

DBMS Laboratory

UE19CS304

5th Semester, Academic Year 2021-22

Week #: 2 - Operation on Neo4j GraphDB

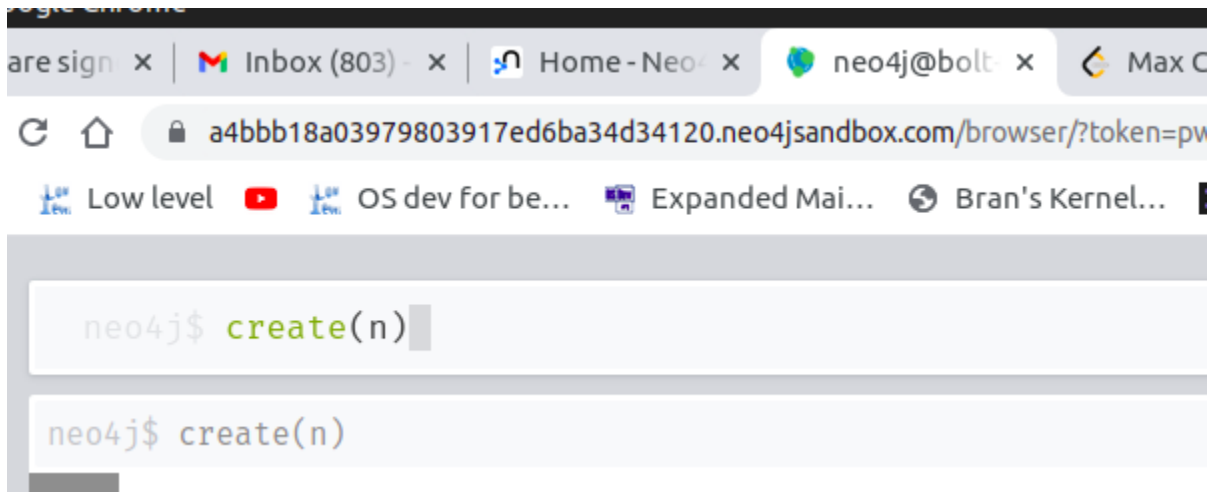
Date: 14/9/2021

Name : SUMUKH RAJU BHAT	SRN : PES1UG19CS519	Section : H
----------------------------	------------------------	----------------

1. Create a single node

create (n)

- Create a node with no props or labels

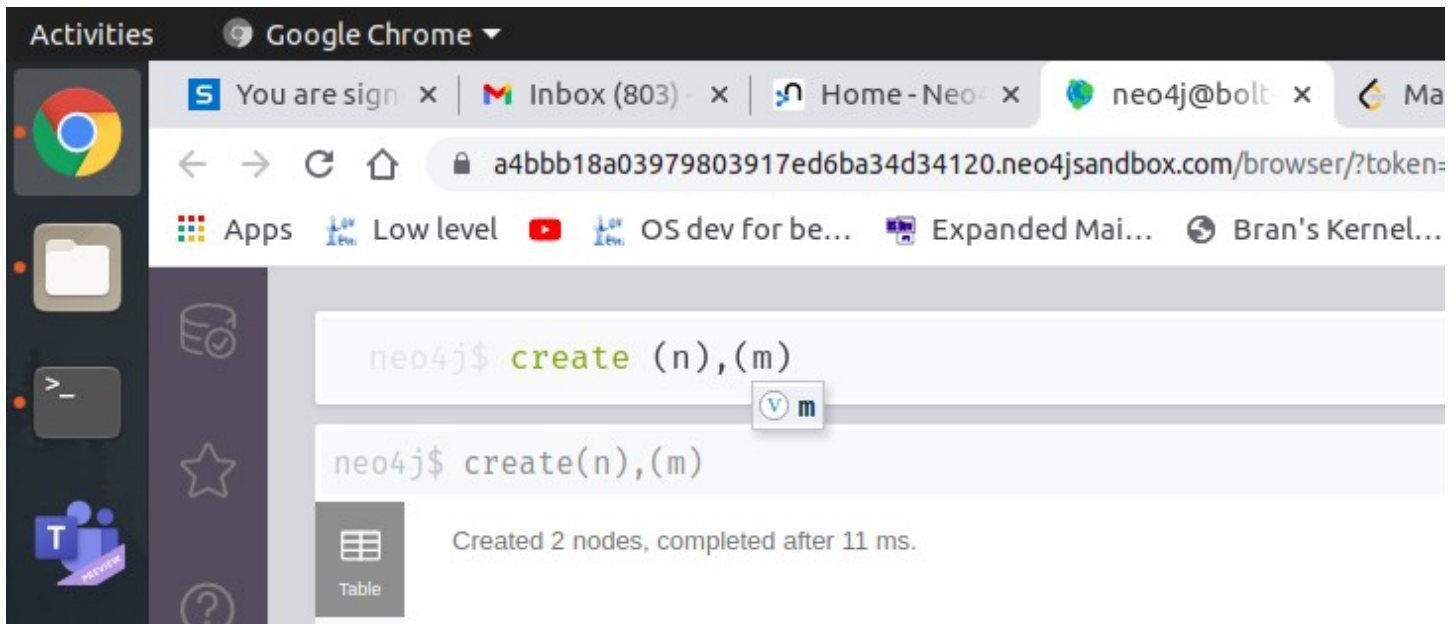


```
neo4j$ create(n)
```

2. Create multiple nodes

create (n), (m)

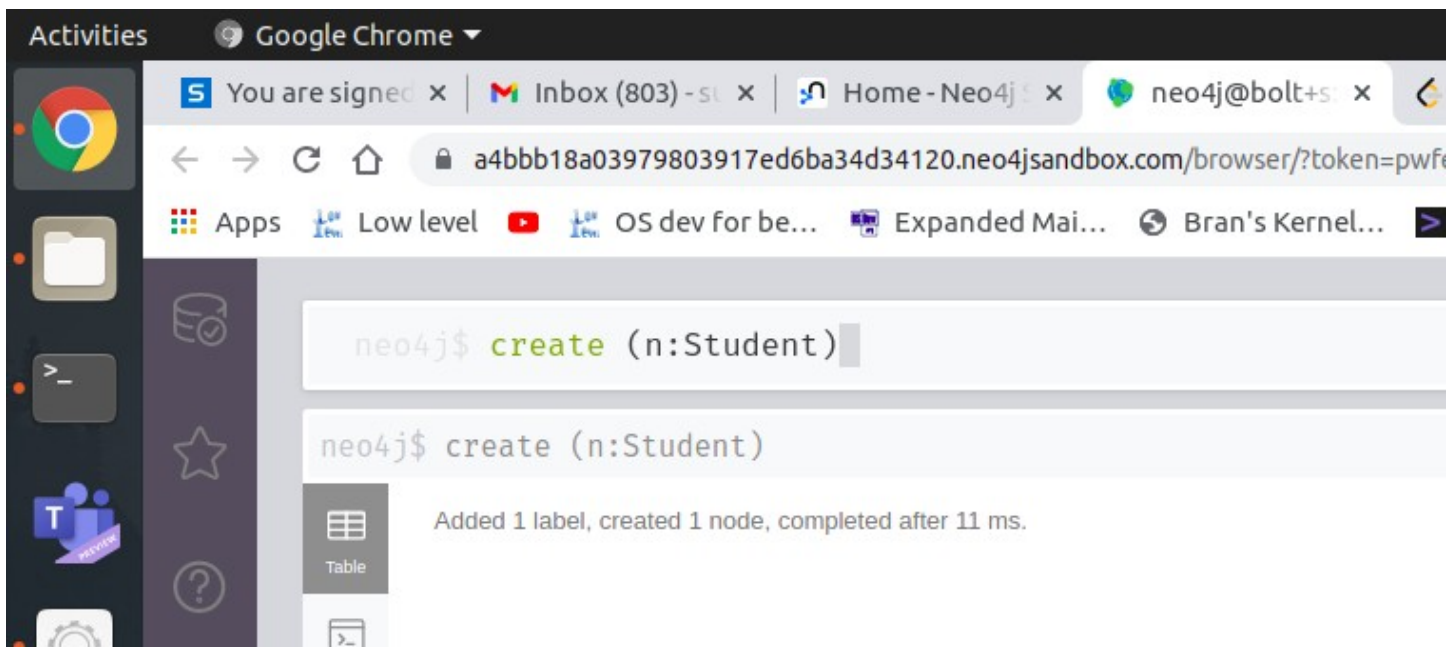
- Create multiple nodes with no labels or props



3. Create a node with label

create (n:Student)

- Create a node with Student label and no props



4. Create a node with label and properties

create (n:Student {name: "ABC", SRN: "123"})

create (n:Student {name: "LMN", SRN: "456"})

create (n:Student {name: "JHI", SRN: "789"})

create (n:Teacher {name: "ZHA", FID: "439"})

create (n:Teacher{name: "OPQ", FID: "731"})

- Create 3 nodes with label Student and props name, SRN and 2 nodes with label Teacher with props name, FID.



Result:



5. Create relation between specific nodes

match (x:Teacher), (y:Student)

where x.name = "ZHA" and y.name = "JHI"

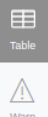
create (x)-[tea:teaches]->(y)

- Creates a "teaches" relationship between a teacher JHI and student ZHA.

```
1 match (x:Teacher), (y:Student)
2 where x.name = "ZHA" and y.name = "JHI"
3 create(x) -[te: teaches]->(y)
```

```
neo4j$ match (x:Teacher), (y:Student) where x.name = "ZHA" and y.name = "JHI" create(x) -[te: teaches]->(y)
```

Created 1 relationship, completed after 18 ms.

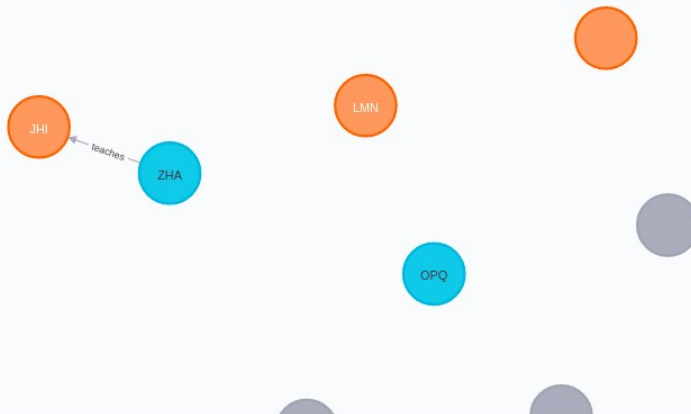


Result:

```
neo4j$
```

```
neo4j$ match(n) return n
```

*(6) Student(4) Teacher(2)
*(1) teaches(1)



Displaying 9 nodes, 1 relationships.

6. Create relation between nodes

```
match(x:Student), (y:Teacher)
```

```
create (y)-[]->(x)
```

- Create a relationship between all Teacher and Student nodes as “teaches”

```
1 match(x:Student),(y:Teacher)
2 create (y)-[tea2:teaches]→(x)
```

neo4j\$ match(n) return n

Graph

*(6) Student(4) Teacher(2)

*(9) teaches(9)

Table

Text

Code

Displaying 6 nodes, 9 relationships

7. Display the graph

match(n) return n

- Displays all the nodes

```
neo4j$ match(n) return n
```

Graph

*(6) Student(4) Teacher(2)

*(9) teaches(9)

Table

Text

Code

Displaying 6 nodes, 9 relationships

8. Display specific nodes based on label and ids

match(n:Teacher) return n

- Displays only the nodes with label Teacher

The image shows the Neo4j Cypher query editor. The query entered is `neo4j$ match(n:Teacher) return n`. The results are displayed in a graph view, showing two blue circular nodes labeled "ZHA" and "OPQ". The left sidebar contains icons for Graph, Table, Text, and Code, with the Graph icon selected. Above the graph view, a summary bar indicates `*(2)` and `Teacher(2)`.

match(n:Teacher) where id(n) = 8 return n

- Displays the Teacher node with condition that id = 8

The image shows the Neo4j Cypher query editor. The query entered is `1 match(n:Teacher)`, `2 where id(n) = 8`, and `3 return n`. The results are displayed in a graph view, showing a single blue circular node labeled "ZHA". The left sidebar contains icons for Graph, Table, Text, and Code, with the Graph icon selected. Above the graph view, a summary bar indicates `*(1)` and `Teacher(1)`.

match(n:Student) return n.name

- Returns the name prop of all the Student labeled nodes.

```
neo4j$ match(n:Student) return n.name
```

```
neo4j$ match(n:Student) return n.name
```



Table



Text



Code

n.name

1

null

2

"ABC"

3

"LMN"

4

"JHI"

Started streaming 4 records after 9 ms and completed after 10 ms

9. Display Relationships

match(x)--(y) return x

- Return all the relationships in the db. The output showcases the output only when relationship was added.

```
neo4j$ match(x)--(y) return x
```

```
neo4j$ match(x)--(y) return x
```



Graph



Table



Text



Code

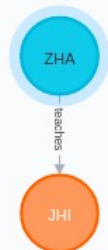
*(2)

Student(1)

Teacher(1)

*(1)

teaches(1)



match(x) -- (y) return x.name

- Return the name prop of all the nodes who have a outgoing(In the example x is related to y implies x has a outgoing relationship and y has incoming relationship) relationships in the db. (The output showcases the output only when relationship was added.)

```
neo4j$ match(x)--(y) return x.name
```

name

```
neo4j$ match(x)--(y) return x.name
```

	x.name
1	"JHI"
2	"ZHA"

10. Update specific node

match(x:Student {name: "JHI"}) Set x.name = "GTA"

- Update the name prop of the Student node where name was originally JHI. (The output showcases the output only when relationship was added.)

```
neo4j$ match(x:Student {name: "JHI"}) Set x.name = "GTA"
```

```
neo4j$ match(x:Student {name: "JHI"}) Set x.name = "GTA"
```

Set 1 property, completed after 21 ms.

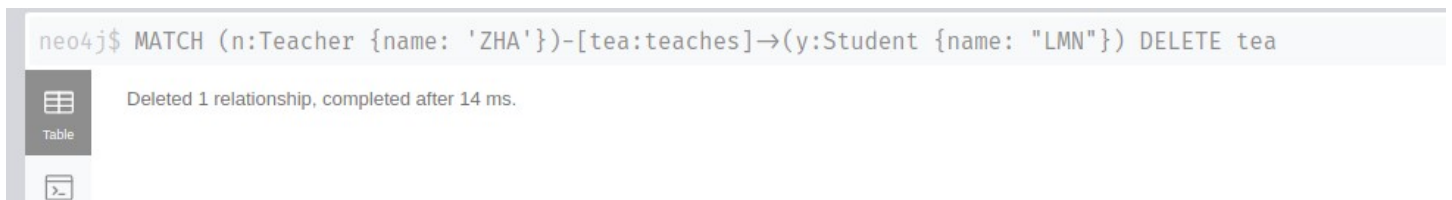
Result:



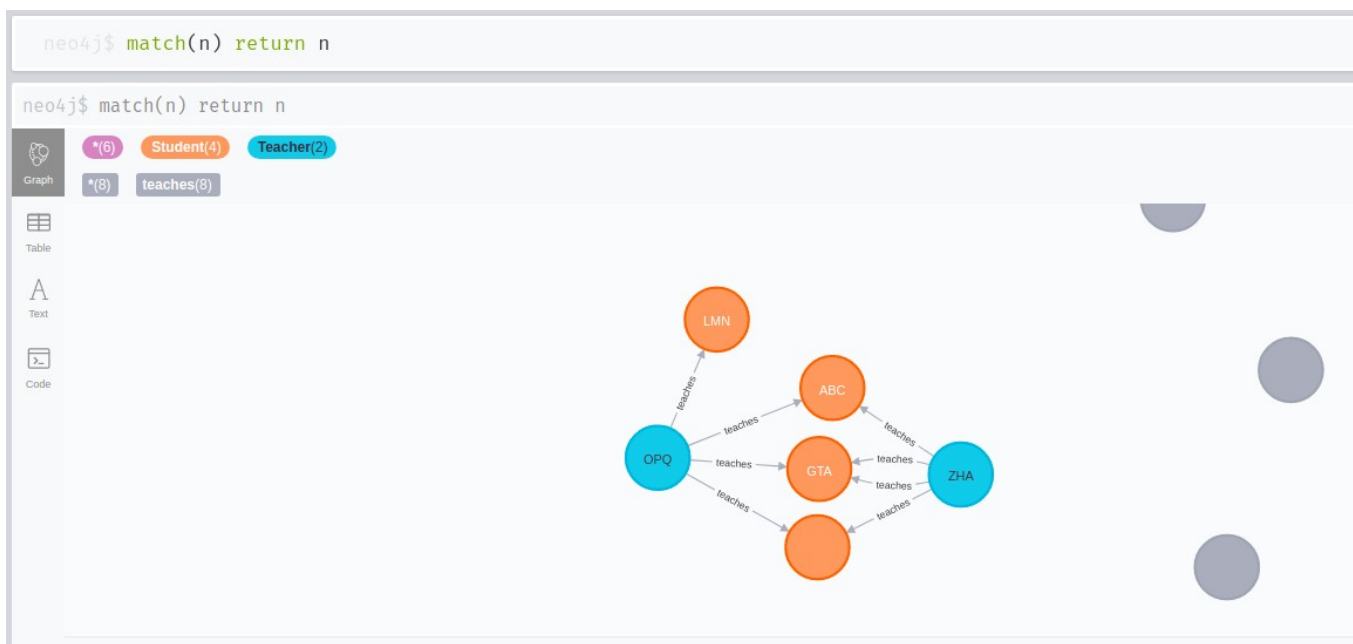
11. Delete certain relationship

match(n:Teacher {name: "ZHA"})-[tea:teaches]->(y:Student {name: "LMN"}) delete tea

- Delete the teaches relationship between Teacher node with name prop LMN and Student node with name prop LMN.



Result:



12. Delete all relationship

match (n)-[tea:teaches]->(y:Student) delete tea

- Deletes all the teaches relationship between any node to Student nodes.

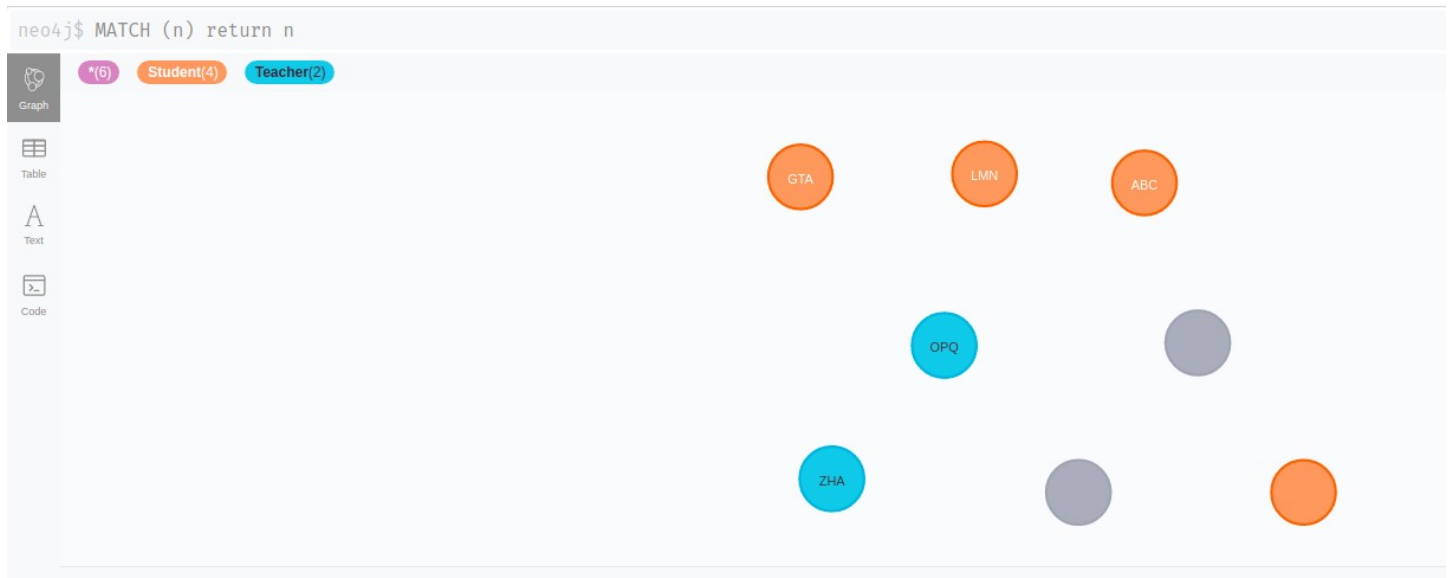
w level  OS dev for be...  Expanded Mai...  Bran's Kernel...  The Miss

```
neo4j$ MATCH (n)-[tea:teaches]->(y:Student) DELETE tea
```

```
j$ MATCH (n)-[tea:teaches]->(y:Student) DELETE tea
```

Deleted 8 relationships, completed after 14 ms.

Result:



13. Delete certain node

match(n) where ID(n) = 0 delete n

- Delete the node with id = 0

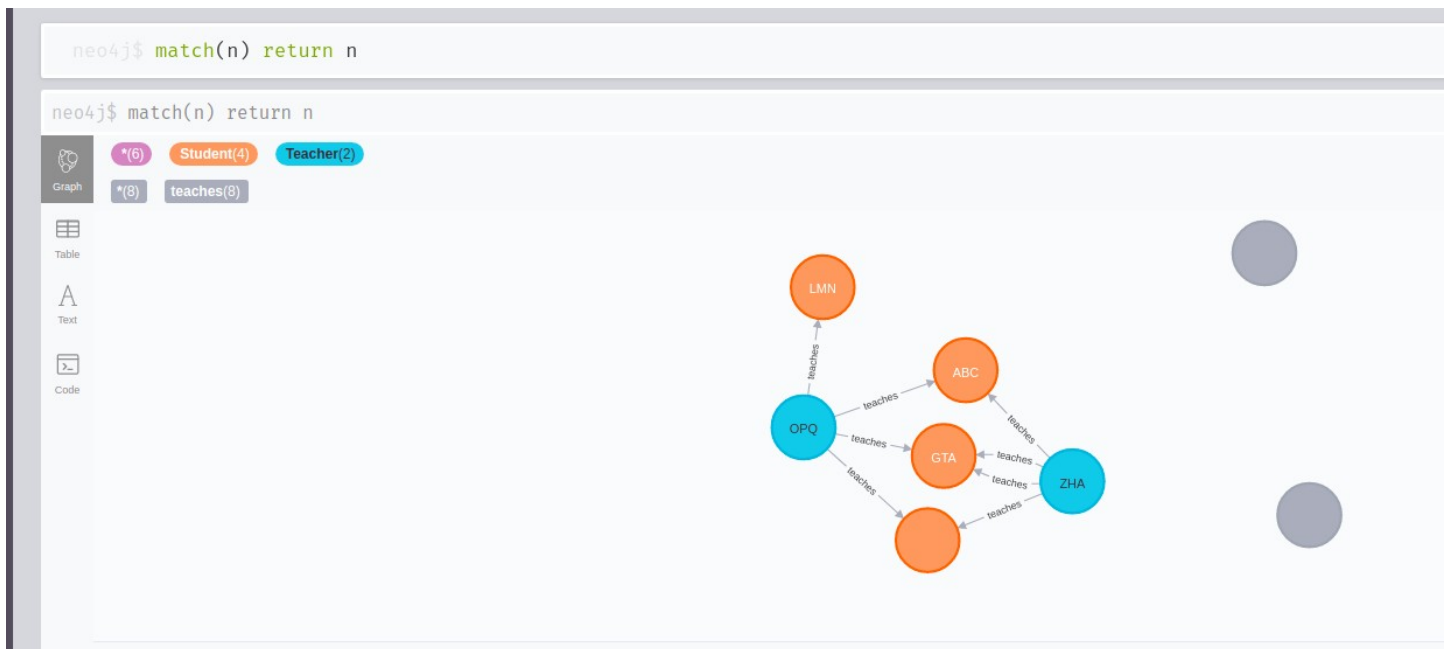
```
neo4j$ match(n) where ID(n) = 0 DELETE n
```

Deleted 1 node, completed after 9 ms.

Table

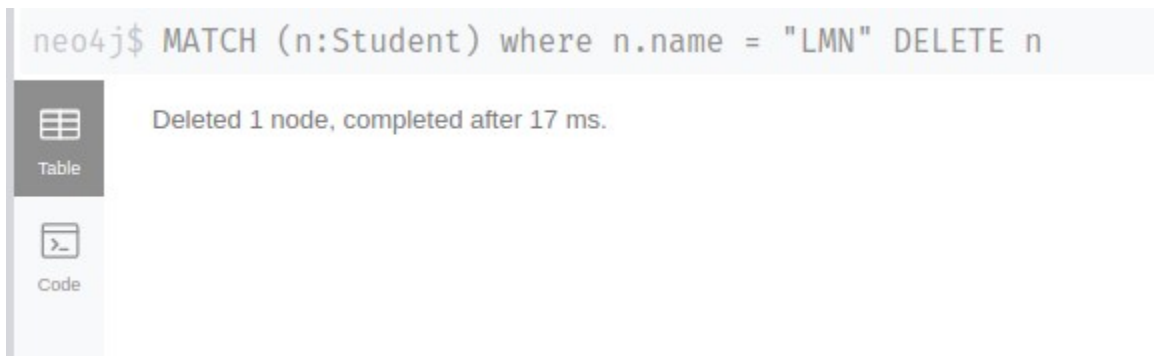
Code

Result:



match(n:Student) where n.name = "LMN" delete n

- Delete the Student node with name prop as LMN



Result:





14. Delete all nodes

match(n) delete n

- After all relationships are deleted, delete all the nodes in the db

```
neo4j$ match(n) delete n
```


```
neo4j$ match(n) delete n
```

 Deleted 7 nodes, completed after 22 ms.


Result:

```
neo4j$ match(n) return n
```

```
neo4j$ match(n) return n
```

 (no changes, no records)