COMP- 8115-M50 Database systems

Quiz – 8

1. **[25 pts] Check out the Cypher queries in the companygraphdb.txt file and draw its graph diagram (directed graph with relationships). The companygraphdb.txt file can be found in the related module on canvas. You can draw it on a blank paper and take a photo with your smartphone and copy/paste it in your quiz answer form.**

   To use the cypher Queries which are provided in the companygraphdb.txt and draw its diagram.

   So, now I need to create a graph or draw a graph using the data provided. Before drawing a graph. Let's understand Neo4j

   It is a graph-based database management system which is developed by Neo4j corporation and is described by its developers as an ACID-complaint transactional database with native graph storage and processing.

   There are some good advantages and features of the Neo4j

   Features:

   - ✓ It is a SQL-like easy query language Neo4j CQL
   - ✓ It follows the property Graph Data Model
   - ✓ It supports Indexes by using Apache
   - ✓ It supports UNIQUE constraints
   - ✓ It contains a UI to execute CQL commands: Neo4j Data browser
   - ✓ It supports all the ACID Properties and many more!

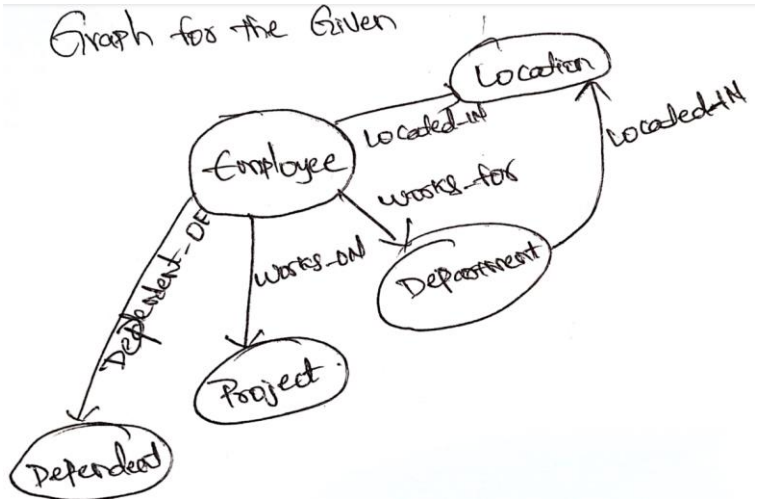
   Advantages:

   - ✓ It is very easy to represent connected data
   - ✓ It is very easy and faster to retrieve /traversal/navigation of more connected data
   - ✓ It represents semi-structured data very easily
   - ✓ It is a simple and powerful data model.

   So, This is enough more for the introduction lets get into Question

   The node labels in the given db are
   *(80), Department (6), Employee(40),Location(12),Project(11),Dependent(11)
   The relationship Types are
   *(112),Located_IN(13),Works_For(40),Dependent_of(11)

Attaching the Image of the graph below



IMG1: Graph for the given database.

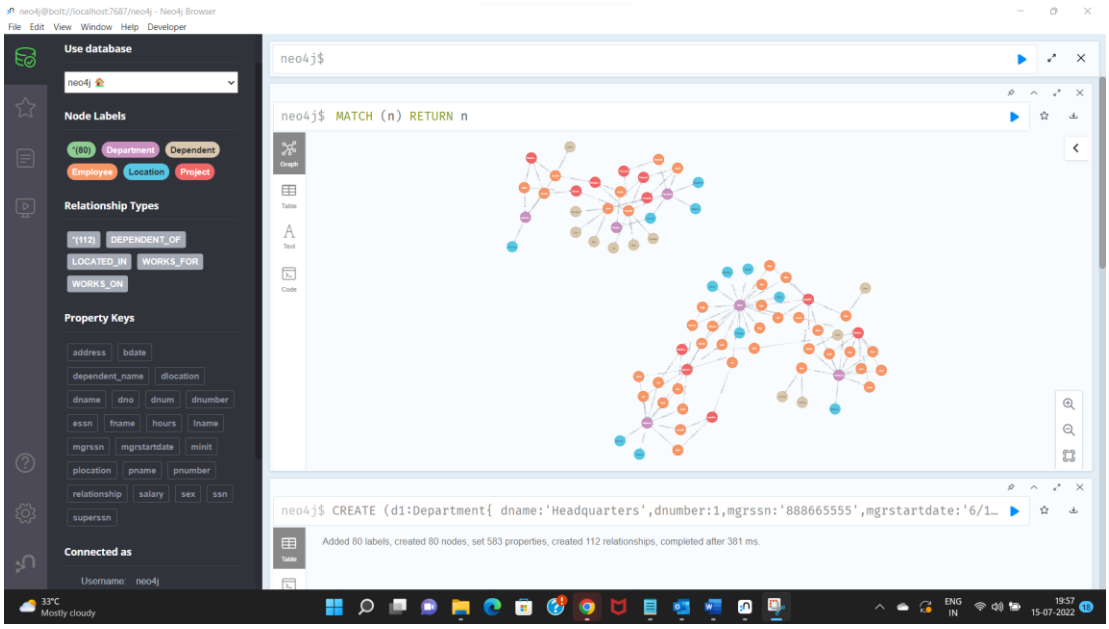
We just drew the graph for the nodes and attributes, but in the graph, we can dig deeper and draw a complete graph which is shown in the below question.

Explanation In own words:

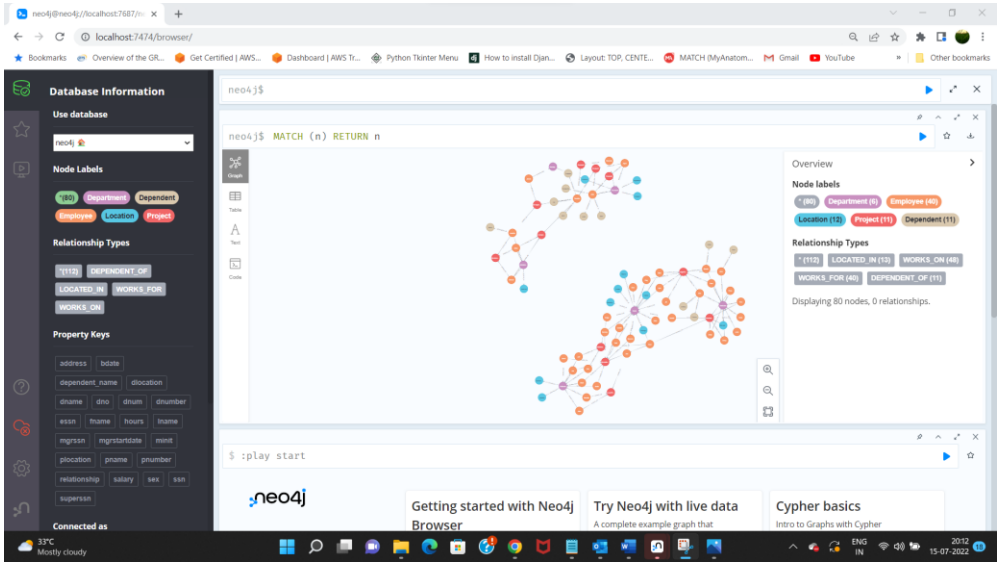
Every employee in the company works for a project and be dependent of some dependent. Employee does the project from some department of the company and the department is located in the some location.

2. **[25 pts] Using the queries in the companygraphdb.txt file, generate a new graph database called "companygraphdb" in Neo4j. After you generate the new database, include a screenshot of the web user interface (http://localhost:7474/browser/) showing that your company graph database is generated.**

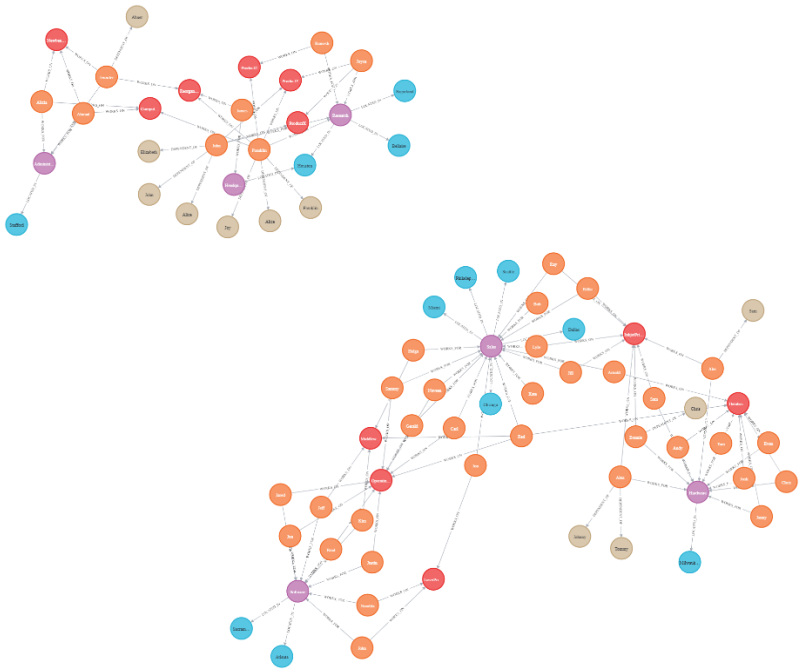
The question is to use the queries provided in the companygraphdb.txt file and create a new graph database



The below screenshot is using the http://localhost:7474/browser/



Also, attaching the PNG image for the better understanding of graph and better visual.



3. **[60 pts] Write Cypher queries for the following use cases (do not put any screenshots, only Cypher queries):**
   a. **[10 pts] Retrieve all the employees (nodes).**
      Match(e:Employee) return e

   b. **[10 pts] List the first name, last name, and salary of all employees.**
      Match (e:Employee) return e.fname,e.lname,e.salary

c. **[10 pts] Retrieve all employees (nodes) who work for the 'Sales' department.**

```
MATCH (e:Employee)-[WORKS_FOR]->(d:Department{dname:'Sales'})
RETURN distinct e,d
```

d. **[10 pts] Retrieve all employees (nodes) who work for the 'Sales', departments (nodes) and their relationship.**

```
MATCH(p:Employee)-[WORKS_FOR]->(d:Department{dname:'Sales'})-
[LOCATED_IN]->(l:Location)RETURN p,d,l
```

e. **[10 pts] List all the department names which are located in Houston.**

```
MATCH(d:Department)-[r:LOCATED_IN]->(l:Location)where
l.dlocation='Houston' RETURN distinct d.dname
```

f. **[Bonus - 5 pts] List all the first names and last names of employees, the project names that they have been working on, and the amount of time they spent on those projects.**

```
Match(e:Employee)-[r:WORKS_ON]->(p:Project)return
e.ssn,e.fname,e.lname,p.pname,r.hours
```

g. **[Bonus – 5 pts] Get the total number of projects that each employee has been working on.**

```
MATCH (e:Employee)-[r:WORKS_ON]->(p:Project) RETURN e.fname, e.ssn
,count(distinct p.pnumber)
```