Week 1 – Introduction to Graph database(NOSQL)

Problem Statement	Use NEO4j and create a sample graph database and perform the following operations on it 1. Create node with varying fields. 2. Add properties to node 3. Add relationships between the nodes . 4. Update an attribute value of the node 5. Retrieve and delete nodes, relationship

1.Create node and relationships

The CREATE clause is used to create nodes and relationships.

1.Create a single node :

Syntax: create (n)   // create a single node without label

2. Create multiple nodes

Syntax: create(n),(m)

3. Create a node with a label

Syntax: create(n:lable name)
Ex: CREATE (n:Person)

Use match(n) return(n) to view the nodes

4. Create a node with multiple labels

To add labels when creating a node, use the syntax below. In this case, we add two labels.
Ex: CREATE (n:Person:Swedish)

5. Create node and add labels and properties

When creating a new node with labels, you can add properties at the same time
Syntax: Create(n:lablename {properties and values});
Ex: CREATE (n:Person {name: 'Andy', title: 'Developer'})

6. create nodes with parameters as properties

Syntax: 1. Define the property with parameter name. below example props is the parameter name
Ex:

```
  "props" : {
    "name" : "Andy",
    "position" : "Developer"
  }
}
```

2 add the parameter using the create clause

```
CREATE (n:Person $props)
RETURN n
```

2. Create Relationships between the nodes

Syntax:

```
Match (node1) ,(node2)
Where condition
Create (node1) [relation type ] ->(node2)
```

Ex:

```
Match(u:university),(p:Person)
Where p.name='x' and u.name='pes'
Create(p)-[stu:studiedAT] -> (u)
```

Creates the relationship Studiedat between person x and pes university
Use match(n) return(n) to see the result

3. Read nodes and attributes (Node finding)

1. Get all nodes
By just specifying a pattern with a single node and no labels, all nodes in the graph will be returned**.**
Match(n) return(n)

2. Get all nodes with a label
Getting all nodes with a label on them is done with a single node pattern where the node has a label on it.
MATCH (movie:Movie)
RETURN movie.title
Returns all the movies in the database.

3. Related nodes
The symbol -- means related to, without regard to type or direction of the relationship.
MATCH (director {name: 'Oliver Stone'})--(movie)
RETURN movie.title
Returns all the movies directed by 'Oliver Stone'.

4. Update or set a value

Syntax:

MATCH (n:Node)

Set n. propertyvalue = ‘newvalue’

5. Delete operation

1. Delete all node
To delete a node, use the DELETE clause.
Match(n) delete (n)   // delete all the nodes
If the relationship exist we need to delete the relationship first before we delete the node
Delete the relationship
Match(n) detach (n)

2. Delete single node Syntax: match(filter) delete (n) Ex: MATCH (n:Person {name: 'UNKNOWN'}) DELETE n