

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

Name : Aditi Chhajed

Reg.No : 22108100

***Aim:***

Spatial Relationship Functions.

***Theory:***

1. ***ST\_Contains()***

* *Checks if one geometry fully contains another geometry.*
* *Example: Imagine a park represented as a polygon and a bench within the park represented as a point. If you check whether the park polygon `ST\_Contains()` the bench point, it would return `TRUE` because the point is entirely inside the park boundary.*

***2. ST\_Within()***

* *The converse of `ST\_Contains()`. It tests whether a geometry is fully within another geometry.*
* *Example: Consider a house (point) inside a city boundary (polygon). If you query whether the house `ST\_Within()` the city, it would return `TRUE` since the house is entirely within the city's boundaries.*

***3. ST\_Covers() and ST\_CoveredBy()***

* *`ST\_Covers()` checks if one geometry contains all points of another, including if they touch boundaries. `ST\_CoveredBy()` checks if a geometry is covered by another.*
* *Example: If a walking trail (line) exactly aligns with a park boundary (polygon), `ST\_Contains()` may return `FALSE` (because it touches but doesn't lie strictly inside). However, `ST\_Covers()` would return `TRUE` since the park covers all points of the trail.*

***4. ST\_Intersects()***

* *Tests if two geometries share any space, whether by overlapping, touching, or containment.*
* *Example: Two roads crossing at an intersection would return `TRUE` if tested with `ST\_Intersects()` since they share common space where they cross.*

***5. ST\_Disjoint()***

* *Returns `TRUE` if two geometries share no space.*
* *Example: If a lake (polygon) and a mountain (polygon) do not touch or overlap, they are `ST\_Disjoint()`, meaning they share no space and are separate.*

***6. ST\_Overlaps()***

* *Determines if two geometries share some but not all points. They must have the same dimension and not be contained entirely within one another.*
* *Example: Two adjacent city districts with a common border but that do not fully contain each other would return `TRUE` for `ST\_Overlaps()`.*

***7. ST\_Touches()***

* *Returns `TRUE` if the geometries have at least one point in common but their interiors do not overlap.*
* *Example: Two countries sharing a border would return `TRUE` with `ST\_Touches()`, as they meet along their boundary without sharing interior space.*

***8. ST\_DWithin()***

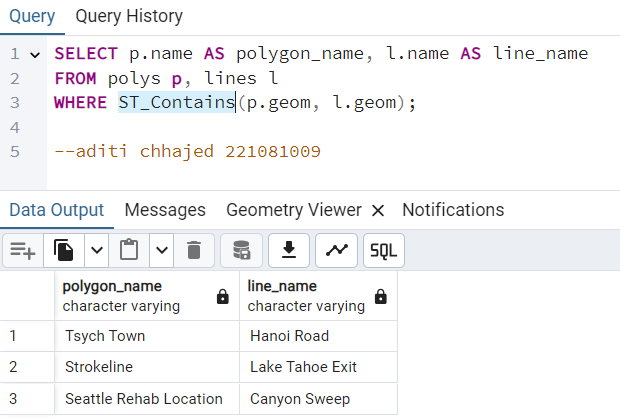
* *Tests if two geometries are within a specified distance of one another.*
* *Example: Checking if a school (point) is within 500 meters of a park (polygon) would return `TRUE` if the distance is less than or equal to 500 meters.*

***9. ST\_DFullyWithin()***

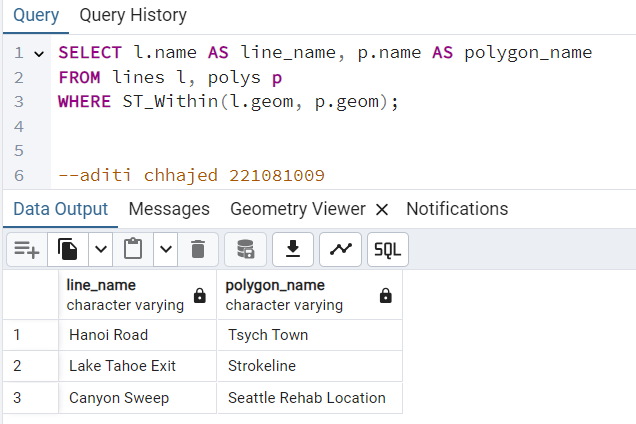
* *Similar to `ST\_DWithin()`, but every point of the first geometry must be within the distance of the second.*
* *Example: A river (line) must be entirely within 1 kilometer of a national park boundary (polygon) for `ST\_DFullyWithin()` to return `TRUE`. If any part of the river extends beyond this distance, it returns `FALSE`.*

***Implementation:***

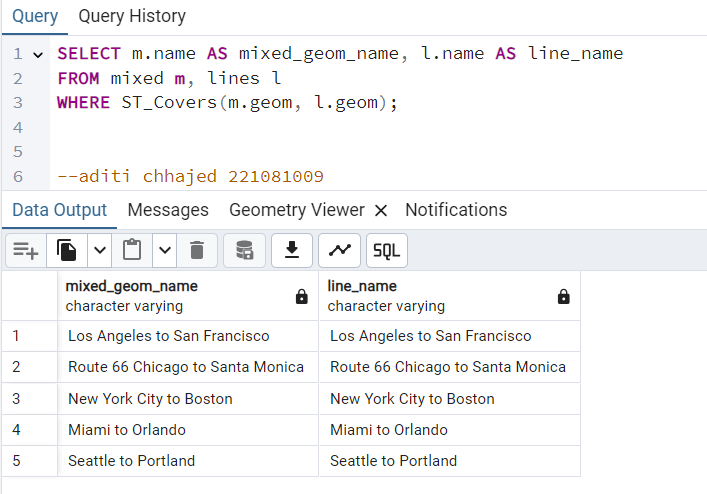
1. ***ST\_Contains:***

******

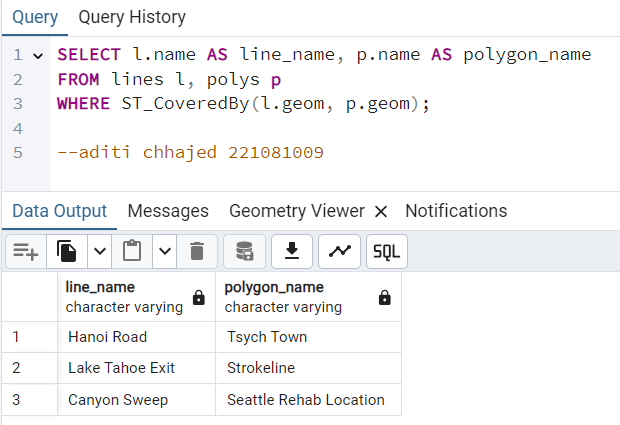
1. ***ST\_Within():***

******

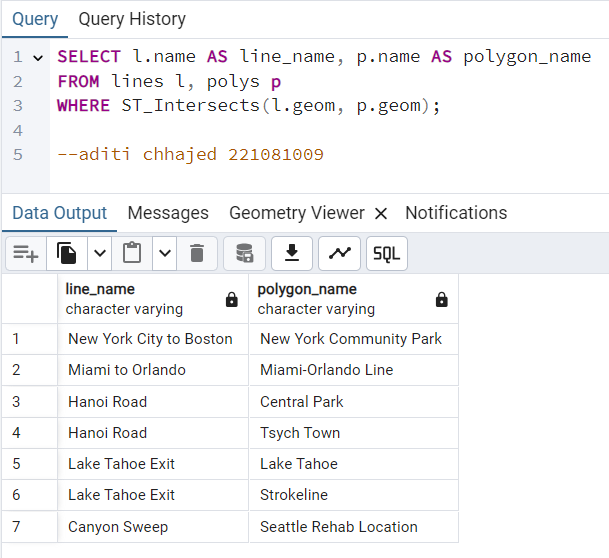
1. ***ST\_Covers():***

******

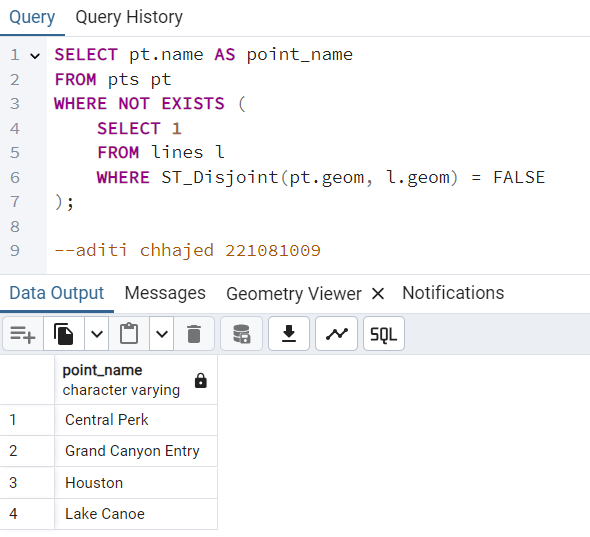
1. ***ST\_CoveredBy():***

******

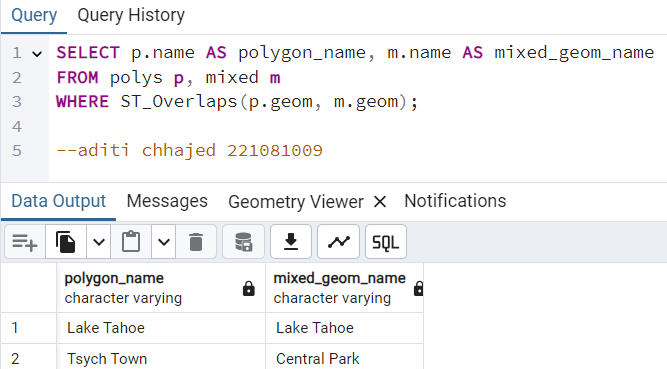
1. ***ST\_Intersects():***

******

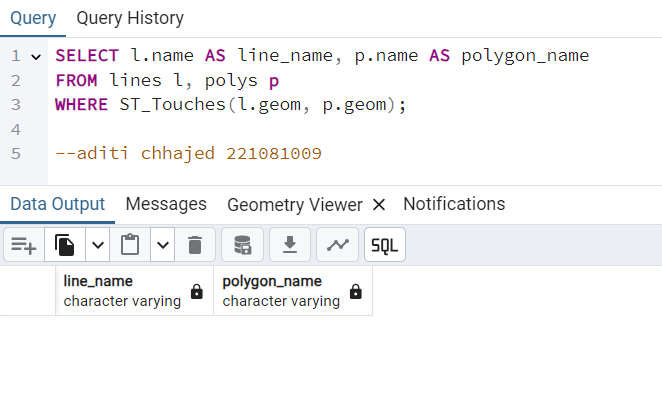
1. ***ST\_Disjoint():***

******

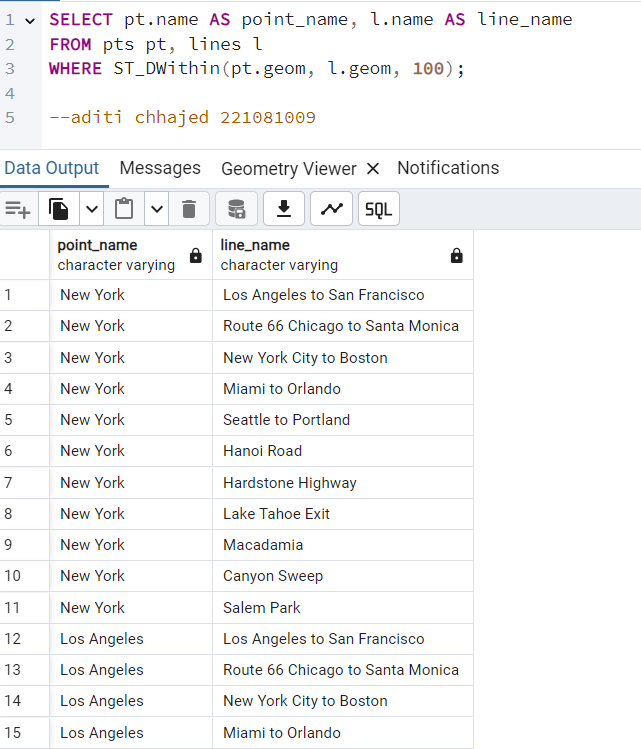
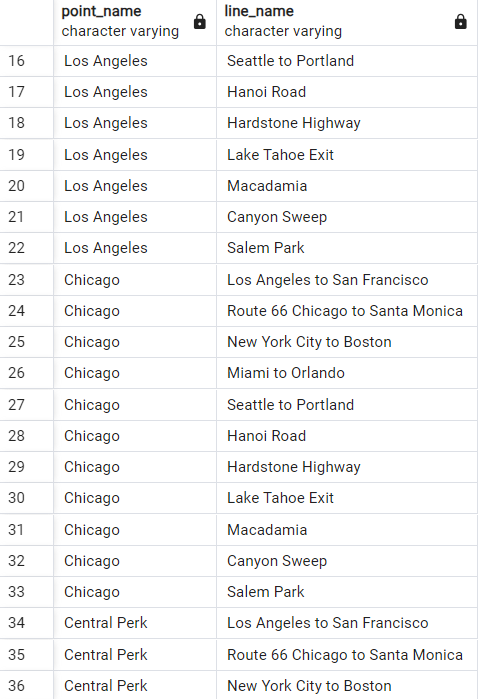
1. ***ST\_Overlaps():***

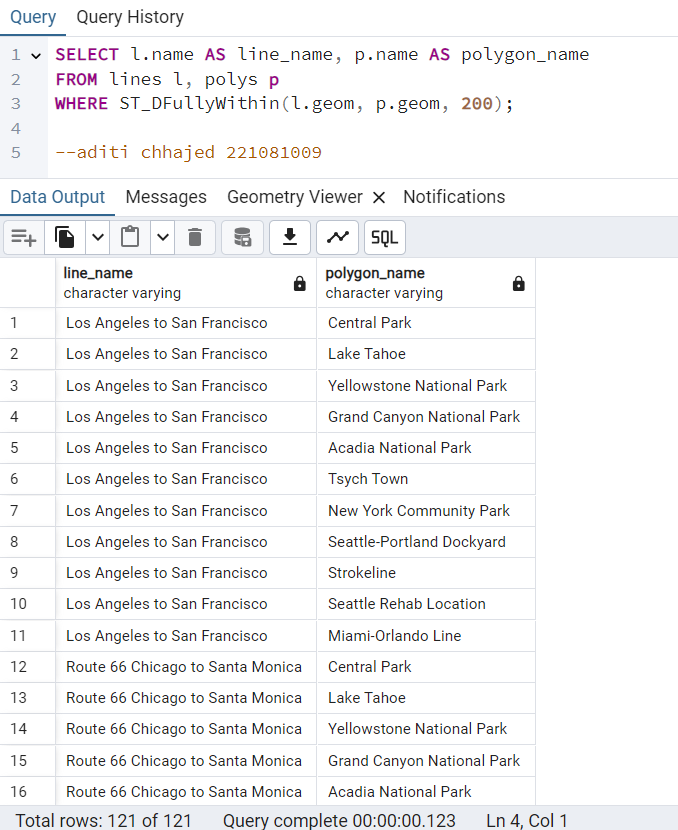
******

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

******

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

****** ******

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

***Putting all 121 results is out of the scope of this pdf.***

***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.*