TASK 10

TASK 11: CRUD operations in Graph databases

Perform GraphQL/Neo4j graph space design for recommendation engines. Also perform CRUD operations like creating, inserting, querying, finding, deleting operations on graph spaces.

AIM: To perform CRUD operations like creating, inserting, querying, finding, deleting operations on graph spaces.

Create Node with Properties

Properties are the key-value pairs using which a node stores data. You can create a node with properties using the CREATE clause. You need to specify these properties separated by commas within the flower braces "{ }".

Syntax

Following is the syntax to create a node with properties.

CREATE (node:label { key1: value, key2: value,.................. })

Returning the Created Node

To verify the creation of the node, type and execute the following query in the dollar prompt.

MATCH (n) RETURN n

Creating Relationships

We can create a relationship using the CREATE clause. We will specify relationship within the square braces "[ ]" depending on the direction of the relationship it is placed between hyphen " - " and arrow " → " as shown in the following syntax.

Syntax

Following is the syntax to create a relationship using the CREATE clause.

CREATE (node1)-[:RelationshipType]->(node2)

Creating a Relationship Between the Existing Nodes

You can also create a relationship between the existing nodes using the MATCH clause.

Syntax

Following is the syntax to create a relationship using the MATCH clause.

MATCH (a:LabeofNode1), (b:LabeofNode2)

WHERE a.name = "nameofnode1" AND b.name = " nameofnode2"

CREATE (a)-[: Relation]->(b)

RETURN a,b

Deleting a Particular Node

To delete a particular node, you need to specify the details of the node in the place of “n” in the above query.

Syntax

Following is the syntax to delete a particular node from Neo4j using the DELETE clause.

MATCH (node:label {properties ....})

DETACH DELETE node

Create a graph database for student course registration, create student and dept node and insert values of properties.

1. Create Nodes:

CREATE (n:student {Sid: "VTU14500", Sname: "John", deptname: "CSE"})

Output: Added 1 label, created 1 node, set 3 properties, completed after 232 ms.

CREATE (n:student {Sid: "VTU14501", Sname: "Dharsana", deptname: "EEE"})

Output: Added 1 label, created 1 node, set 3 properties, completed after 16 ms.

CREATE (n:student {Sid: "VTU14502", Sname: "Vijay", deptname: "CSE"})

Output: Added 1 label, created 1 node, set 3 properties, completed after 12 ms.

CREATE (n:dept {deptname: "CSE", deptid: "d001"})

Output: Added 1 label, created 1 node, set 2 properties, completed after 72 ms.

2. Match Command to Select All Nodes:

```
MATCH (n) RETURN n
```

3. Match Command to Select All Student Nodes:

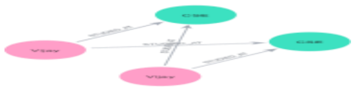
```
MATCH (n:student) RETURN n
```

4. Create Relationships Between Students and Departments with Arrows (Directed Relationships):

a. Create Relationship for Student "Vijay" with Department "CSE":

```
MATCH (s:student), (d:dept)
WHERE s.Sname = 'Vijay' AND d.deptname = 'CSE'
CREATE (s)-[st:STUDIED_AT]->(d)
RETURN s, st, d
```

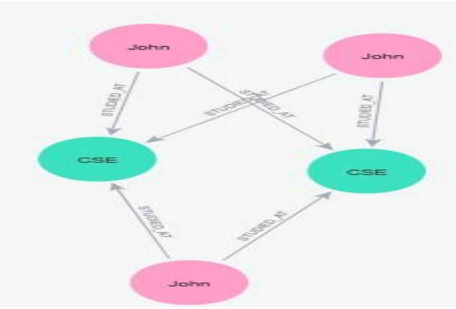
Output (Visualization with Arrows):



b. Create Relationship for Student "John" with Department "CSE":

```
MATCH (s:student), (d:dept)
WHERE s.Sname = 'John' AND d.deptname = 'CSE'
CREATE (s)-[st:STUDIED_AT]->(d)
RETURN s, st, d
```

Output (Visualization with Arrows):



5. Match All Nodes Again (with Arrows in the Output):

```
MATCH (n) RETURN n
```

- This will return all nodes with the relationships, and in the graph visualization, the arrows will indicate the direction of relationships.

6. Delete a Node from Student (Delete Dharsana):

```
MATCH (n:student {Sname: 'Dharsana'})
DELETE n
```

Output:

```
Deleted 1 node, completed after 10834 ms.
```