Exercise 8 :

Exploring Neo4j Aura: Creating and Manipulating Graph Data

Aim:

- To familiarize with Neo4j Aura, a cloud-based graph database platform.
- To learn how to create and manipulate nodes and relationships in Neo4j Aura using Cypher queries.

Algorithm:

- Creating and Displaying a Single Node
- Create a node with no specific label or properties and name it 'sample'.
- Use a MATCH query to find and return all nodes in the database, which at this point should only return the 'sample' node.
- Deleting a Specified Node
- Use a MATCH query to locate the previously created 'sample' node by its name or identifying characteristic.
- Delete the identified 'sample' node from the database.
- Creating and Displaying Multiple Nodes

Individually create two nodes named 'Raja' and 'Ravi' without specifying any labels or properties.

Retrieve and display all nodes, which will include 'Raja' and 'Ravi'.

- Deleting Nodes by IDs

Identify and delete the node with ID 1 (assumed to be 'Raja').

Similarly, find and delete the node with ID 2 (assumed to be 'Ravi').

- Deleting All Created Nodes

Match all existing nodes in the database regardless of their labels or properties.

- Delete all nodes matched in the previous step, effectively clearing the database.

Creating a Node with a Label and Displaying It.

CODE:

Query Operations

1. Create a single node named 'sample' without a caption and display it.

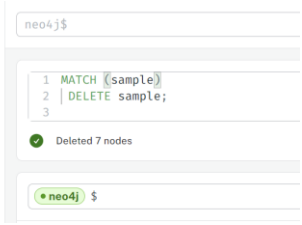
CREATE (sample);

MATCH (n) RETURN n;

2. Delete the single node created with the name 'sample'.

MATCH (sample);

DELETE sample;



3. Create two single nodes named 'Raja' and 'Ravi' without a caption and display them.

CREATE (Raja);

CREATE (Ravi);

MATCH (n) RETURN n;

4. Delete the nodes created above using their IDs.

MATCH (n)

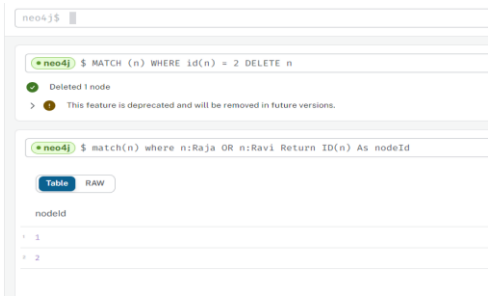
WHERE id(n) = 1

DELETE n

MATCH (n)

WHERE id(n) = 2

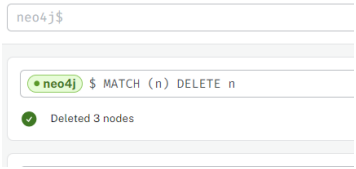
DELETE n



5. Delete all nodes created so far.

MATCH (n)

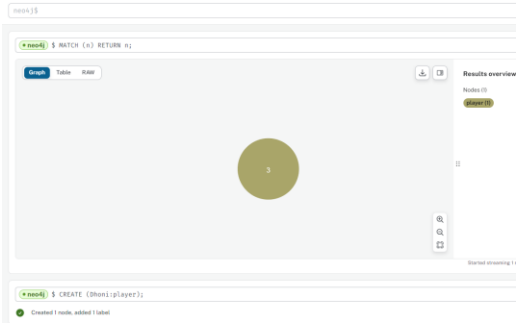
DELETE n



6. Create a single node named 'Dhoni' with the caption 'player' and display it.

CREATE (Dhoni:player);

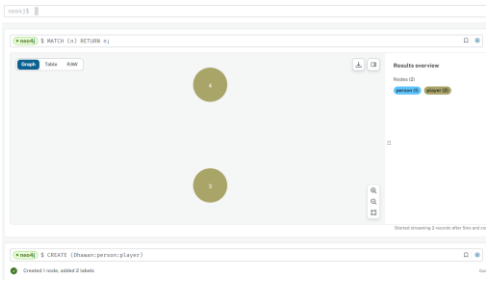
MATCH (n) RETURN n;



7. Create a node named 'Dhawan' with multiple labels 'person' and 'player' and display it.

CREATE (Dhawan:person:player)

MATCH (n) RETURN n;



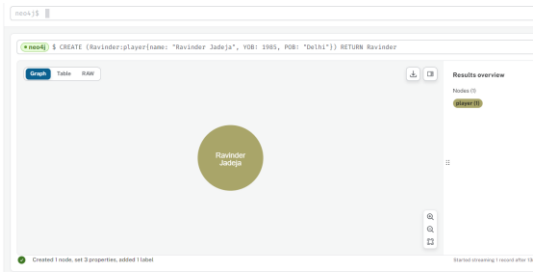
8. Create a node named 'Sachin' with the caption 'player' and properties like name, year of birth (YOB), and place of birth (POB).

CREATE (Sachin:player{name: "Sachin Tendulkar", YOB: 1985, POB: "Delhi"})

MATCH (n) RETURN n;

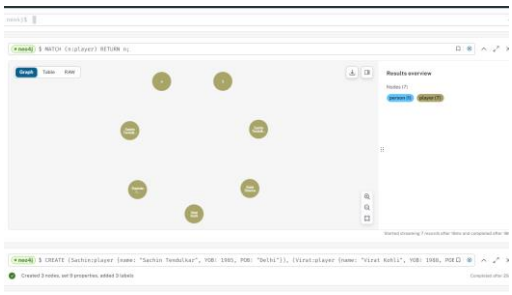
9. Create a node named 'Ravinder' with the caption 'player' and properties like name, year of birth (YOB), and place of birth (POB), and return the node without using the MATCH statement.

CREATE (Ravinder:player{name: "Ravinder Jadeja", YOB: 1985, POB: "Delhi"}) RETURN Ravinder



10. Create multiple nodes with properties like name, year of birth (YOB), and place of birth (POB).

```
CREATE (Sachin:player {name: "Sachin Tendulkar", YOB: 1985, POB: "Delhi"}),
    (Virat:player {name: "Virat Kohli", YOB: 1988, POB: "Delhi"}),
    (Rohit:player {name: "Rohit Sharma", YOB: 1987, POB: "Nagpur"});
MATCH (n) RETURN n
```



11. Create nodes 'Dhawan' and 'Ind' and establish a relationship 'BATSMAN_OF' between them.

CREATE (Dhawan:player {name: "Shikar Dhawan", YOB: 1985, POB: "Delhi"})

CREATE (Ind:Country {name: "India"})

CREATE (Dhawan)-[r:BATSMAN_OF]->(Ind)

12. Establish a relationship between the player node labeled 'Shikar Dhawan' and the country node labeled 'India'. Include details such as the number of matches played (5) and the batting average (90.75).

MATCH (player:player {name: "Shikar Dhawan"}), (country:Country {name: "India"})

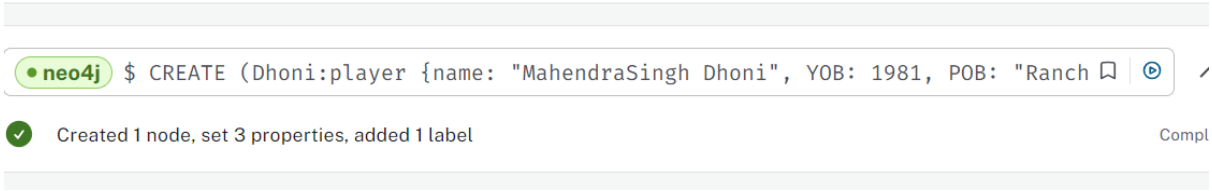
CREATE (player)-[r:BATSMAN_OF {Matches: 5, Avg: 90.75}]->(country)

RETURN player, r, country;

13. Create a player node for Mahendra Singh Dhoni, born in Ranchi in 1981. Describe the characteristics of the newly created node and then remove the property 'POB' and label 'player' using the REMOVE statement.

Step :1 (Create Node)

CREATE (Dhoni:player {name: "MahendraSingh Dhoni", YOB: 1981, POB: "Ranchi"})



Step :2 (Remove Property)

MATCH (Dhoni:player {name: "MahendraSingh Dhoni", YOB: 1981, POB: "Ranchi"})

REMOVE Dhoni.POB

RETURN Dhoni

Step: 3 (Remove Label)

MATCH (Dhoni:player {name: "MahendraSingh Dhoni", YOB: 1981, POB: "Ranchi"})

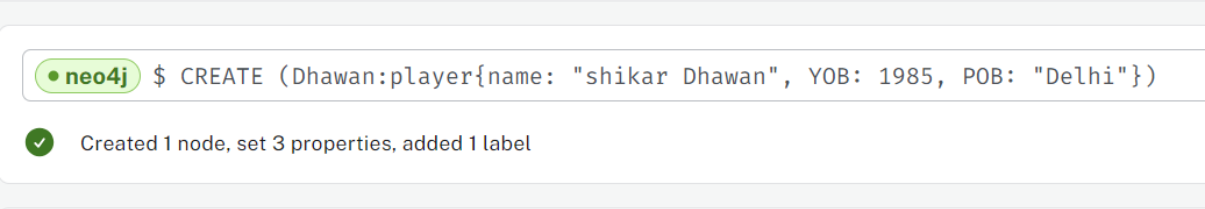
REMOVE Dhoni:player

RETURN Dhoni

14. Create a player node for Shikhar Dhawan, born in Delhi in 1985. After creating the player node, find the node representing Shikhar Dhawan, update its property to reflect his highest score of 187, and return the updated node.

Step :1

CREATE (Dhawan:player{name: "shikar Dhawan", YOB: 1985, POB: "Delhi"})

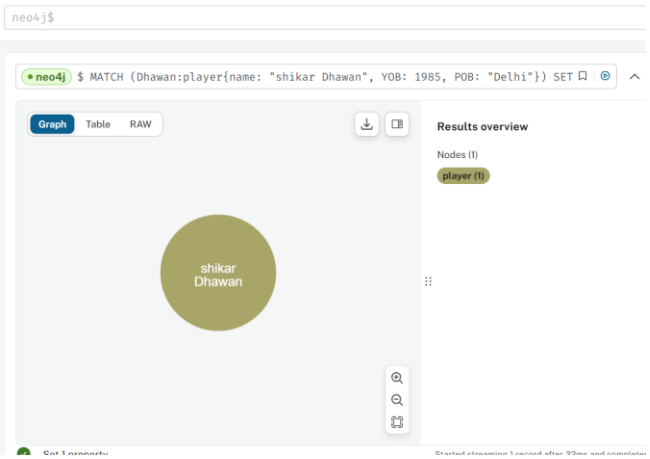


Step :2

MATCH (Dhawan:player{name: "shikar Dhawan", YOB: 1985, POB: "Delhi"})

SET Dhawan.highestscore = 187

RETURN Dhawan



Result:

Here comprehensive overview of operations for creating, displaying, deleting, and modifying nodes in a Neo4j database is Executed and Output is verified.