Task No: 11 CRUD Operations in Graph Database

Dt: 14/10/25

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 clouse. you need to specify these properties seperated by commas within the flower braces "{}".

Syntax:

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

Returning the Create 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 clouse we will specify the relationship within the square braces "[]" depending on the direction of the relationship it is placed between hypen "-" and arrow "→".

CREATE (node 1)-[:Relationship Type]→(node 2)

Creating Relationship between the existing nodes

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

Syntax:

MATCH (a: label of Node 1), (b: label of Node 2)

WHERE a.name = "name of node 1" AND

b.name = "name of node 2"

CREATE (a)-[: Relation]→(b)

RETURN a, b;

Delete 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:

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: "VTU1500", sname: "John", deptname: "cs
```

Output:

Added, label, created, node, set 3 properties,

Completed after 232 ms

```
Create (n: Student {sid: "VTU14501", sname: "Dharsana", deptnam
"EEE"})
```

Output:

Added 1 label, created, node, set 3 properties completed afte
16ms

```
Create (n: Student {sid: "VTU14502", sname: "Vijay", dept nam
"CSE"})
```

Output:

Added 1 label, created, node, set 3 properties completed aft
12ms

```
Create (n: dept {dept name: "csc", deptid: "dep1"})
```

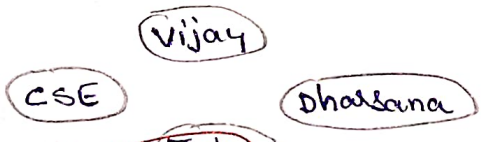
Output:

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

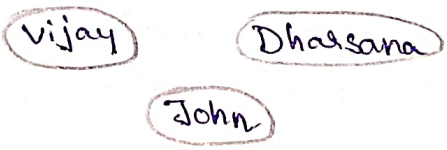
Select all the nodes in your database using match Command

```
match (n) return (n)
```

Output



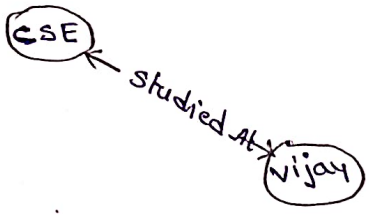

```
match (n: Student) return (n)
```



a) create Relationship between student and CSE

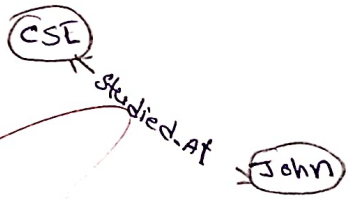
```
MATCH (S: student), (d : dept) WHERE S.Sname = 'Vijay'
AND d.dept name = 'CSE',
CREATE (S) - [st: STUDIED - AT] → (d)
return s,d
```

Output :

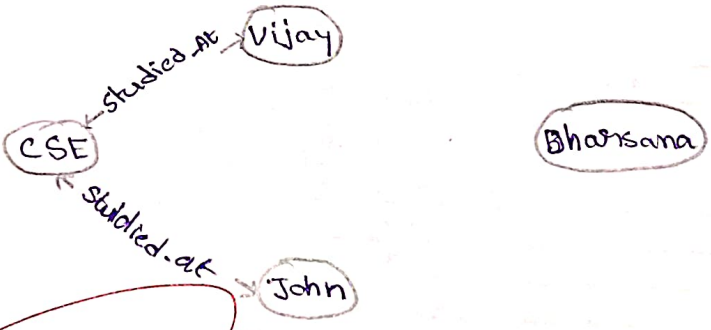



```
MATCH (S: student), (d: dept) where S.Sname = 'John' AND d.dept
Name = 'CSE"
CREATE (S) - [st: STUDIED - AT] → (d)
return s,d
```

Output :

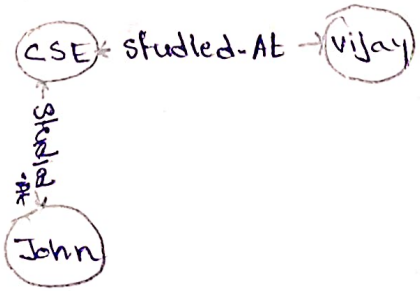


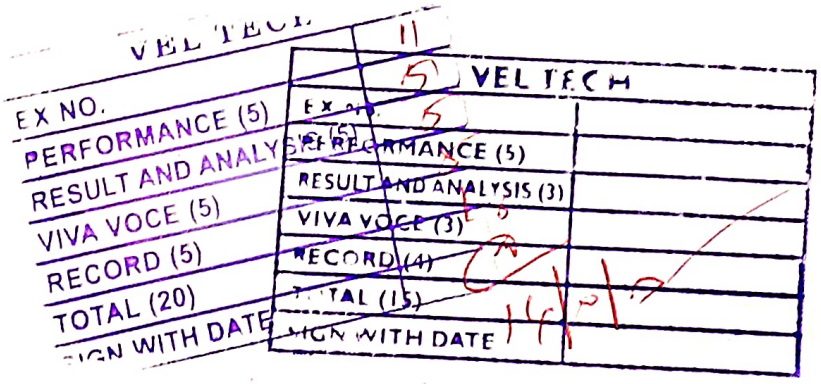

Match (n) return(n)




b) delete a node from student :

```
match (n: student {sname: 'Dhansana'}) DELETE (n)
Deleted, node, completed after 10834 ms
```






Result:

Thus the implementation of CRUD operations like Creating, inserting, finding and removing operations using Graph DB is successfully executed.