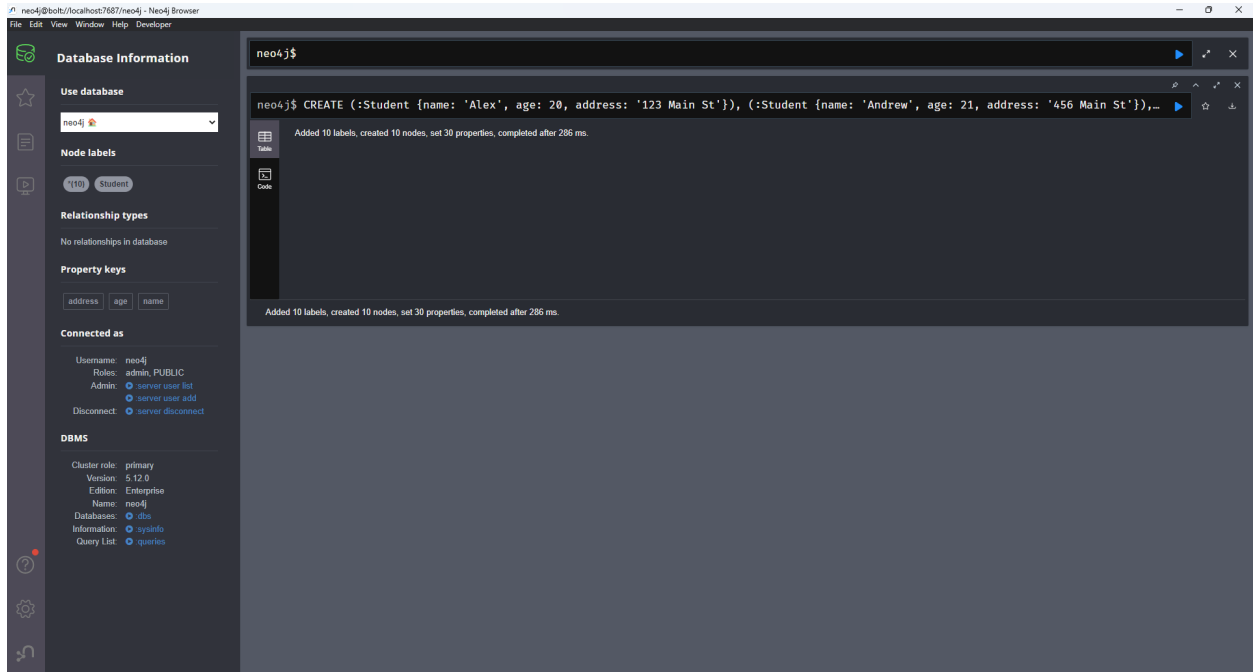


HomeWork8_AparnaBharathi_Suresh

Question 1:

Create 10 students (Alex, Andrew, Arnold, Ally, Bob, Brad, Bran, Chris, Charles, David) nodes with names, ages, and addresses. Display all nodes

Screenshot 1:





Query 1:

```
CREATE (:Student {name: 'Alex', age: 20, address: '123 Main St'}),  
      (:Student {name: 'Andrew', age: 21, address: '456 Main St'}),  
      (:Student {name: 'Arnold', age: 22, address: '789 Main St'}),  
      (:Student {name: 'Ally', age: 19, address: '321 Main St'}),  
      (:Student {name: 'Bob', age: 20, address: '234 Main St'}),  
      (:Student {name: 'Brad', age: 21, address: '567 Main St'}),  
      (:Student {name: 'Bran', age: 22, address: '890 Main St'}),  
      (:Student {name: 'Chris', age: 19, address: '432 Main St'}),  
      (:Student {name: 'Charles', age: 20, address: '345 Main St'}),  
      (:Student {name: 'David', age: 21, address: '678 Main St'})  
  
MATCH (n) RETURN n
```

Question 2:

Display min age, max-age, and average age among all students.

Screenshot 2:

The screenshot displays the Neo4j Desktop interface. On the left, the 'Database Information' sidebar shows the 'neo4j' database selected. The 'Node labels' section lists 'Student' with 110 nodes. The 'Relationship types' section shows no relationships. The 'Property keys' section lists 'address', 'age', and 'name'. The 'Connected as' section shows the user 'neo4j' with roles 'admin, PUBLIC'. The 'DBMS' section shows the cluster role 'primary', version '5.12.0', edition 'Enterprise', name 'neo4j', and databases 'db', 'system', and 'queries'.

The main window shows two query results. The top query is `neo4j$ MATCH (s:Student) RETURN MIN(s.age) AS MinAge, MAX(s.age) AS MaxAge, AVG(s.age) AS AvgAge`. The result is a table with three columns: MinAge, MaxAge, and AvgAge. The values are 19, 22, and 20.5 respectively. The bottom query is `neo4j$ MATCH (n) RETURN n`. The result is a graph visualization showing 10 nodes, all labeled 'Student'. The nodes are connected by lines, representing relationships. The graph is displayed in a dark theme.

MinAge	MaxAge	AvgAge
19	22	20.5

Started streaming 1 records after 1 ms and completed after 1 ms.

Overview

Node labels

Student (110)

Displaying 10 nodes, 0 relationships.

Query 2:

`MATCH (s:Student) RETURN MIN(s.age) AS MinAge, MAX(s.age) AS MaxAge, AVG(s.age) AS AvgAge`

Question 3:

Display only Brad node.

Screenshot 3:

The screenshot displays the Neo4j Browser interface. On the left, the 'Database Information' sidebar shows the database name 'neo4j', node labels, relationship types, and property keys. The main workspace shows a Cypher query: `neo4j$ MATCH (s:Student{name : "Brad"}) RETURN s`. The result is a single node labeled 'Brad'. On the right, the 'Node properties' panel shows the properties for the 'Student' node: `<id>` 5, `address` 567 Main St, `age` 21, and `name` Brad.

Database Information

Use database: neo4j

Node labels: Student

Relationship types: No relationships in database

Property keys: address, age, name

Connected as: Username: neo4j, Roles: admin, PUBLIC, Admin: server user list, server user add, server disconnect

DBMS: Cluster role: primary, Version: 5.12.0, Edition: Enterprise, Name: neo4j, Databases: dba, sysinfo, queries

neo4j\$ MATCH (s:Student{name : "Brad"}) RETURN s

Node properties

Student

<id> 5

address 567 Main St

age 21

name Brad

neo4j\$ MATCH (s:Student) RETURN MIN(s.age) AS MinAge, MAX(s.age) AS MaxAge, AVG(s.age) AS AvgAge

	MinAge	MaxAge	AvgAge
1	19	22	20.5

Started streaming 1 records after 1 ms and completed after 1 ms.

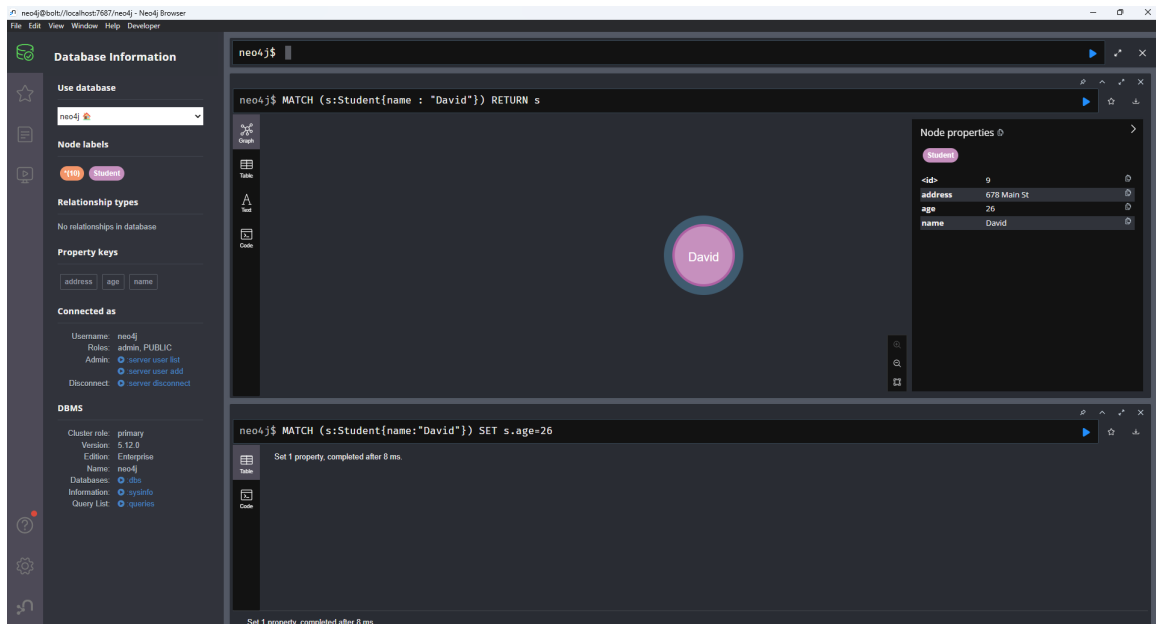
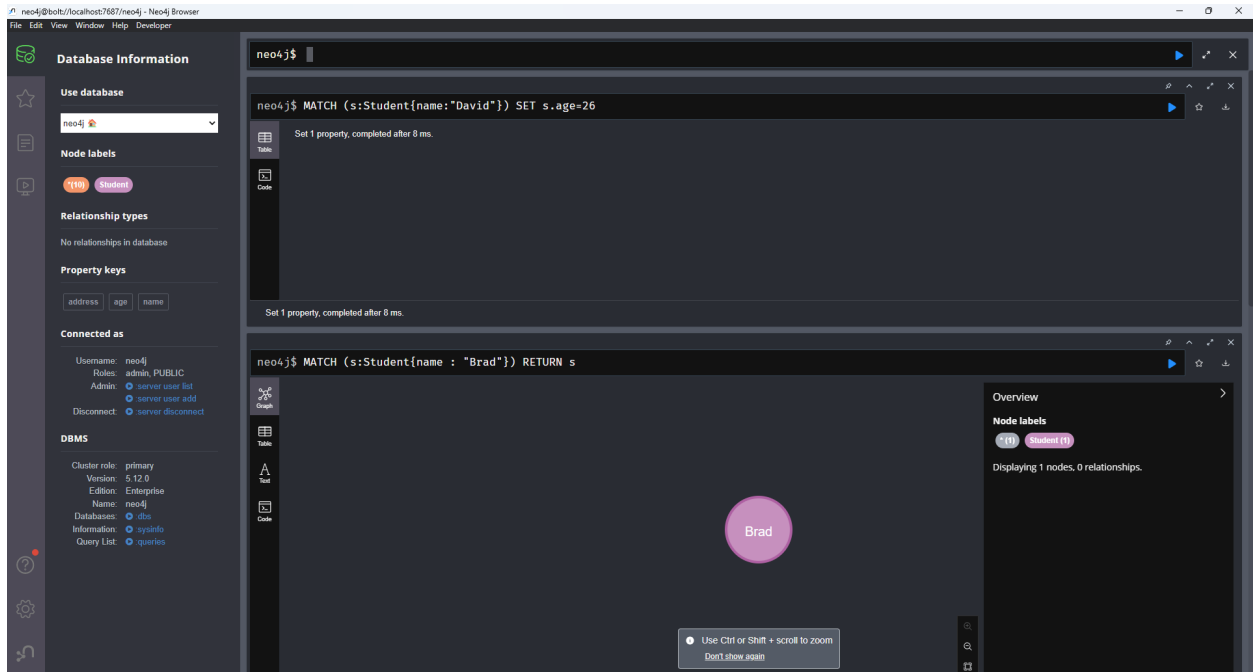
Query 3:

`MATCH (s:Student{name : "Brad"}) RETURN s`

Question 4:

Update age of David to be 26

Screenshot 4:



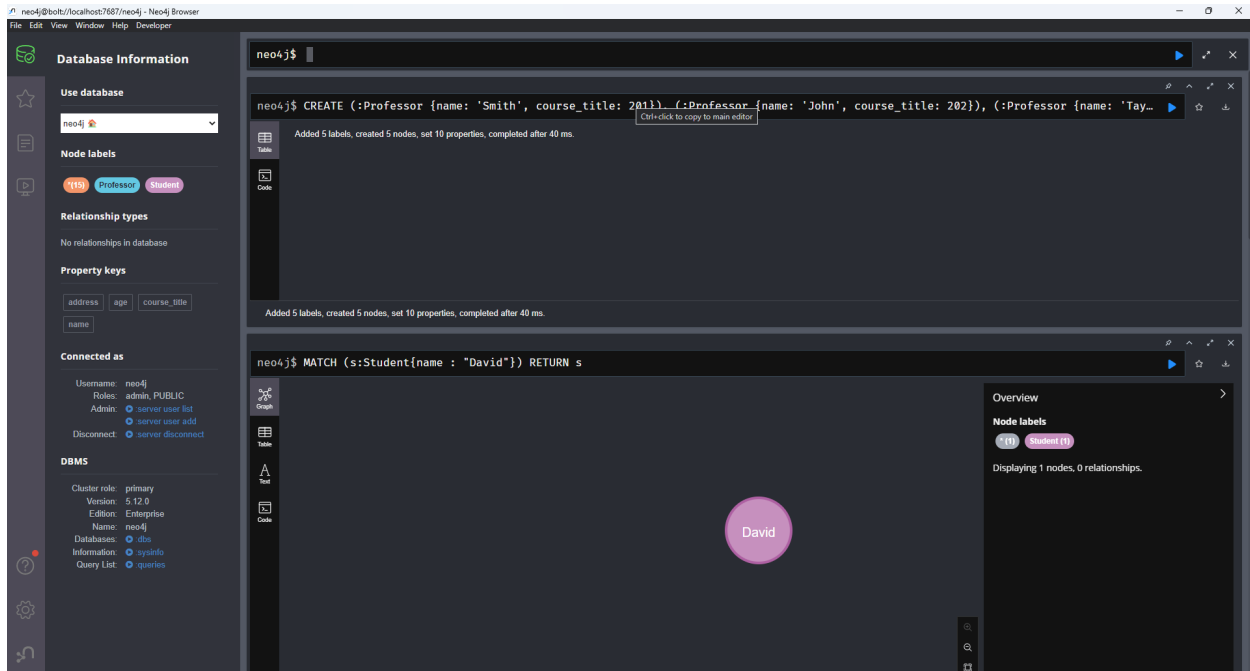
Query 4:

`MATCH (s:Student{name:"David"}) SET s.age=26`

Question 5:

Create 5 Professor (Smith, John, Taylor, James, Tim) nodes with name, and course title (201,202,203,204,205). Display all professor nodes.

Screenshot 5:



Query 5:

```
CREATE (:Professor {name: 'Smith', course_title: 201}),  
      (:Professor {name: 'John', course_title: 202}),  
      (:Professor {name: 'Taylor', course_title: 203}),  
      (:Professor {name: 'James', course_title: 204}),  
      (:Professor {name: 'Tim', course_title: 205})  
  
MATCH (n:Professor) RETURN n
```

Question 6:

Create 'takes_course' relationships (eg; **from:** Semester) between all students and any professor. (Make sure each student takes at least 3 courses) (Eg: Arnold takes_course John) (Arnold takes_course Tim)

Screenshot 6:

The screenshot displays the Neo4j Browser interface. On the left, the 'Database Information' sidebar shows the following details:

- Use database:** neo4j
- Node labels:** 115 (Professor), 115 (Student)
- Relationship types:** 30 (takes_course)
- Property keys:** address, age, course_title, name
- Connected as:** Username: neo4j, Roles: admin, PUBLIC, Admin: server user list, server user add, server disconnect, Disconnect: server disconnect
- DBMS:** Cluster role: primary, Version: 5.12.0, Edition: Enterprise, Name: neo4j, Databases: 0, Information: sysinfo, sysinfo, Query List: queries

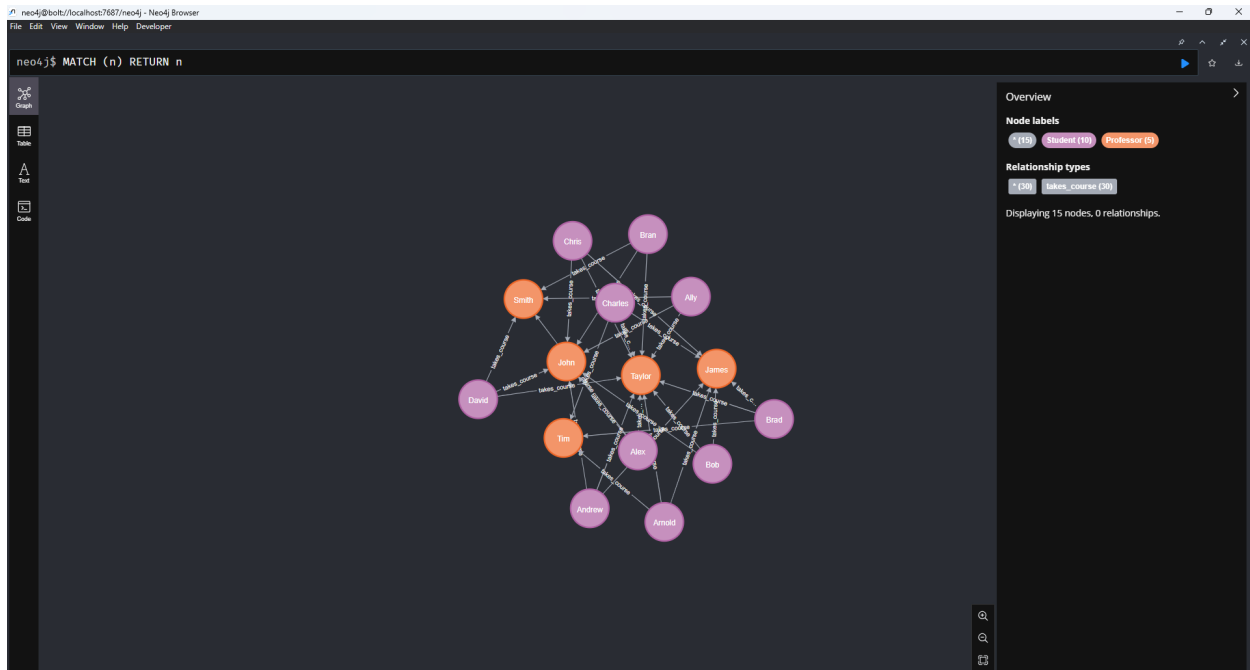
The main query editor shows the following Cypher query:

```
neo4j$ MATCH (s1:Student {name: 'Alex'}), (s2:Student {name: 'Andrew'}), (s3:Student {name: 'Arnold'}), (s4:Student {name: 'Ally'}), (s5:Student {name: 'Tim'})
```

Below the query, it states: 'Created 30 relationships, completed after 1147 ms.'

The graph view shows a network of nodes. The nodes are labeled with their names and roles: Alex (Student), Andrew (Student), Arnold (Student), Ally (Student), Tim (Student), John (Professor), and Tim (Professor). The graph shows connections between students and professors.

The 'Overview' panel on the right indicates: 'Node labels: 115 (Student), 115 (Professor), 30 (takes_course). Displaying 15 nodes, 0 relationships.'



Query 6:

```
MATCH (s1:Student {name: 'Alex'}), (s2:Student {name: 'Andrew'}),(s3:Student {name: 'Arnold'}),(s4:Student {name: 'Ally'}),(s5:Student {name: 'Bob'}),(s6:Student {name: 'Brad'}),(s7:Student {name: 'Bran'}),(s8:Student {name: 'Chris'}),(s9:Student {name: 'Charles'}),(s10:Student {name: 'David'}), (p1:Professor {name: 'Smith'}),(p2:Professor {name: 'John'}),(p3:Professor {name: 'Taylor'}),(p4:Professor {name: 'James'}),(p5:Professor {name: 'Tim'})
```

```
CREATE (s1)-[:takes_course]->(p1),(s1)-[:takes_course]->(p2),(s1)-[:takes_course]->(p3),
      (s2)-[:takes_course]->(p2),(s2)-[:takes_course]->(p3),(s2)-[:takes_course]->(p4),
      (s3)-[:takes_course]->(p3),(s3)-[:takes_course]->(p4),(s3)-[:takes_course]->(p5),
      (s4)-[:takes_course]->(p1),(s4)-[:takes_course]->(p2),(s4)-[:takes_course]->(p3),
      (s5)-[:takes_course]->(p2),(s5)-[:takes_course]->(p3),(s5)-[:takes_course]->(p4),
      (s6)-[:takes_course]->(p3),(s6)-[:takes_course]->(p4),(s6)-[:takes_course]->(p5),
      (s7)-[:takes_course]->(p1),(s7)-[:takes_course]->(p2),(s7)-[:takes_course]->(p3),
      (s8)-[:takes_course]->(p2),(s8)-[:takes_course]->(p3),(s8)-[:takes_course]->(p4),
      (s9)-[:takes_course]->(p3),(s9)-[:takes_course]->(p4),(s9)-[:takes_course]->(p5),
      (s10)-[:takes_course]->(p1),(s10)-[:takes_course]->(p2),(s10)-[:takes_course]->(p3)
```

Question 7:

Display courses Ally took.

Screenshot 7:

The screenshot shows the Neo4j Browser interface. On the left is a sidebar with 'Database Information', 'Use database', 'Node labels', 'Relationship types', 'Property keys', 'Connected as', and 'DBMS'. The main area displays a query and its results.

Query:

```
neo4j$ MATCH (s:Student{name:"Ally"})-[takes_course]->(p:Professor) RETURN p.course_title AS Course
```

Results:

Course
203
202
201

Started streaming 3 records in less than 1 ms and completed in less than 1 ms.

Overview:

Node labels: 115 Student (115), Professor (2)

Relationship types: 1 DB, takes_course (2)

Displaying 15 nodes, 0 relationships.

Query 7:

MATCH

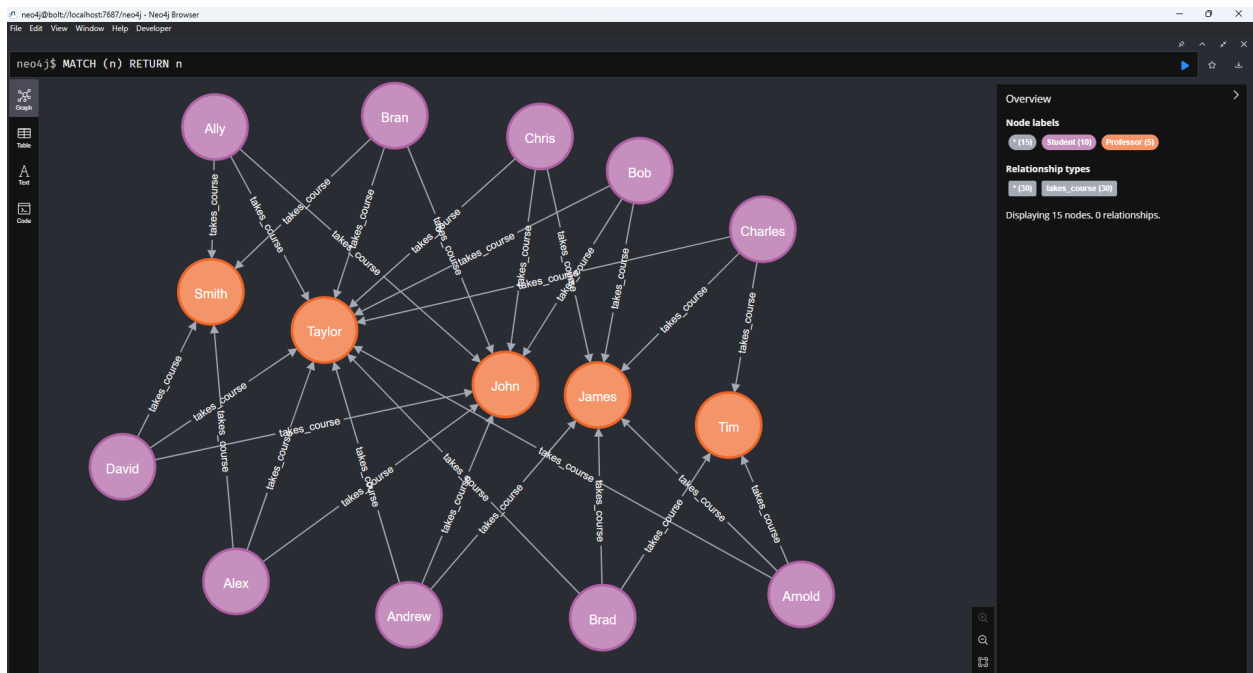
(s:Student{name:"Ally"})-[takes_course]->(p:Professor)

RETURN p.course_title AS Course

Question 8:

Display all 15 nodes with relationships

Screenshot 8:



Query 8:

`MATCH (n) RETURN n`

Question 9:

Delete nodes Bob, Bran, James

Screenshot 9:

The screenshot displays the Neo4j Browser interface. On the left sidebar, the 'Database Information' panel shows the current database as 'neo4j'. Under 'Node labels', 'Student' and 'Professor' are listed. Under 'Relationship types', 'takes_course' is listed. The 'Connected as' section shows the user 'neo4j' with roles 'admin' and 'PUBLIC'. The 'DBMS' section shows the cluster role as 'primary', version as '5.12.0', and edition as 'Enterprise'.

The main query editor shows the following Cypher query:

```
neo4j$ MATCH (n) WHERE n.name IN ['Bob', 'Bran', 'James'] DETACH DELETE n
```

The execution result shows: 'Deleted 3 nodes, deleted 11 relationships, completed after 41 ms.' Below this, a table titled 'Course' displays the following data:

	Course
1	203
2	202
3	201

The bottom panel shows a graph visualization with a node labeled 'Charles' and various relationships.

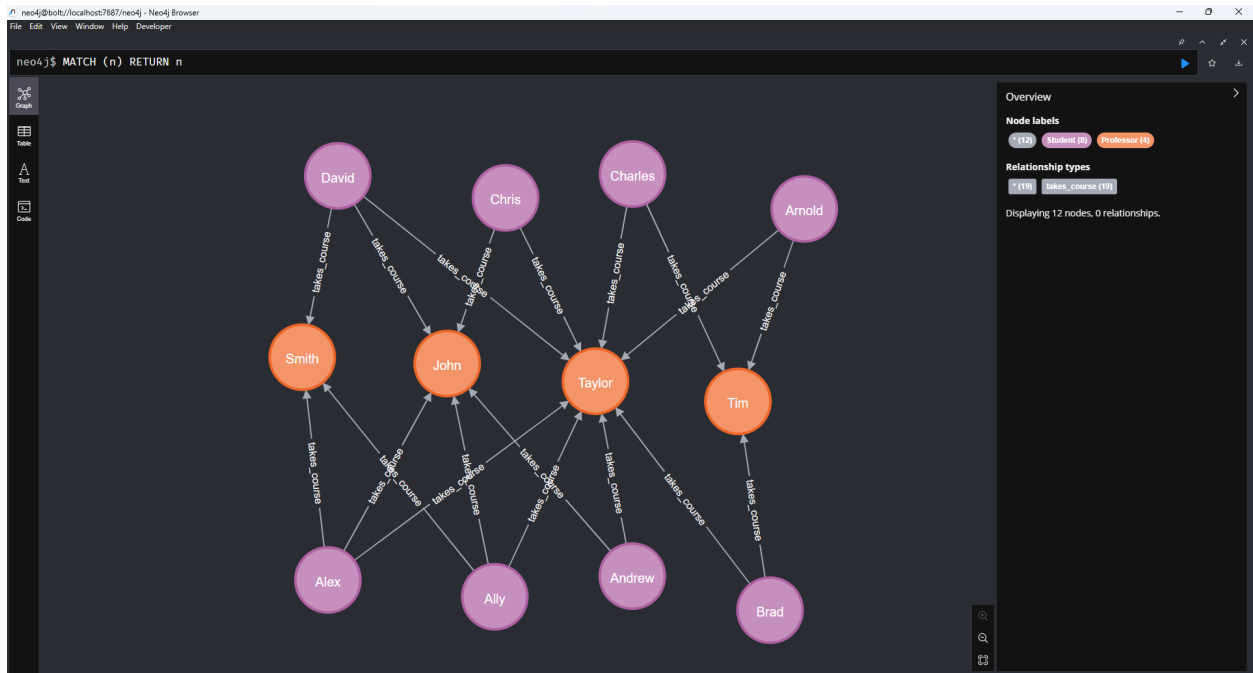
Query 9:

`MATCH (n) WHERE n.name IN ['Bob', 'Bran', 'James'] DETACH DELETE n`

Question 10:

Display the graph

Screenshot 10:



Query 10:

`MATCH (n) RETURN n`