

## **Assignment 6**

**Himanshu Goyal(hgoyal)**

## 1) Database:

```
CREATE (Dev:Department {title:'IT-Dev', departmentId:1, Description:'Development'})
```

```
CREATE (Himanshu:Employee {name:'Himanshu Goyal', Role:'Senior Dev'})
```

```
CREATE (Sonal:Employee {name:'Sonal Agarwal', Role:'Dev-1'})
```

```
CREATE (Himanshu)-[:WORK_IN {role:["Senior Dev"]}]>(Dev),  
(Sonal)-[:WORK_IN {role:["Dev-1"]}]>(Dev)
```

```
CREATE (Test:Department {title:'IT-Test', departmentId:2, Description:'Testing'})
```

```
CREATE (Priyanka:Employee {name:'Priyanka Goyal', Role:'Senior Test'})
```

```
CREATE (Mona:Employee {name:'Mona Goyal', Role:'Test-1'})
```

```
CREATE (Priyanka)-[:WORK_IN {role:["Senior Test"]}]>(Test),  
(Mona)-[:WORK_IN {role:["Test-1"]}]>(Test)
```

```
CREATE (Support:Department {title:'IT-Support', departmentId:3, Description:'Operation  
Support'})
```

```
CREATE (Rituraj:Employee {name:'Rituraj Singh', Role:'Senior Support'})
```

```
CREATE (Joshita:Employee {name:'Josita Singh', Role:'Support-1'})
```

```
CREATE (Rituraj)-[:WORK_IN {role:["Senior Support"]}]>(Support),  
(Joshita)-[:WORK_IN {role:["Support-1"]}]>(Support),  
(Himanshu)-[:WORK_IN {role:["Senior Support"]}]>(Support)
```

```
CREATE (HR:Department {title:'IT-HR', departmentId:4, Description:'Human Resource'})
```

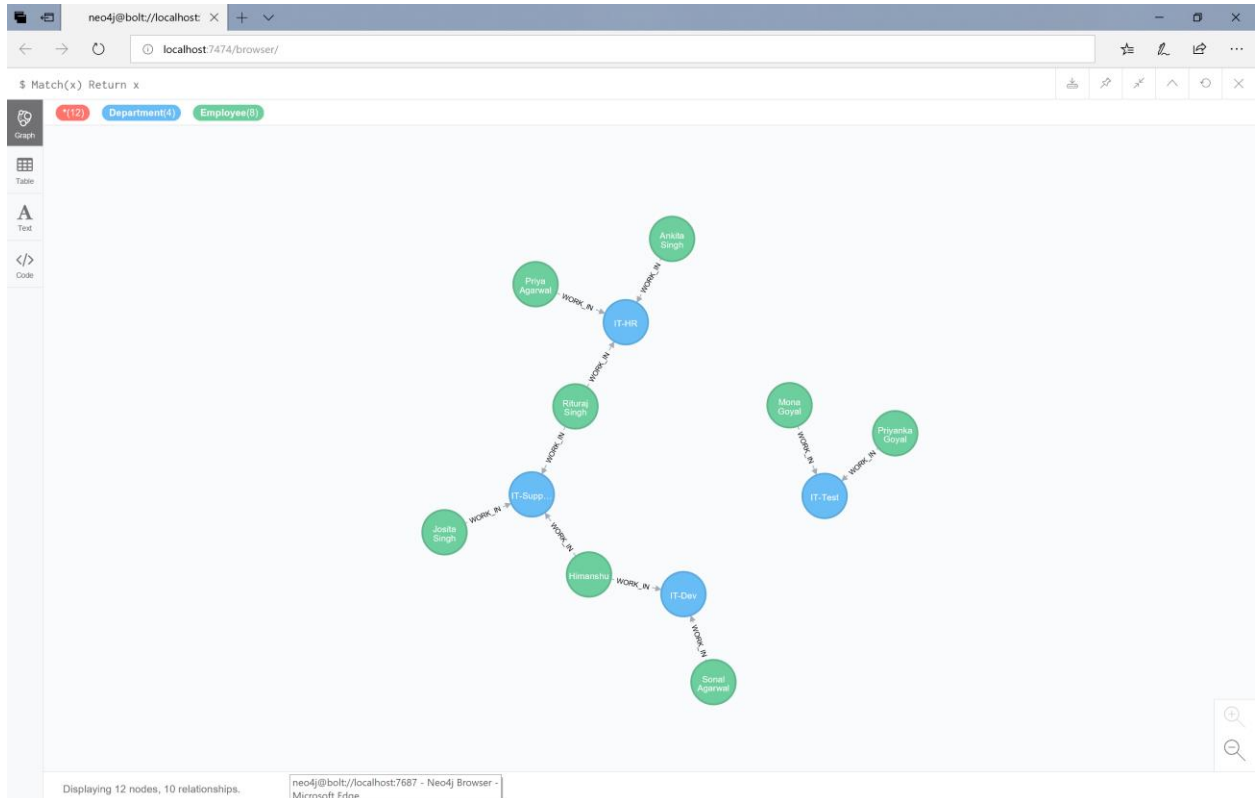
```
CREATE (Ankita:Employee {name:'Ankita Singh', Role:'Senior HR'})
```

```
CREATE (Priya:Employee {name:'Priya Agarwal', Role:'HR-1'})
```

```
CREATE (Ankita)-[:WORK_IN {role:["Senior HR"]}]->(HR),
(Priya)-[:WORK_IN {role:["HR-1"]}]->(HR),
(Rituraj)-[:WORK_IN {role:["Lead HR"]}]->(HR)
```

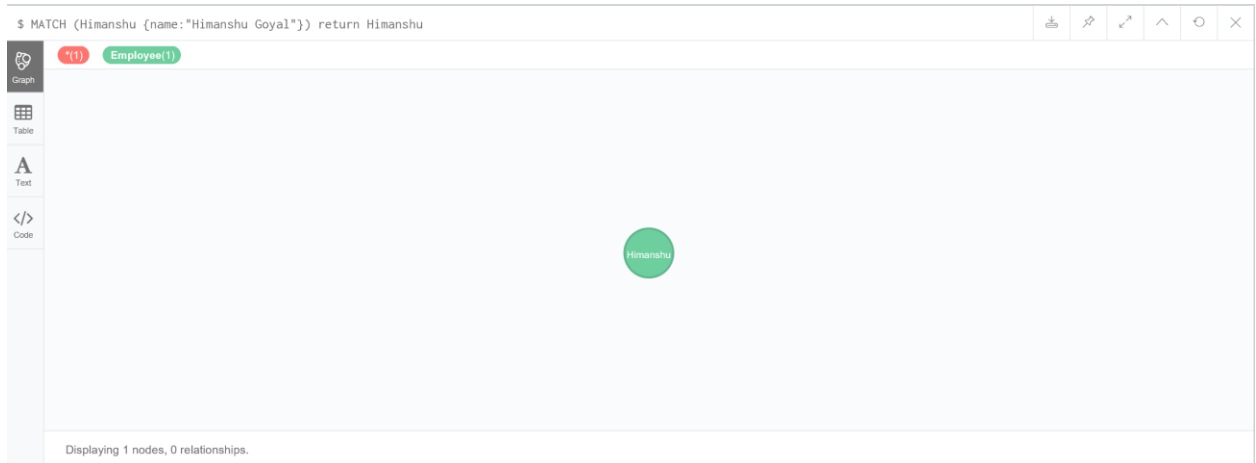
## 2) Return All the nodes.

Match(x) Return x



## 3) Return a single node where employee name matches.

MATCH (Himanshu {name:"Himanshu Goyal"}) return Himanshu



4) **Return all the Department and order by department ID.**

`MATCH(Department:Department) return Department.Description  
order by Department.departmentId`

The screenshot shows a Cypher query interface with the query: `$ MATCH(Department:Department) return Department.Description order by Department.departmentId`. The interface includes a sidebar with icons for Table, Text, and Code. The main area displays a table with the following data:

Department.Description
"Development"
"Testing"
"Operation Support"
"Human Resource"

At the bottom, it states 'Started streaming 4 records after 1 ms and completed after 25 ms.'

5) **Update role for an employee.**

`Match (Rituraj:Employee) where Rituraj.name = "Rituraj Singh"  
set Rituraj.role = "IT Head" return Rituraj.name as  
name,Rituraj.role as Role`

The screenshot shows a Cypher query interface with the query: `$ Match (Rituraj:Employee) where Rituraj.name = "Rituraj Singh" set Rituraj.role = "IT Head" return Rituraj.name as name,Rituraj.role as Role`. The interface includes a sidebar with icons for Table, Text, and Code. The main area displays a table with the following data:

name	Role
"Rituraj Singh"	"IT Head"

At the bottom, it states 'Set 1 property, started streaming 1 records after 22 ms and completed after 24 ms.'

6) **Add Salary property for all the employees.**

`MATCH (Himanshu:Employee) SET Himanshu.Salary = 80000 RETURN Himanshu`

`MATCH (Sonal:Employee) SET Sonal.Salary = 80000 RETURN Sonal`

`MATCH (Priyanka:Employee) SET Priyanka.Salary = 80000 RETURN Priyanka`

MATCH (Mona:Employee) SET Mona.Salary = 80000 RETURN Mona  
 MATCH (Rituraj:Employee) SET Rituraj.Salary = 80000 RETURN Rituraj  
 MATCH (Joshita:Employee) SET Joshita.Salary = 80000 RETURN Joshita  
 MATCH (Ankita:Employee) SET Ankita.Salary = 80000 RETURN Ankita  
 MATCH (Priya:Employee) SET Priya.Salary = 80000 RETURN Priya

\$ MATCH (Sonal:Employee) SET Sonal.Salary = 80000 RETURN Sonal

The screenshot shows the Neo4j query interface with the query: `$ MATCH (Sonal:Employee) SET Sonal.Salary = 80000 RETURN Sonal`. The result is displayed in a table view with 3 rows. The first row shows Sonal as a Senior Dev with name Himanshu Goyal and salary 80000. The second row shows Sonal as a Dev-1 with name Sonal Agarwal and salary 80000. The third row shows Sonal as a Senior Test with an empty name and salary field.

id	name	role	salary
1	Himanshu Goyal	Senior Dev	80000
2	Sonal Agarwal	Dev-1	80000
3		Senior Test	

Set 8 properties, started streaming 8 records after 3 ms and completed after 3 ms.

## 7) Delete Salary property for all the employees.

MATCH (Himanshu:Employee) REMOVE Himanshu.Salary

\$ Match(x) Return x

The screenshot shows the Neo4j query interface with the query: `$ Match(x) Return x`. The result is displayed in a table view with 3 rows. The first row shows a department with title IT-Dev, description Development, and departmentId 1. The second row shows Sonal as a Senior Dev with name Himanshu Goyal. The third row shows Sonal as a Dev-1 with name Sonal Agarwal.

id	name	role	salary
1			
2	Himanshu Goyal	Senior Dev	
3	Sonal Agarwal	Dev-1	

Started streaming 12 records in less than 1 ms and completed after 1 ms.

## 8) Return 2 departments only.

MATCH(Department:Department) return Department.Description order by  
 Department.departmentId limit 2

```
$ MATCH(Department:Department) return Department.Description order by Department.departmentId limit 2
```

Department.Description
"Development"
"Testing"

Started streaming 2 records after 3 ms and completed after 8 ms.

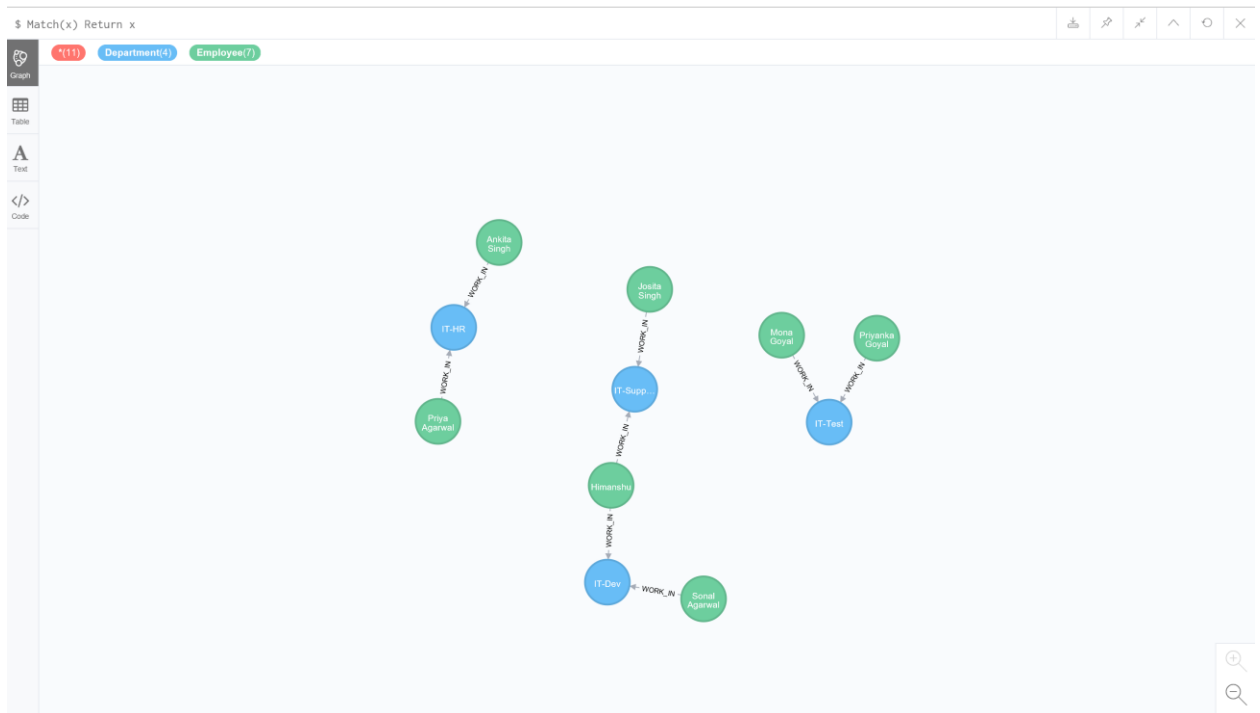
## 9) Delete Employee Rituraj Singh and all it's link

MATCH (Rituraj:Employee) where Rituraj.name="Rituraj Singh" Optional MATCH (Rituraj)-[r]-()  
delete Rituraj,r;

```
$ MATCH (Rituraj:Employee) where Rituraj.name="Rituraj Singh" Optional MATCH (Rituraj)-[r]-() delete Rituraj,r;
```

Deleted 1 node, deleted 2 relationships, completed after 21 ms.
---

Deleted 1 node, deleted 2 relationships, completed after 21 ms.



**10) Remove the testing Department and all its link.**

MATCH (Test:Department) where Test.title="IT-Test" Optional MATCH (Test)-[r]-() Delete Test,r

The image displays a Cypher query execution interface. At the top, a graph visualization shows a network of nodes and relationships. Blue nodes represent departments: IT-HR, IT-Supp., and IT-Dev. Green nodes represent employees: Priya Agarwal, Ankita Singh, Jyoti Singh, Himanshu, Sonal Agarwal, and Pooja Goyal. Relationships are labeled 'WORKS\_ON'. The graph shows IT-HR connected to Priya Agarwal and Ankita Singh, IT-Supp. connected to Jyoti Singh and Himanshu, and IT-Dev connected to Himanshu and Sonal Agarwal. Pooja Goyal is an isolated node.

Below the graph, the Cypher query is entered in the console:

```
$ MATCH (Test:Department) where Test.title="IT-Test" Optional MATCH (Test)-[r]-() Delete Test,r
```

The console output shows the result of the query execution:

```
Deleted 1 node, deleted 2 relationships, completed after 3 ms.
```

The interface includes a sidebar with 'Table' and 'Code' views, and a bottom status bar indicating the execution details.