Create Node with properties:-

properties key-value pairs using which a node stores data. create a node with properties using CREATE clause and need to specify these properties separated by commas within flower braces "{}".

Syntax

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

To verify the creation of node, type and execute the following query in dollar prompt.

Syntax: MATCH (n) RETURN n.

Creating Relationships:-

To create a relationship using CREATE clause and specify relationship within the square braces "[]" depending on direction of relationship it is placed b/w hyphen "-" "arrow" ->.

Syntax: CREATE (node1)-[Relationship type]->(node2).

Syntax:

~~MATCH~~
MATCH (a:Label of Node1), (b:Label of node2).
CREATE (a)-[:Relation]->(b) RETURN a, b.

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

Create a cricket board Node:

Create (Cb:cricket Board {BoardID: "BID01"; "Name: chennai cricket Board";
Address: 'chennai', phone: 9887766944}) return cb.

Create Team Nodes:

Create (t1:Team {teamID: 'CC B01', Name: Raj: AUSExpress;
coach (ci AAmesh'; captain: 'SAMAPATH KUMAR} return t1

Create (t2 Team {teamID: "CC B02", Board: B1D01: name: 'Ava Express coach:
J. KARTHIK; captain: Y JOHN}) return t2.

22/4/25 TASK 9 : CRUD operations in graph Database.

Aim: To perform CRUD operations like creating, inserting querying finding, deleting operations on graph spaces.

The steps to get with Neo4j's Aura graph Database:-

Step1: Copy and paste the following link into your web browser.
https://neo4j.com/cloud/platform/aura-graph-database/?ref=docs-get-started-dropdown

Step2: Click on "Start Free".

Step3: chose the option to "continue with google.

Step4: click the "open" button.

Step5: After clicking "open", a text file will be automatically downloaded. this file contains your user ID and password details.

Step6: close the get started with Neo4j with beginner guides if it's open.

Step7: You're now ready to begin practicing with graph Database.

Create player nodes:

Create (P1 player {player ID: '1' team ID: 'CCB01', Name: 'Raj', Age: 23, Date of Birth: '24-Jun-1998', playing role: 'Bowler', email: raj@gmail.com}), return P1

Create (P2 player {player ID: '33' team ID: 'CCB01', Name: 'Arun', Age: 23, Date of Birth: '08-Jan-1999', playing role: 'Batsman', email: balaje@gmail.com}), return P2

Creating Relationship among cricketboard and team:

match (cb: cricketboard {Board ID: 'BID01'}), (t1: team {team ID: 'CCB01'})
Create (cb)-[r1: has]->(t1) return cb, r1, t1.

match (cb: cricketboard {Board ID: 'BID01'}), (t2: team {team ID: 'CCB01'})
Create (cb)-[r2: has]->(t2) return cb, r2, t2

Creating relationship among players and team:

match (P1: player {player ID: '1'}) (t1: team {team ID: 'CCB01'}) create (P1)-[r1: player]->(t1) return P1, t1, r1

match (P2: player {player ID: '33'}), (t1: team {team ID: 'CCB01'}) create (P2)-[r2: player]->(t1) return P2, r2, t2

VEL TECH - CSE	
EX NO.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	

Result: thus, the CRUD operations (creating, inserting, querying, finding, deleting) operations on graph space executed successfully.