

Assignment No. 12

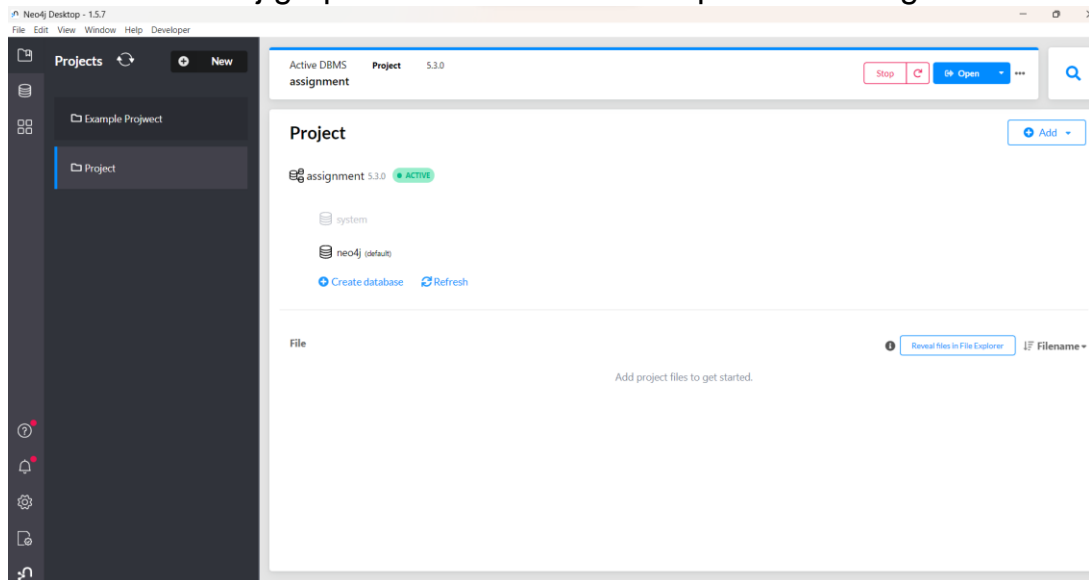
Spatial and Geographic Data

Geospatial is the natural domain for Graph Database **Use Neo4j and Neo4j Spatial**

Problem Statement : Finding Things Close to Other Things.
Application in : location-based services on the web

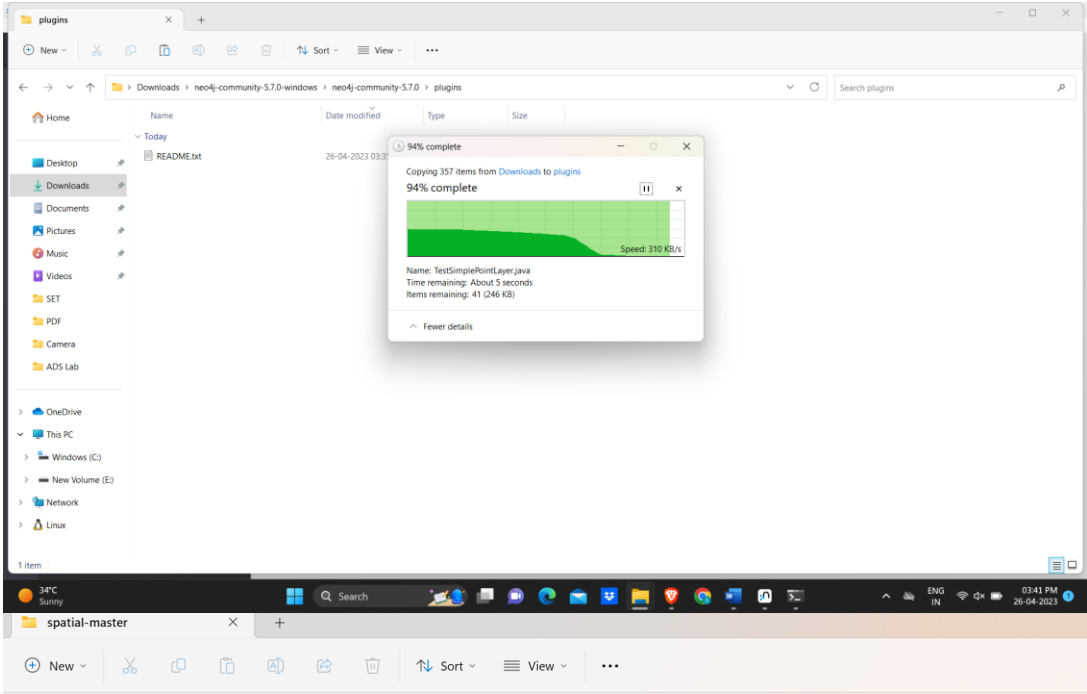
Task :

1. Use Neo4j graph database installed in previous assignments.



2. Install/configure Neo4jSpatial (<https://github.com/neo4jcontrib/spatial>) from GitHub. It is the Neo4j plug-in that facilitates geospatial operations on data stored in Neo4j.

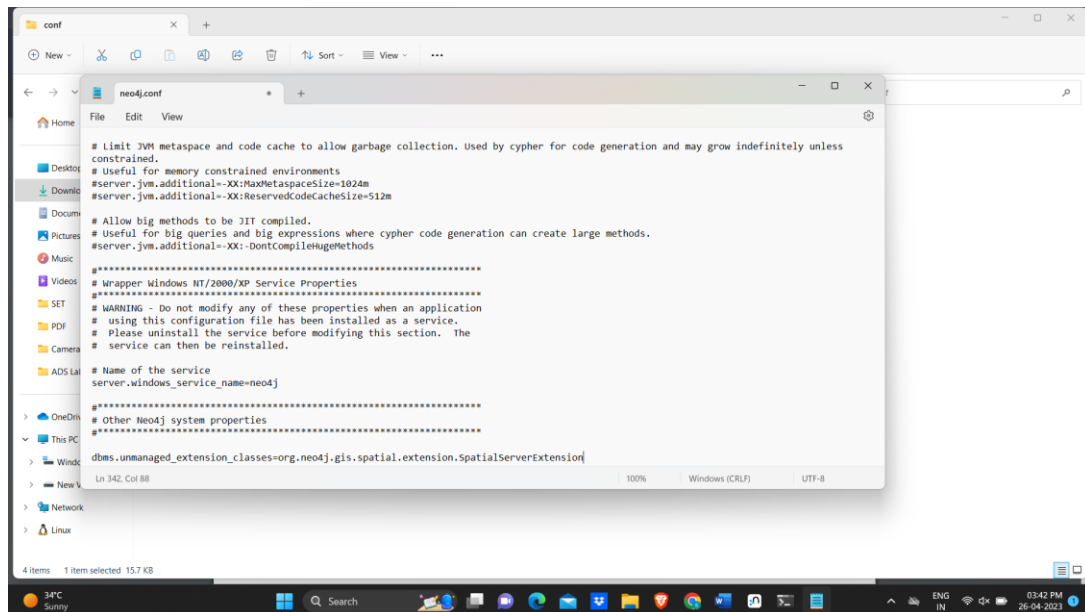
Name : Sharvari Yashwant Patil
PRN : 2020BTECS00077



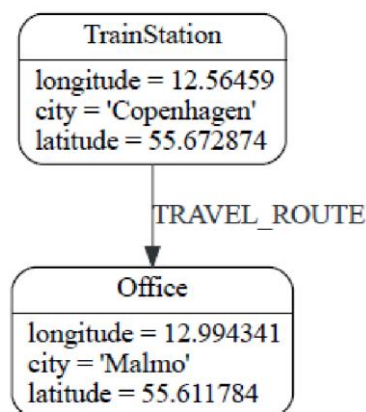
Downloads > spatial-master > spatial-master

	Name	Date modified	Type	Size
Today				
	two-street.osm	24-04-2023 20:35	OSM File	6 KB
	withinDistance.osm	24-04-2023 20:35	OSM File	640 KB
	map.osm	24-04-2023 20:35	OSM File	505 KB
	map2.osm	24-04-2023 20:35	OSM File	9,831 KB
	neo.sld.xml	24-04-2023 20:35	XML File	16 KB
	one-street.osm	24-04-2023 20:35	OSM File	2 KB
	osgi.bnd	24-04-2023 20:35	BND File	1 KB
	pom.xml	24-04-2023 20:35	XML File	32 KB
	sample.osm	24-04-2023 20:35	OSM File	2 KB
	.gitignore	24-04-2023 20:35	Git Ignore Source ...	1 KB
	.travis	24-04-2023 20:35	Yaml Source File	1 KB
	LICENSE	24-04-2023 20:35	Text Document	36 KB
	LICENSES	24-04-2023 20:35	Text Document	97 KB
	Makefile	24-04-2023 20:35	File	1 KB
	NOTICE	24-04-2023 20:35	Text Document	3 KB
	README	24-04-2023 20:35	Markdown Source ...	29 KB
	utils	24-04-2023 20:35	File folder	
	src	24-04-2023 20:35	File folder	
	shp	24-04-2023 20:35	File folder	
	sld	24-04-2023 20:35	File folder	

Name : Sharvari Yashwant Patil
PRN : 2020BTECS00077



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

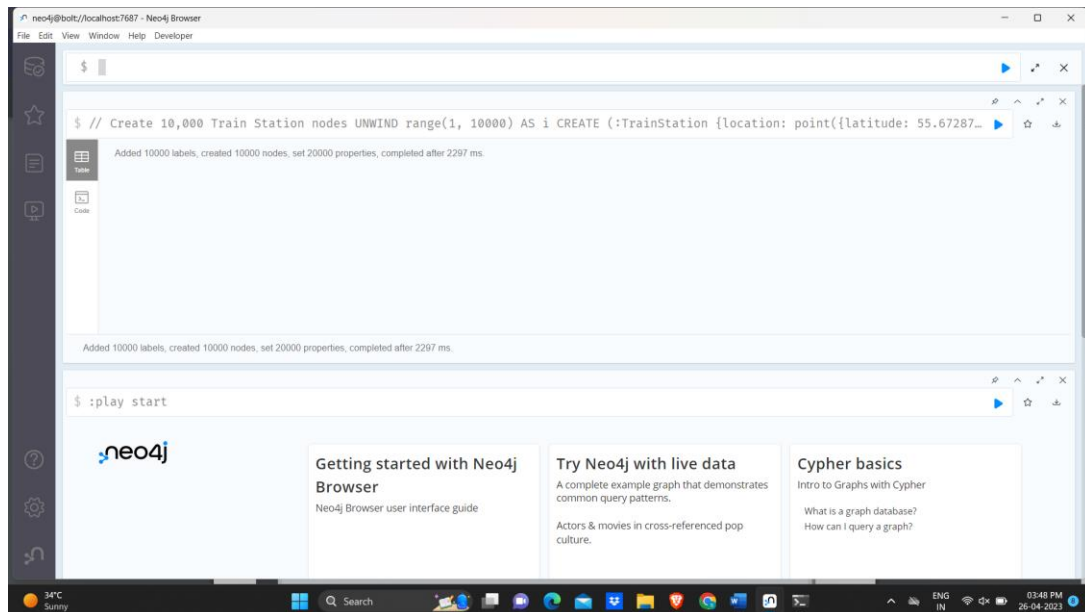


Cypher script :

// Create 10,000 Train Station nodes

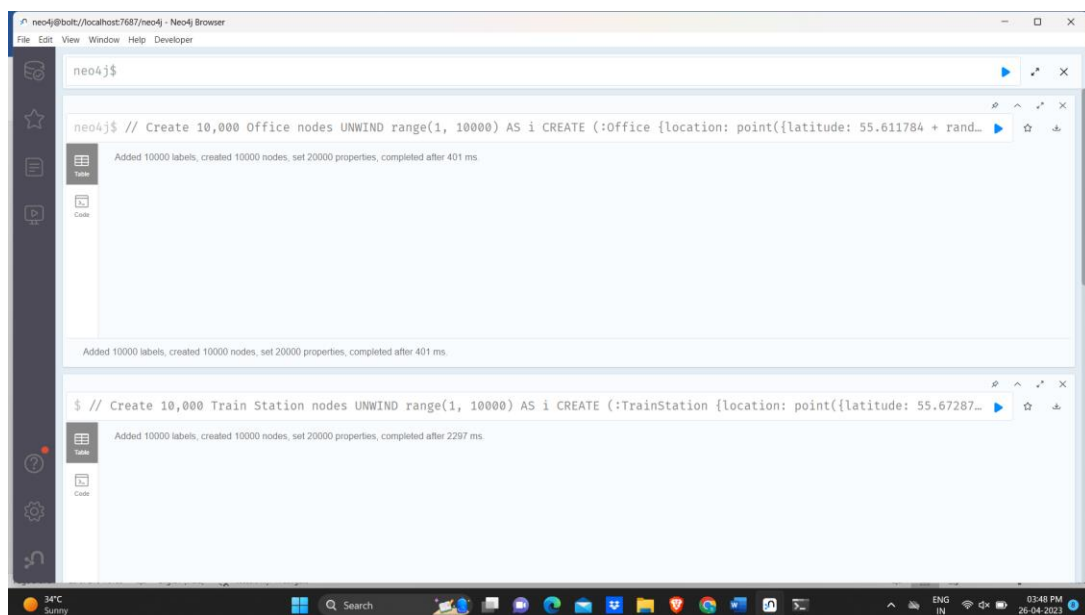
UNWIND range(1, 10000) AS i CREATE (:TrainStation {location:
point({latitude: 55.672874 + rand()*0.5, longitude: 12.56459 +
rand()*0.5}), city: 'Copenhagen'});

Name : Sharvari Yashwant Patil
PRN : 2020BTECS00077



// Create 10,000 Office nodes

UNWIND range(1, 10000) AS i CREATE (:Office {location:
point({latitude: 55.611784 + rand()*0.5, longitude: 12.994341 +
rand()*0.5}), city: 'Malmo'});



4. Use the point() , distance() function of Neo4j to answer the queries "***which things close/nearest to which other things***".
5. Demonstrate the result by firing different cypher queries (write CQL statement).

Name : Sharvari Yashwant Patil
PRN : 2020BTECS00077

1. Find the nearest train station to each office in Malmo:

The top screenshot shows the Neo4j Browser interface with the following Cypher query:

```
neo4j$ MATCH (o:Office {city: 'Malmo'}) MATCH (t: TrainStation {city: 'Copenhagen'}) RETURN o, t ORDER BY point.distance (o.lo...
```

The result is a table with columns 'o' and 't'. Each row contains a JSON object for an Office node and a JSON object for a TrainStation node, both with their respective coordinates.

The bottom screenshot shows the same data visualized as a graph. The Office nodes are represented by orange dots, and the TrainStation nodes are represented by blue dots. The graph shows the spatial distribution of these nodes. The right sidebar shows the 'Overview' tab with 'Node labels' and a count of 192 for Office and 14 for TrainStation. It also states 'Displaying 192 nodes, 0 relationships.'

2. Find the closest Office to each Train Station in Copenhagen:

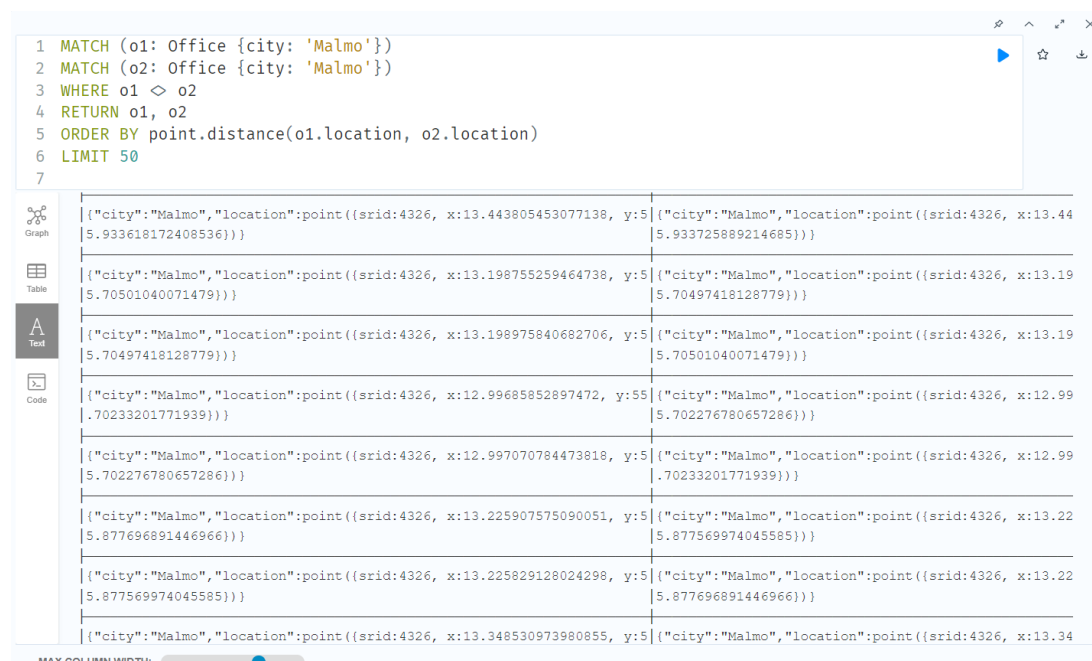
Name : Sharvari Yashwant Patil
PRN : 2020BTECS00077



```
1 MATCH (t:TrainStation {city: 'Copenhagen'})
2 MATCH (o:Office {city: 'Malmo'})
3 RETURN t, o
4 ORDER BY point.distance(t.location, o.location)
5 LIMIT 50
6
```

{ "city": "Copenhagen", "location": point({srid:4326, x:13.05098293028284, y:55.680240960836}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.05567995982611531}) }
{ "city": "Copenhagen", "location": point({srid:4326, x:13.00407679750133, y:55.72997446547808}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.00573000934254134}) }
{ "city": "Copenhagen", "location": point({srid:4326, x:12.99932807581581, y:55.7730405928264}) }	{ "city": "Malmo", "location": point({srid:4326, x:12.995773072922469495}) }
{ "city": "Copenhagen", "location": point({srid:4326, x:13.02250755022969, y:56.10846605291365}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.02610845069233796}) }
{ "city": "Copenhagen", "location": point({srid:4326, x:13.051812099870006, y:55.98789375077564}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.055988058250419165}) }
{ "city": "Copenhagen", "location": point({srid:4326, x:13.043606938194472, y:55.819393431766784}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.04581972370533159}) }
{ "city": "Copenhagen", "location": point({srid:4326, x:13.019508153624312, y:55.92729143271016}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.01592694784878002}) }

3. Find the closest Office to each other Office in Malmo.



```
1 MATCH (o1: Office {city: 'Malmo'})
2 MATCH (o2: Office {city: 'Malmo'})
3 WHERE o1 < o2
4 RETURN o1, o2
5 ORDER BY point.distance(o1.location, o2.location)
6 LIMIT 50
7
```

{ "city": "Malmo", "location": point({srid:4326, x:13.443805453077138, y:55.933618172408536}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.445933725889214685}) }
{ "city": "Malmo", "location": point({srid:4326, x:13.198755259464738, y:55.70501040071479}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.19570497418128779}) }
{ "city": "Malmo", "location": point({srid:4326, x:13.198975840682706, y:55.70497418128779}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.19570501040071479}) }
{ "city": "Malmo", "location": point({srid:4326, x:12.99685852897472, y:55.70233201771939}) }	{ "city": "Malmo", "location": point({srid:4326, x:12.995702276780657286}) }
{ "city": "Malmo", "location": point({srid:4326, x:12.997070784473818, y:55.702276780657286}) }	{ "city": "Malmo", "location": point({srid:4326, x:12.9970233201771939}) }
{ "city": "Malmo", "location": point({srid:4326, x:13.225907575090051, y:55.877696891446966}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.225877569974045585}) }
{ "city": "Malmo", "location": point({srid:4326, x:13.225829128024298, y:55.877696891446966}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.225877569974045585}) }
{ "city": "Malmo", "location": point({srid:4326, x:13.348530973980855, y:55.877696891446966}) }	{ "city": "Malmo", "location": point({srid:4326, x:13.348530973980855}) }

Note : Follow the submission guidelines.

Deadline : 23/04/2023

Dr. B. F. Momin
Course Coordinator