

TASK-11 CRUD Operations in Graph databases

Aim: To Perform CRUD operations like creating, inserting querying finding deleting operations on graph space

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 {key: value, key; value, ...})`

creating Relationships:-

we can create a relationship using CREATE clause we will specify Relationship with in the square braces "`[]`" depending on the directions of the relationship.

Syntax:

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

CREATE (node1) - [: Relationship Type] -> (node2)
creating Relationship Between the Existing Node
you can also create a relationship between the
existing node using the MATCH clause
syntax

following is the syntax to create a relationship
causing the MATCH clause

```
MATCH (a:label1 Node1) (b:label2 Node2)
WHERE a.name = "name of node 1" AND b.name =
      "name of node 2"
RETURN a, b
```

Deleting a Particular Node

To delete Particular node you need to specify
the details of node in the place of "n" in the
above query
syntax:

following is the syntax to delete a Particular
node from Neovt using DELETE clause
MATCH (node:label {properties...})

DETACH DELETE node

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


```
create (n:student {sid:"uiu14500",  
  sname:"John"  
  deptname:"cse"})  
output
```

added 1 label created 1 node, set 3 Properties
completed after 232 ms

```
create (n:student {sid:"uiu14501",  
  sname:"Dhoshona"  
  deptname:"EEE"})  
output
```

Added 1 ~~label~~ created 1 node, set 3 Properties
completed after 16 ms

```
create (n:dept {deptname:"cse"; deptid:"d001"})  
output
```

Added 1 label, created 1 node, set 2

Properties, completed after 72 ms

select all nodes in your database using match
command.

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

output

Vijay

CSE

Dharsana

John

Amatch (n: student s) return (n)

output

Vijay

Dharsana

John

~~match (s: student) (d: dept) WHERE s.name =~~

~~"Vijay" AND d.deptname = "CSE"~~

return s.d

output

CSE ← student → Vijay
- AT

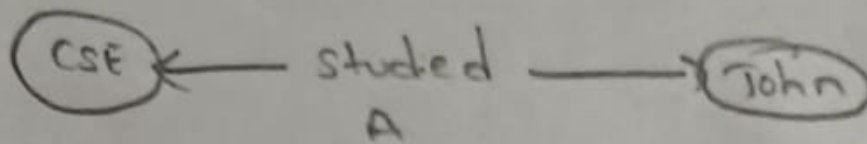
~~MATCH (s: student) (d: dept) WHERE s.sname = 'John'~~

~~AND d.deptname = "CSE"~~

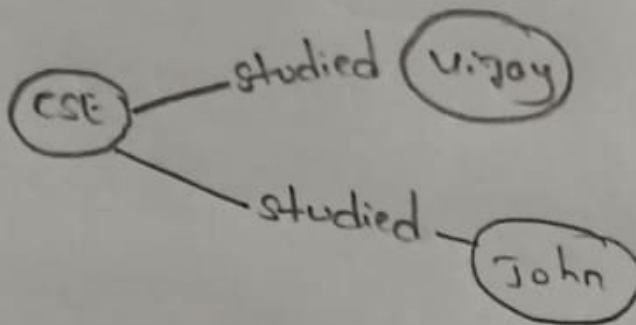
~~CREATE (s)-[s: STUDIED-AT]->(d)~~

return s.d

output



match(n) return(n)



Dharsana

VEL TECH	
X NO.	
PERFORMANCE (5)	4/5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGN WITH DATE	

Result:

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