**ADSL Assignment 12**

**Name** : Sharaneshwar Bharat Punjal

**PRN** : 23520011

**Batch** : T7

Problem Statement: Finding Things Close to Other Things.

Application in: location-based services on the web

**Algorithm**

1. **Install and Configure Neo4j Spatial Plugin:**
   * Download and install the Neo4j Spatial plugin from [GitHub.](https://github.com/neo4j-contrib/spatial) o Follow the instructions to integrate it into your Neo4j installation.
   * Verify the plugin by checking if spatial functions like distance() and point() are available.

**Generate and Add 10,000 Random Location Points:**

* Define random coordinates for each of the 10,000 location points (latitude, longitude).
* Create Place nodes in Neo4j with a property for each location's coordinates.

3. **Execute and Validate Queries:**

* Test the queries by running them in the Neo4j browser and verify that the results return the correct nearest locations.
* Ensure that the plugin’s geospatial functions are correctly interpreting the data and calculating distances between locations.

**Procedure**

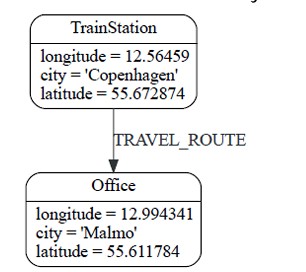
1. **Install Neo4j and Neo4j Spatial Plugin:**
   * Download and install Neo4j from the official website.
   * Download the Neo4j Spatial plugin from GitHub and configure it by placing the plugin in the plugins/ directory of Neo4j.
   * Restart Neo4j to load the plugin.
2. **Create Random Location Data:**
   * Use a scripting language like Python or JavaScript to generate 10,000 random geographic coordinates (latitude and longitude).
   * Write a script to add these points as nodes into Neo4j with a Place label and a location property for each point's coordinates.
3. **Write and Execute Cypher Queries:**
   * Use the Cypher queries mentioned earlier to find the nearest places to a

specified location or another place.

* + Run queries in the Neo4j browser or through a Neo4j client (e.g., using Python’s neo4j package) to fetch and display results.

1. **Optimize and Analyze:**
   * Test the scalability and performance of the queries, especially when handling a large number of locations.
   * Consider indexing location data for faster search performance.

1. Write CQL (Cypher Query Language) script to add randomly 10,000 location points as follows. Assume any data.

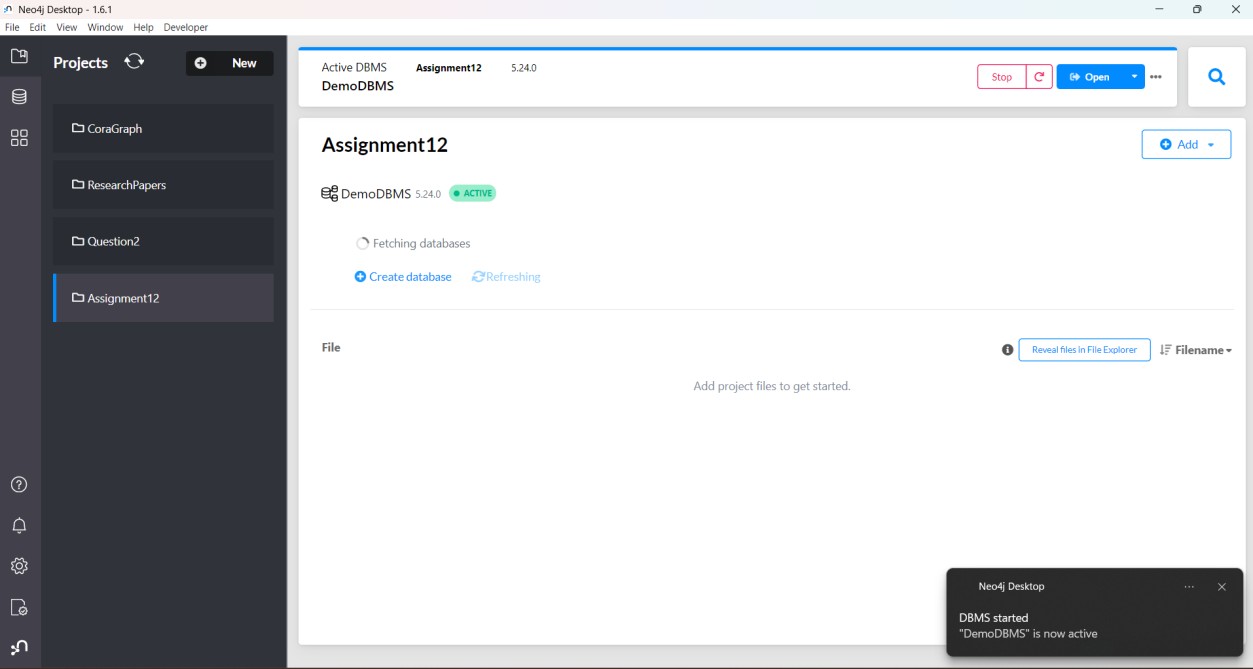


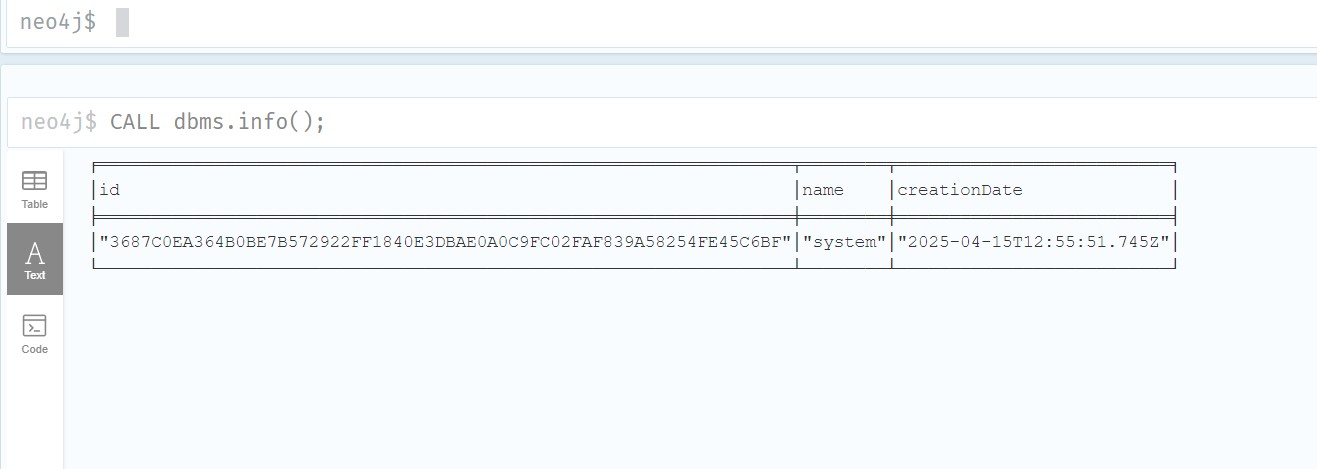
1. Use the point() , distance() function of Neo4j to answer the queries “which things close/nearest to which other things”.

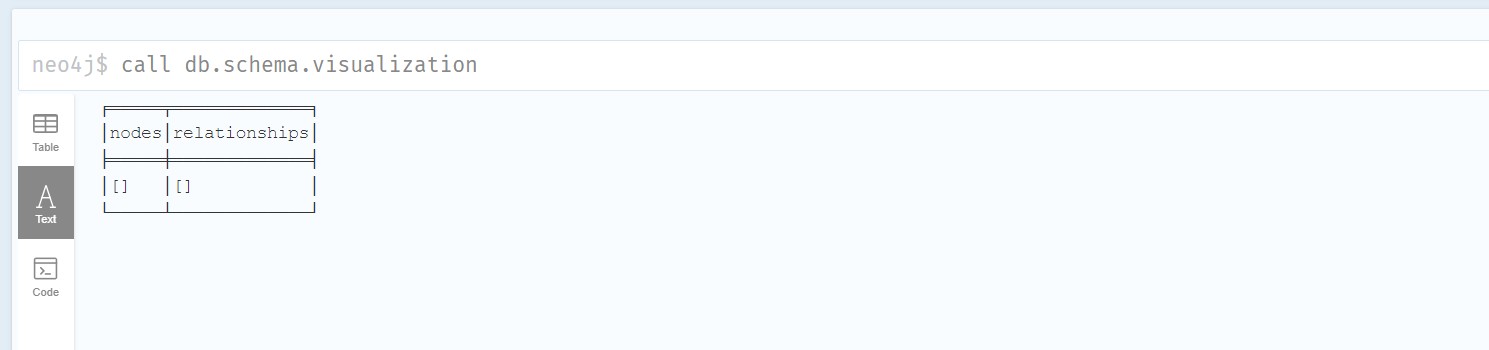
Demonstrate the result by firing different cypher queries (write CQL statement).

**Result:**

Creating a project and database in Neo4J



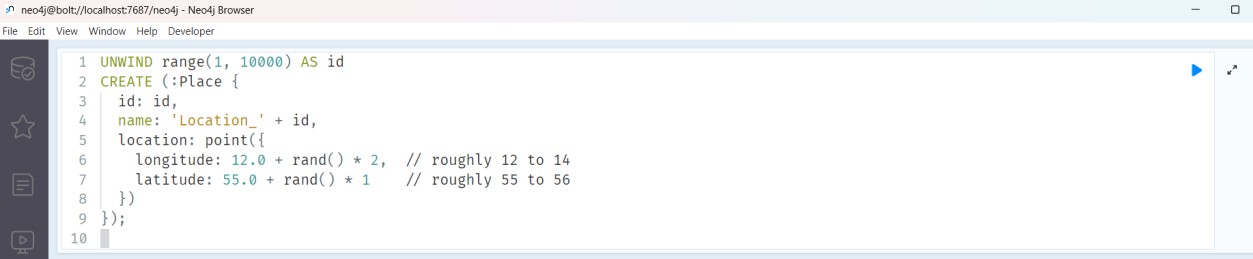




Creating Sample Data (TrainStation → Office)

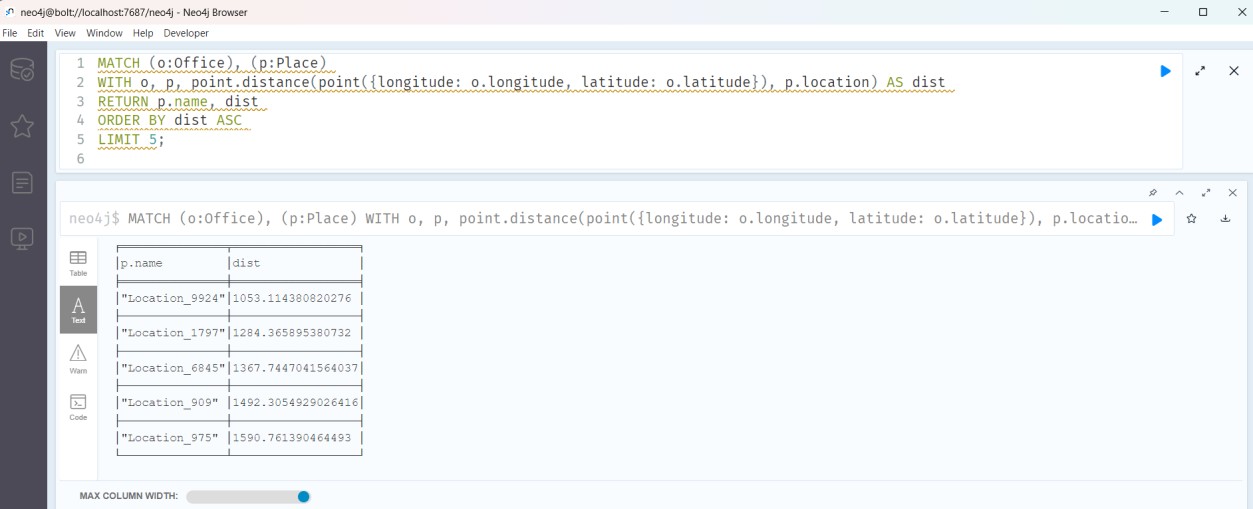


Generate 10,000 Random Location Nodes





Find Nearest Places Using distance()



Find all Places within 5km of the TrainStation



**Conclusion**

This assignment demonstrates the application of Neo4j and its Spatial plugin for performing location-based queries on large datasets. By utilizing the point() and distance() functions, the assignment successfully illustrates how to find the nearest objects in a graph, which can be useful in various real-world applications such as geospatial search services. The integration of Neo4j with geospatial data allows for efficient querying and analysis of proximity relationships, which is crucial in applications like location-based services, geographic information systems (GIS), and route optimization.