Introduction to Graph database(NOSQL)

Week #2

Problem Statement	Use NEO4j and create a sample graph database and perform the following operations on it 1. Create node with varying fields. 2. Add properties to node 3. Add relationships between the nodes . 4. Update an attribute value of the node 5. Retrieve and delete nodes, relationship

sudo neo4j start

available at http://localhost:7474/

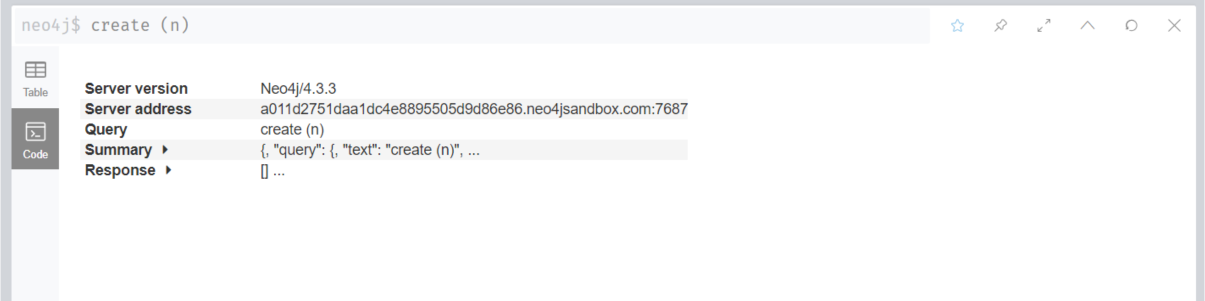
//clear all records

MATCH (n) DETACH DELETE n

1.Create node and relationships

I. *Create a single node :*

Syntax: create (n) // create a single node without label

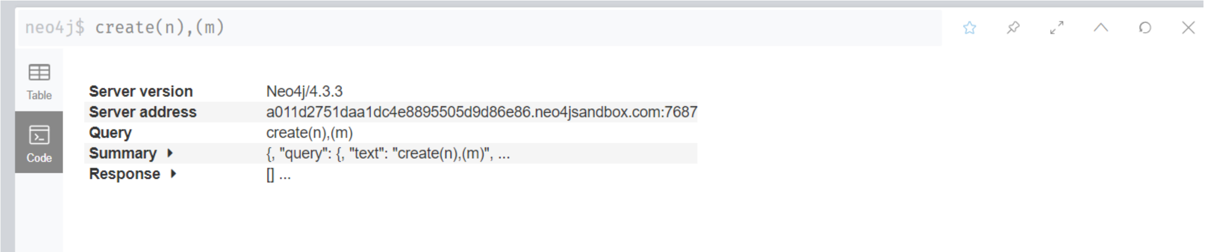


```
neo4j$ create (n)
```

```
Server version    Neo4j/4.3.3
Server address    a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query             create (n)
Summary ▸         {, "query": {, "text": "create (n)", ...
Response ▸        [] ...
```

2. *Create multiple nodes*

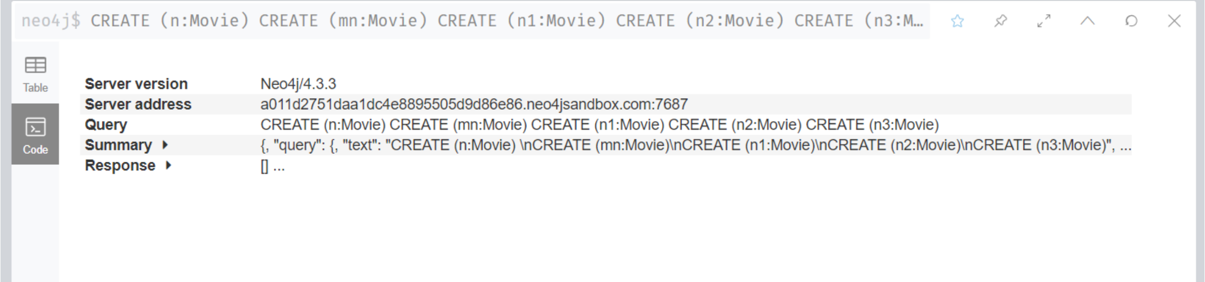
Syntax: create(n),(m)



```
neo4j$ create(n),(m)

Server version     Neo4j/4.3.3
Server address     a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query              create(n),(m)
Summary ▸          {, "query": {, "text": "create(n),(m)", ...
Response ▸         [] ...
```

3. *Create a node with a label*

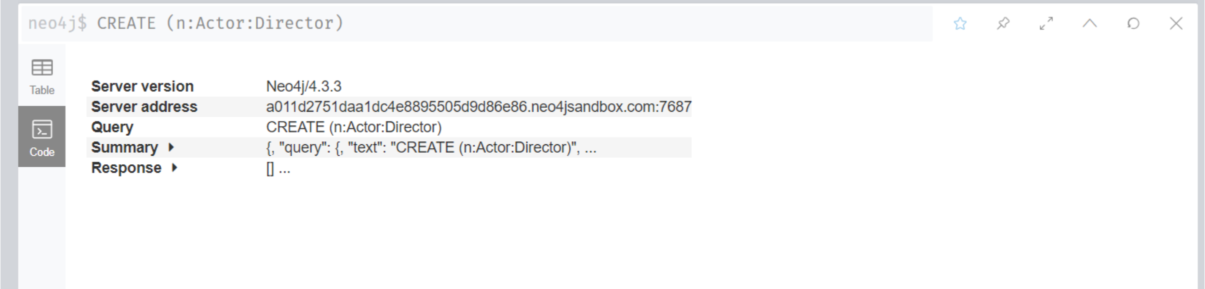
Syntax: create(n:lable name)



```
neo4j$ CREATE (n:Movie) CREATE (mn:Movie) CREATE (n1:Movie) CREATE (n2:Movie) CREATE (n3:M…

Server version     Neo4j/4.3.3
Server address     a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query              CREATE (n:Movie) CREATE (mn:Movie) CREATE (n1:Movie) CREATE (n2:Movie) CREATE (n3:Movie)
Summary ▸          {, "query": {, "text": "CREATE (n:Movie) \nCREATE (mn:Movie)\nCREATE (n1:Movie)\nCREATE (n2:Movie)\nCREATE (n3:Movie)", ...
Response ▸         [] ...
```

4. *Create a node with multiple labels*

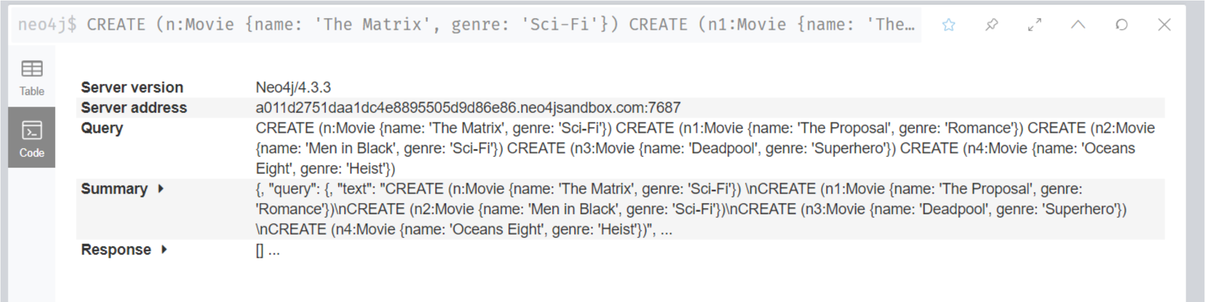
Syntax:CREATE (n:Label1:Label2)



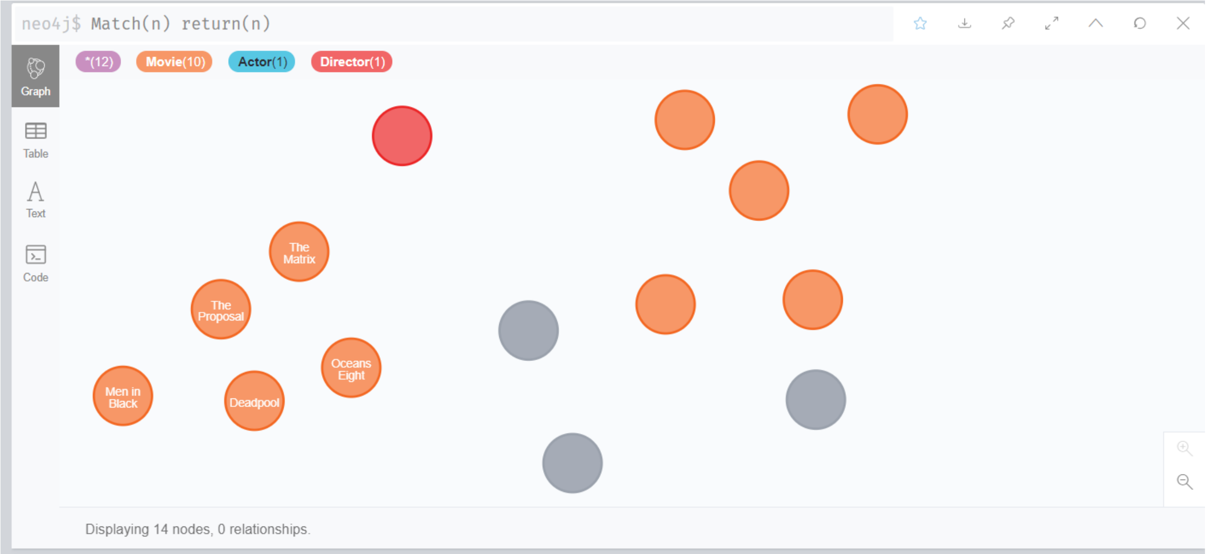
```
neo4j$ CREATE (n:Actor:Director)

Server version     Neo4j/4.3.3
Server address     a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query              CREATE (n:Actor:Director)
Summary ▸          {, "query": {, "text": "CREATE (n:Actor:Director)", ...
Response ▸         [] ...
```

5. *Create node and add labels and properties*

Syntax: Create(n:lablename {properties and values});



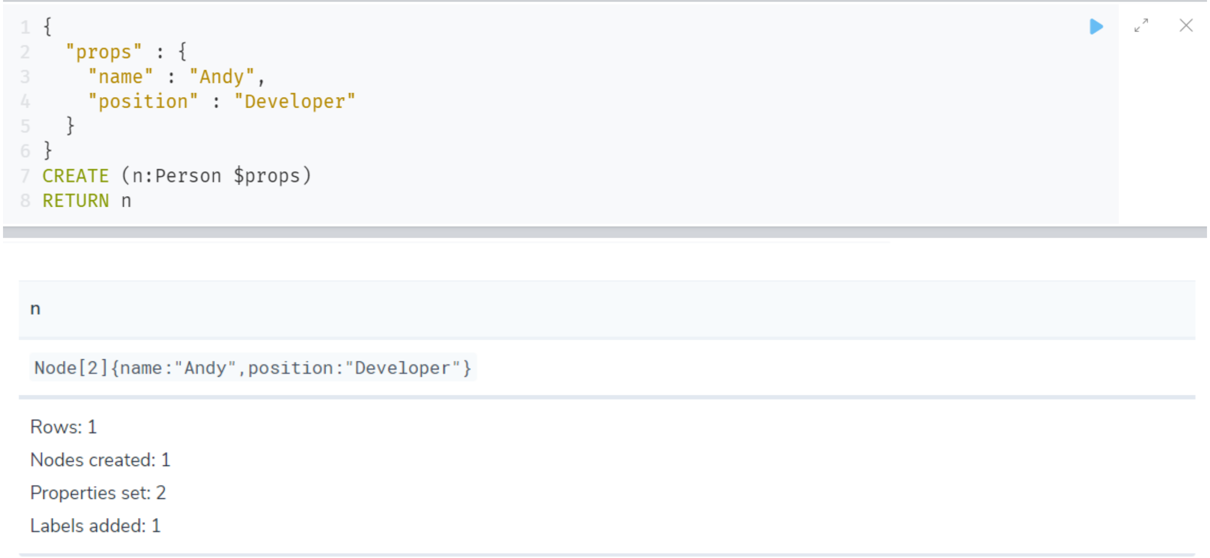
```
neo4j$ CREATE (n:Movie {name: 'The Matrix', genre: 'Sci-Fi'}) CREATE (n1:Movie {name: 'The…

Server version     Neo4j/4.3.3
Server address     a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query              CREATE (n:Movie {name: 'The Matrix', genre: 'Sci-Fi'}) CREATE (n1:Movie {name: 'The Proposal', genre: 'Romance'}) CREATE (n2:Movie
                   {name: 'Men in Black', genre: 'Sci-Fi'}) CREATE (n3:Movie {name: 'Deadpool', genre: 'Superhero'}) CREATE (n4:Movie {name: 'Oceans
                   Eight', genre: 'Heist'})
Summary ▸          {, "query": {, "text": "CREATE (n:Movie {name: 'The Matrix', genre: 'Sci-Fi'}) \nCREATE (n1:Movie {name: 'The Proposal', genre:
                   'Romance'})\nCREATE (n2:Movie {name: 'Men in Black', genre: 'Sci-Fi'})\nCREATE (n3:Movie {name: 'Deadpool', genre: 'Superhero'})
                   \nCREATE (n4:Movie {name: 'Oceans Eight', genre: 'Heist'})", ...
Response ▸         [] ...
```



```
neo4j$ Match(n) return(n)
```

Displaying 14 nodes, 0 relationships.

6. create nodes with parameters as properties



```
{
  "props" : {
    "name" : "Andy",
    "position" : "Developer"
  }
}
CREATE (n:Person $props)
RETURN n
```

```
n
Node[2]{name:"Andy",position:"Developer"}
```

Rows: 1

Nodes created: 1

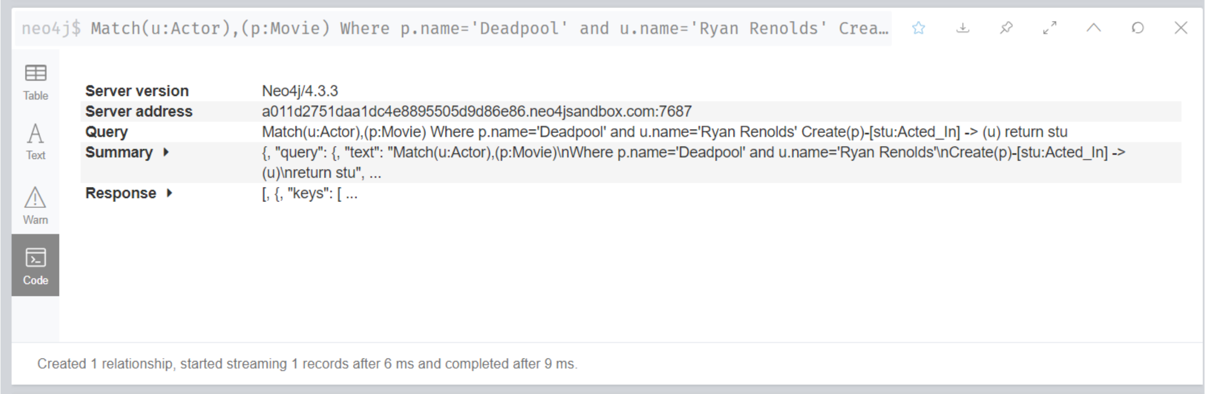
Properties set: 2

Labels added: 1

2. Create Relationships between the nodes

7. Create Relationships between the nodes

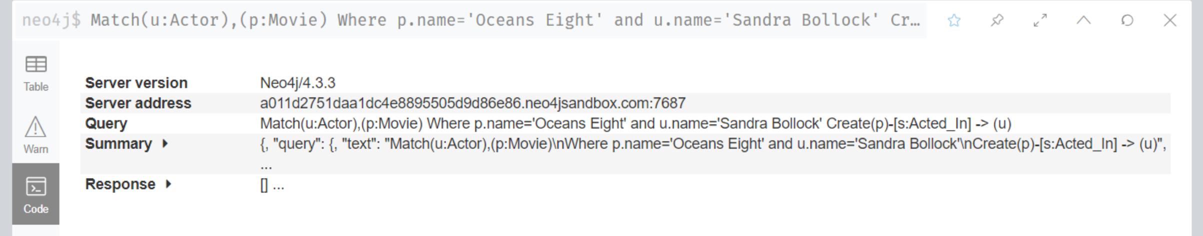
Syntax: Match (node1) ,(node2) Where condition Create (node1) [relation type] ->(node2)



```
neo4j$ Match(u:Actor),(p:Movie) Where p.name='Deadpool' and u.name='Ryan Renolds' Crea…
```

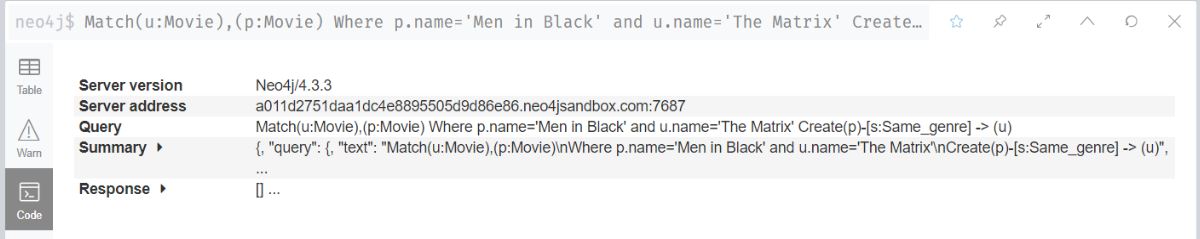
Server version	Neo4j/4.3.3
Server address	a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query	Match(u:Actor),(p:Movie) Where p.name='Deadpool' and u.name='Ryan Renolds' Create(p)-[stu:Acted_In] -> (u) return stu
Summary ▸	{, "query": {, "text": "Match(u:Actor),(p:Movie)\nWhere p.name='Deadpool' and u.name='Ryan Renolds'\nCreate(p)-[stu:Acted_In] -> (u)\nreturn stu", ...
Response ▸	[, {, "keys": [...

Created 1 relationship, started streaming 1 records after 6 ms and completed after 9 ms.



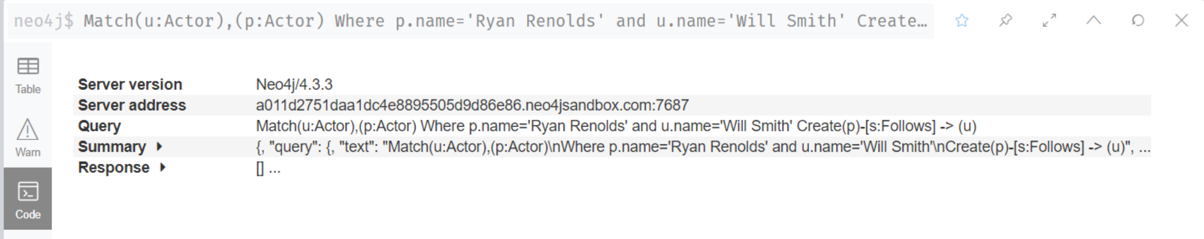
```
neo4j$ Match(u:Actor),(p:Movie) Where p.name='Oceans Eight' and u.name='Sandra Bollock' Cr…
```

```
Server version     Neo4j/4.3.3
Server address     a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query              Match(u:Actor),(p:Movie) Where p.name='Oceans Eight' and u.name='Sandra Bollock' Create(p)-[s:Acted_In] -> (u)
Summary ▸          {, "query": {, "text": "Match(u:Actor),(p:Movie)\nWhere p.name='Oceans Eight' and u.name='Sandra Bollock'\nCreate(p)-[s:Acted_In] -> (u)",
                   ...
Response ▸         [] ...
```



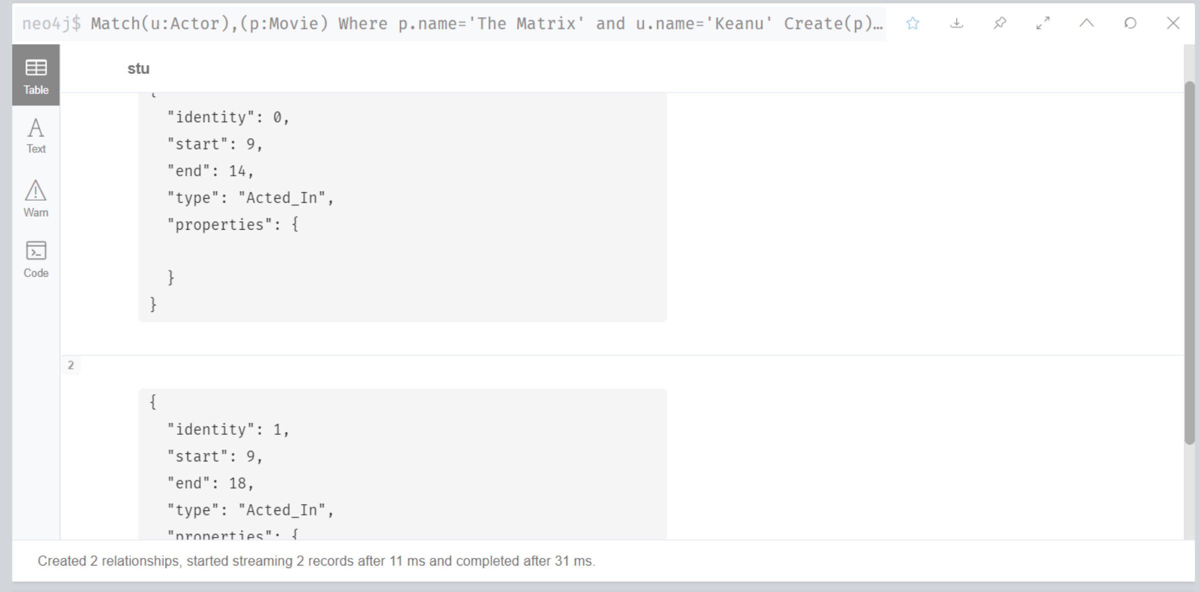
```
neo4j$ Match(u:Movie),(p:Movie) Where p.name='Men in Black' and u.name='The Matrix' Create…
```

```
Server version     Neo4j/4.3.3
Server address     a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query              Match(u:Movie),(p:Movie) Where p.name='Men in Black' and u.name='The Matrix' Create(p)-[s:Same_genre] -> (u)
Summary ▸          {, "query": {, "text": "Match(u:Movie),(p:Movie)\nWhere p.name='Men in Black' and u.name='The Matrix'\nCreate(p)-[s:Same_genre] -> (u)",
                   ...
Response ▸         [] ...
```



```
neo4j$ Match(u:Actor),(p:Actor) Where p.name='Ryan Renolds' and u.name='Will Smith' Create…
```

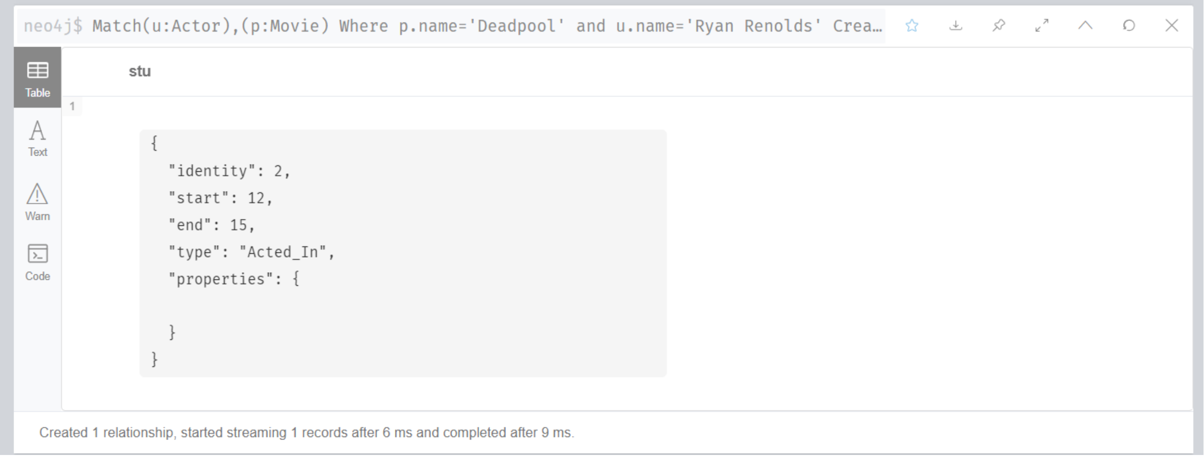
```
Server version     Neo4j/4.3.3
Server address     a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query              Match(u:Actor),(p:Actor) Where p.name='Ryan Renolds' and u.name='Will Smith' Create(p)-[s:Follows] -> (u)
Summary ▸          {, "query": {, "text": "Match(u:Actor),(p:Actor)\nWhere p.name='Ryan Renolds' and u.name='Will Smith'\nCreate(p)-[s:Follows] -> (u)", ...
Response ▸         [] ...
```



```
neo4j$ Match(u:Actor),(p:Movie) Where p.name='The Matrix' and u.name='Keanu' Create(p)…
```

```
        stu
        
          "identity": 0,
          "start": 9,
          "end": 14,
          "type": "Acted_In",
          "properties": {

          }
        }

2
        {
          "identity": 1,
          "start": 9,
          "end": 18,
          "type": "Acted_In",
          "properties": {
```

Created 2 relationships, started streaming 2 records after 11 ms and completed after 31 ms.

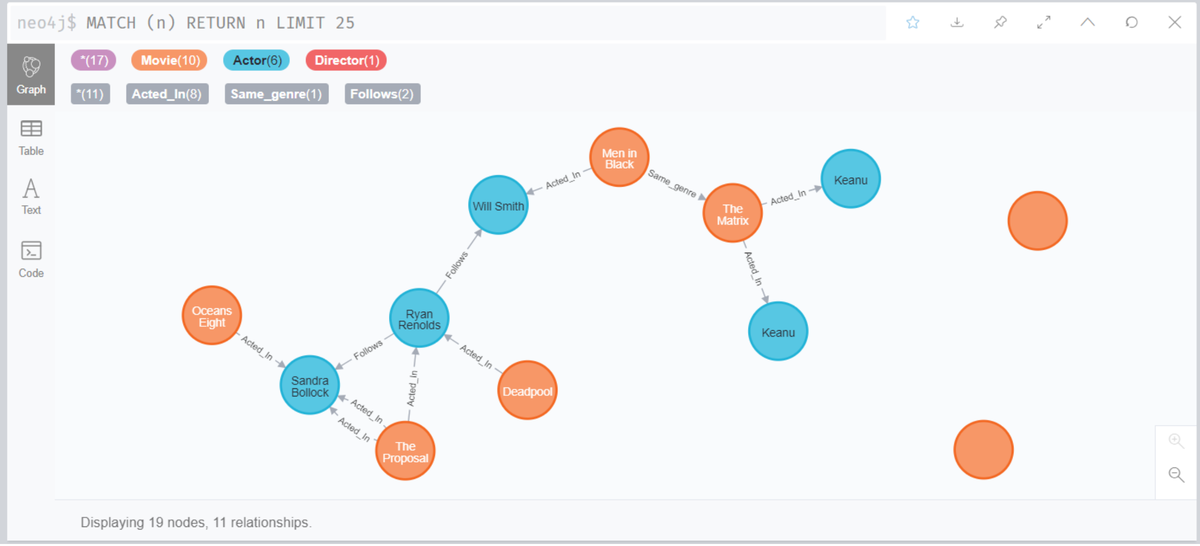


```
neo4j$ Match(u:Actor),(p:Movie) Where p.name='Deadpool' and u.name='Ryan Renolds' Crea…
```

stu

```
1
{
  "identity": 2,
  "start": 12,
  "end": 15,
  "type": "Acted_In",
  "properties": {

  }
}
```

Created 1 relationship, started streaming 1 records after 6 ms and completed after 9 ms.



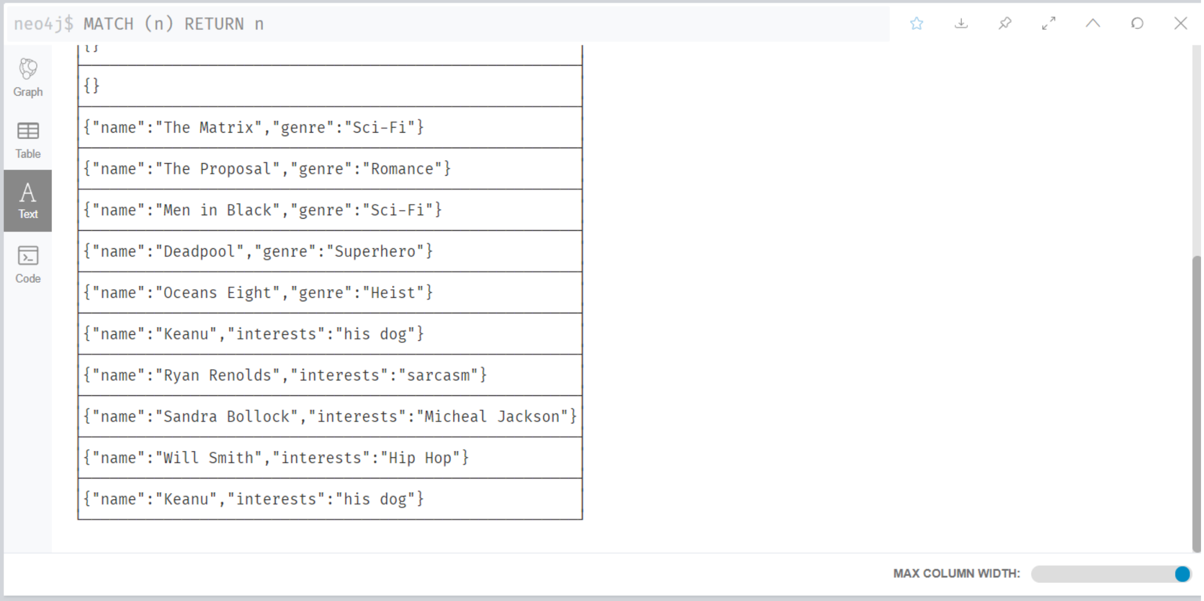
```
neo4j$ MATCH (n) RETURN n LIMIT 25
```

Displaying 19 nodes, 11 relationships.

3. Read nodes and attributes (Node finding)

8. *Get all nodes*

Syntax: Match(n) return(n)

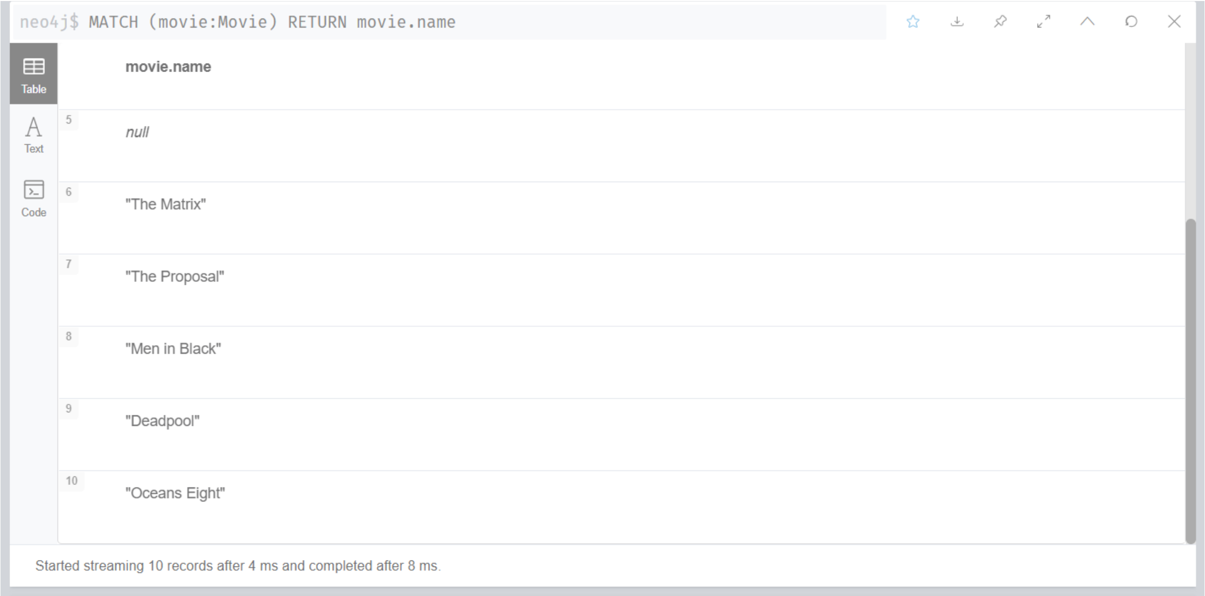


```
neo4j$ MATCH (n) RETURN n
```

```
{}
{}
{"name":"The Matrix","genre":"Sci-Fi"}
{"name":"The Proposal","genre":"Romance"}
{"name":"Men in Black","genre":"Sci-Fi"}
{"name":"Deadpool","genre":"Superhero"}
{"name":"Oceans Eight","genre":"Heist"}
{"name":"Keanu","interests":"his dog"}
{"name":"Ryan Renolds","interests":"sarcasm"}
{"name":"Sandra Bollock","interests":"Micheal Jackson"}
{"name":"Will Smith","interests":"Hip Hop"}
{"name":"Keanu","interests":"his dog"}
```

9. Get all nodes with a label

Syntax: MATCH (movie:Movie) RETURN movie.title



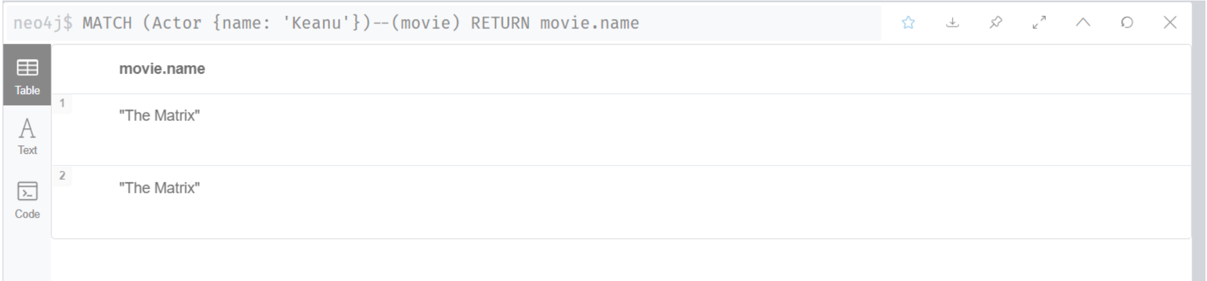
```
neo4j$ MATCH (movie:Movie) RETURN movie.name
```

	movie.name
5	*null*
6	"The Matrix"
7	"The Proposal"
8	"Men in Black"
9	"Deadpool"
10	"Oceans Eight"

Started streaming 10 records after 4 ms and completed after 8 ms.

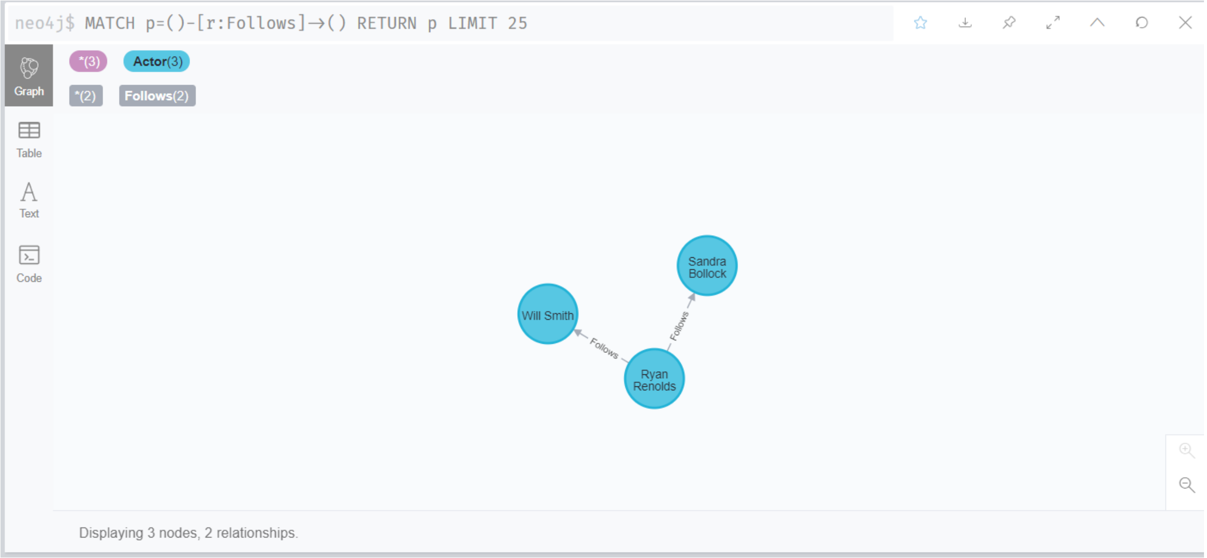
10. *Related nodes*

Syntax: MATCH (director {name: 'Oliver Stone'})--(movie) RETURN movie.title



```
neo4j$ MATCH (Actor {name: 'Keanu'})--(movie) RETURN movie.name
```

	movie.name
1	"The Matrix"
2	"The Matrix"



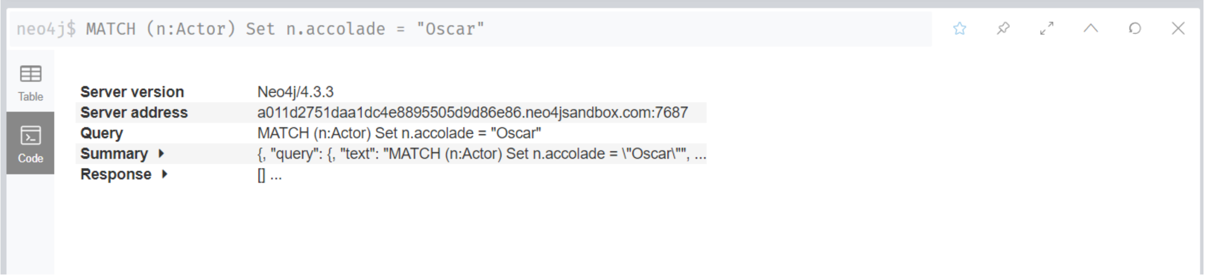
```
neo4j$ MATCH p=()-[r:Follows]→() RETURN p LIMIT 25
```

Displaying 3 nodes, 2 relationships.

4. Update or set a value

11. *Update/set a value*

Syntax: MATCH (n:Node) Set n. propertyvalue = ‘newvalue’



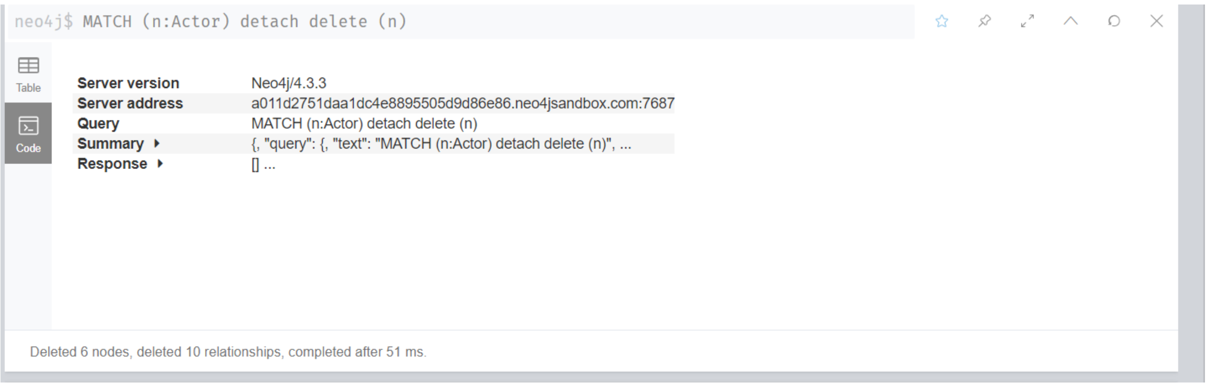
```
neo4j$ MATCH (n:Actor) Set n.accolade = "Oscar"
```

```
Server version    Neo4j/4.3.3
Server address    a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query             MATCH (n:Actor) Set n.accolade = "Oscar"
Summary ▸         {, "query": {, "text": "MATCH (n:Actor) Set n.accolade = \"Oscar\"", ...
Response ▸        [] ...
```

5. Delete operation

12. Delete all node

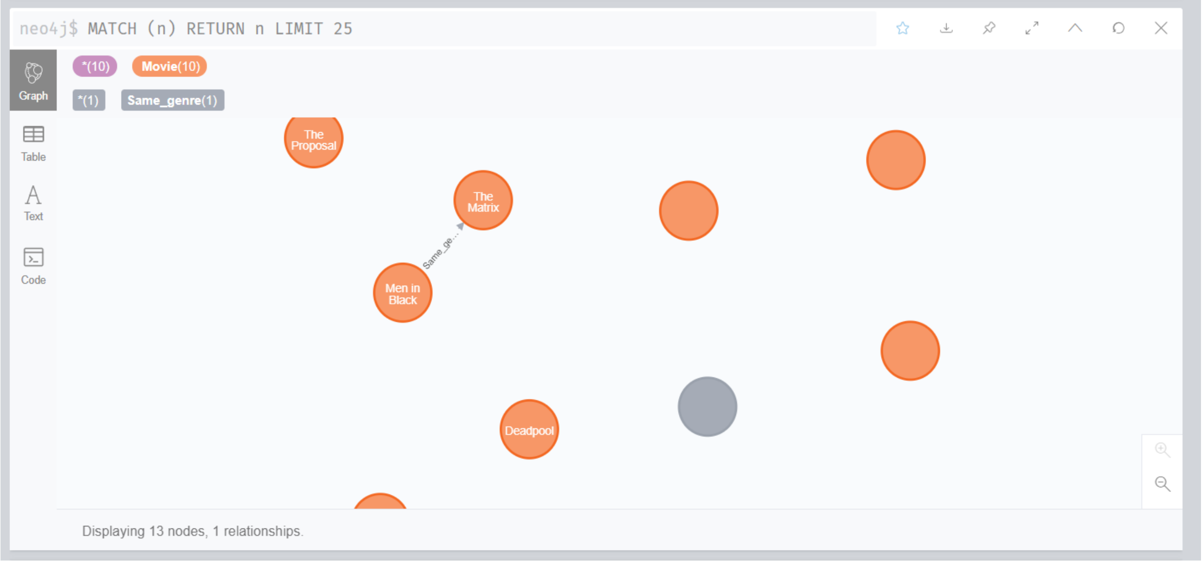
Syntax: Delete the relationship Match(n) detach (n)



```
neo4j$ MATCH (n:Actor) detach delete (n)
```

```
Server version    Neo4j/4.3.3
Server address    a011d2751daa1dc4e8895505d9d86e86.neo4jsandbox.com:7687
Query             MATCH (n:Actor) detach delete (n)
Summary ▸         {, "query": {, "text": "MATCH (n:Actor) detach delete (n)", ...
Response ▸        [] ...
```

Deleted 6 nodes, deleted 10 relationships, completed after 51 ms.

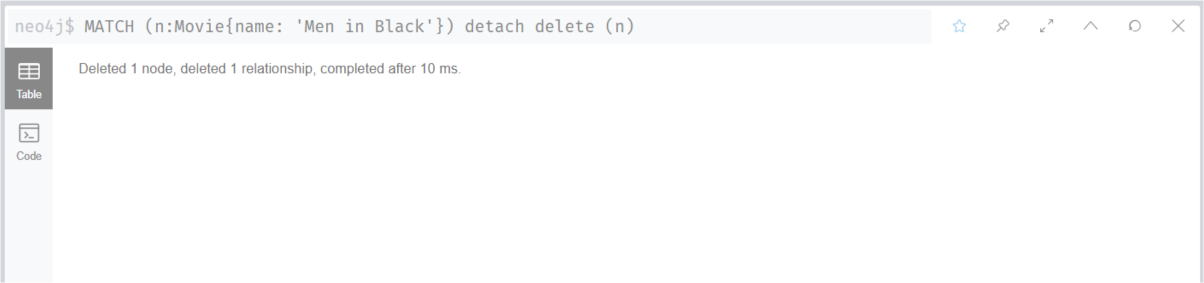


```
neo4j$ MATCH (n) RETURN n LIMIT 25
```

Displaying 13 nodes, 1 relationships.

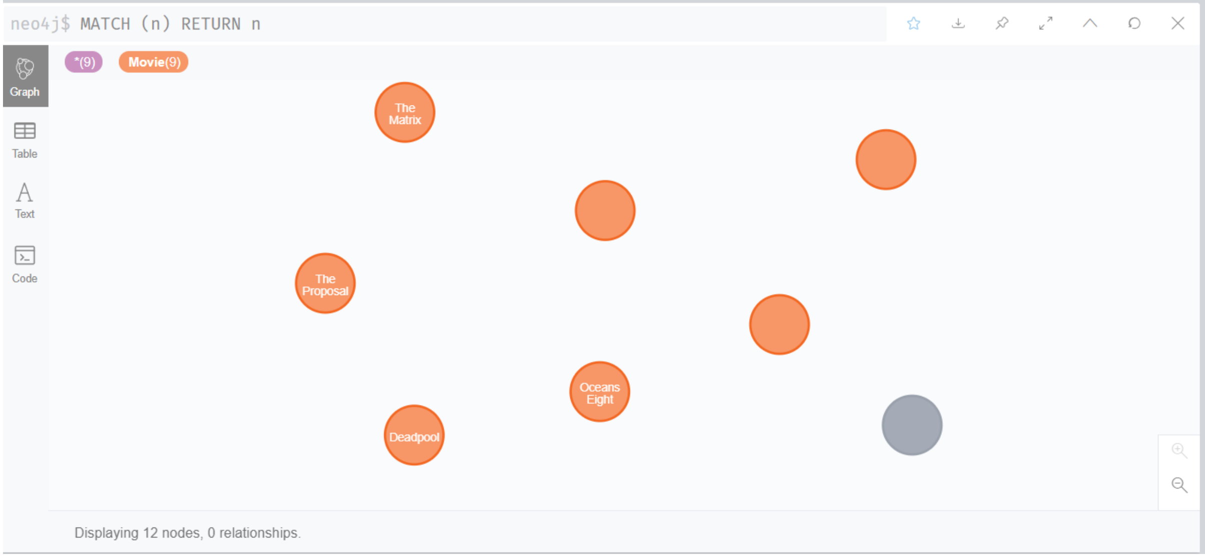
13. *Delete single node*

Syntax: match(filter) delete (n)



```
neo4j$ MATCH (n:Movie{name: 'Men in Black'}) detach delete (n)
```

```
Deleted 1 node, deleted 1 relationship, completed after 10 ms.
```



```
neo4j$ MATCH (n) RETURN n
```

```
Displaying 12 nodes, 0 relationships.
```