**TASK 11- CRUD OPERATIONS IN GRAPH DATABASES**

**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.**

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(w: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"})

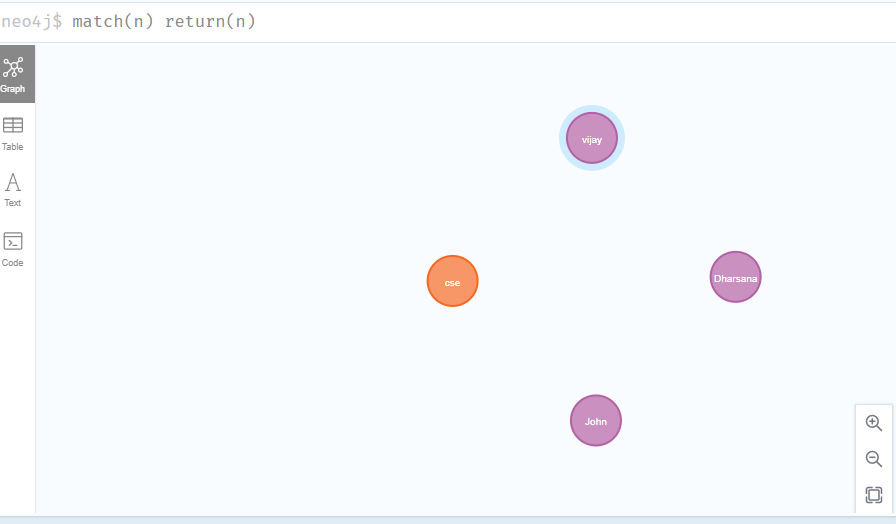
**OUTPUT:**

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

**Select all the nodes in your database using match command**

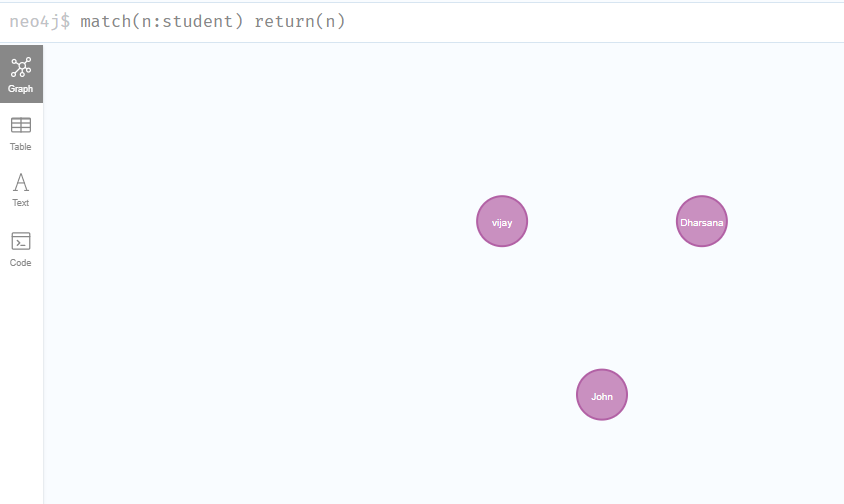
* **match(n) return(n)**

**OUTPUT**



* **match(n:student) return(n)**

**OUTPUT:**



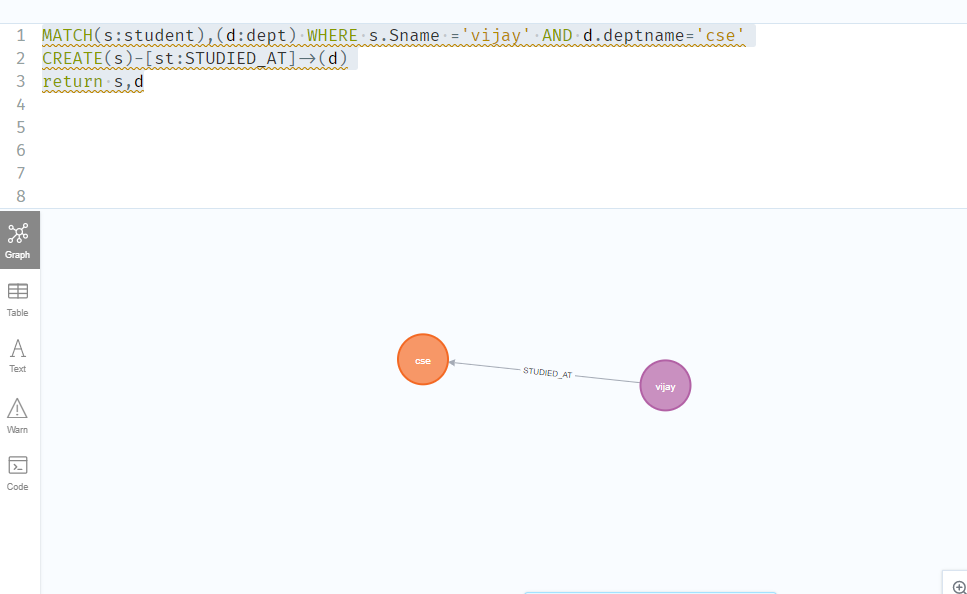
1. **Create relationship between student and cse** .

MATCH(s:student),(d:dept) WHERE s.Sname ='vijay' AND d.deptname='cse'

CREATE(s)-[st:STUDIED\_AT]->(d)

return s,d

**OUTPUT:**

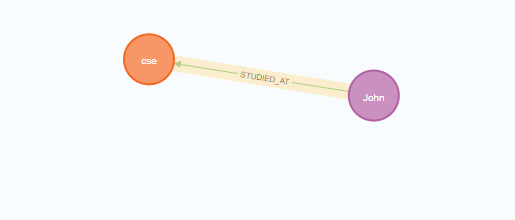


MATCH(s:student),(d:dept) WHERE s.Sname ='John' AND d.deptname='cse'

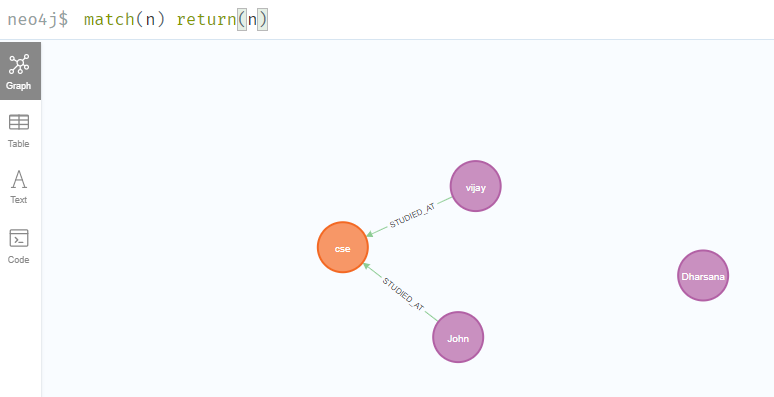
CREATE(s)-[st:STUDIED\_AT]->(d)

return s,d

OUTPUT:



**match(n) return(n)**

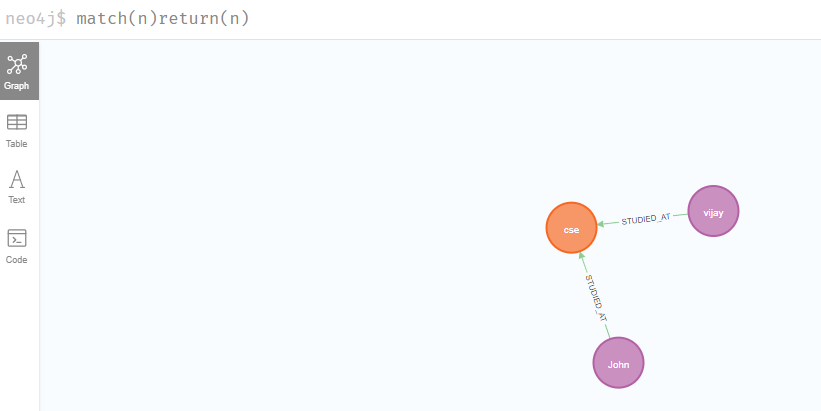


1. **Delete a node from student**

match(n:student{Sname:'Dharsana'}) DELETE(n)

**OUTPUT:**

Deleted 1 node, complaeted after 10834 ms.



**Result**

The implementation of CRUD operations like creating, inserting, finding and removing operations using GraphDB is successfully executed.