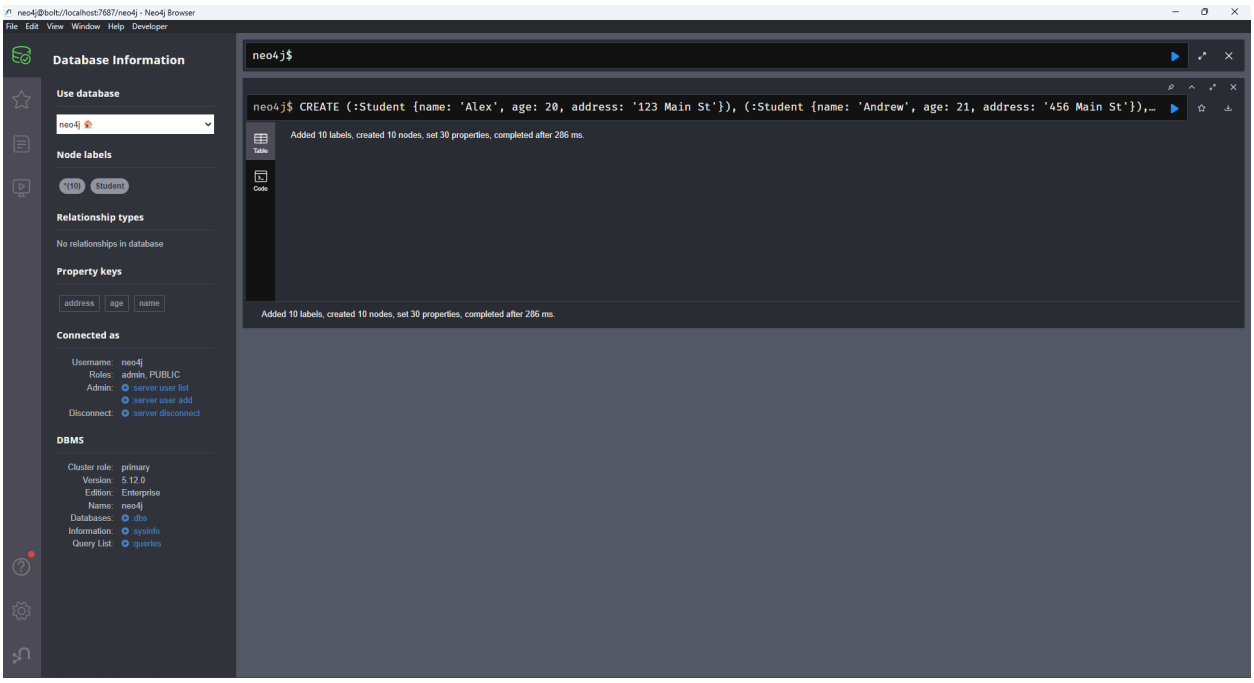
HomeWork8_AparnaBharathi_Suresh

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:



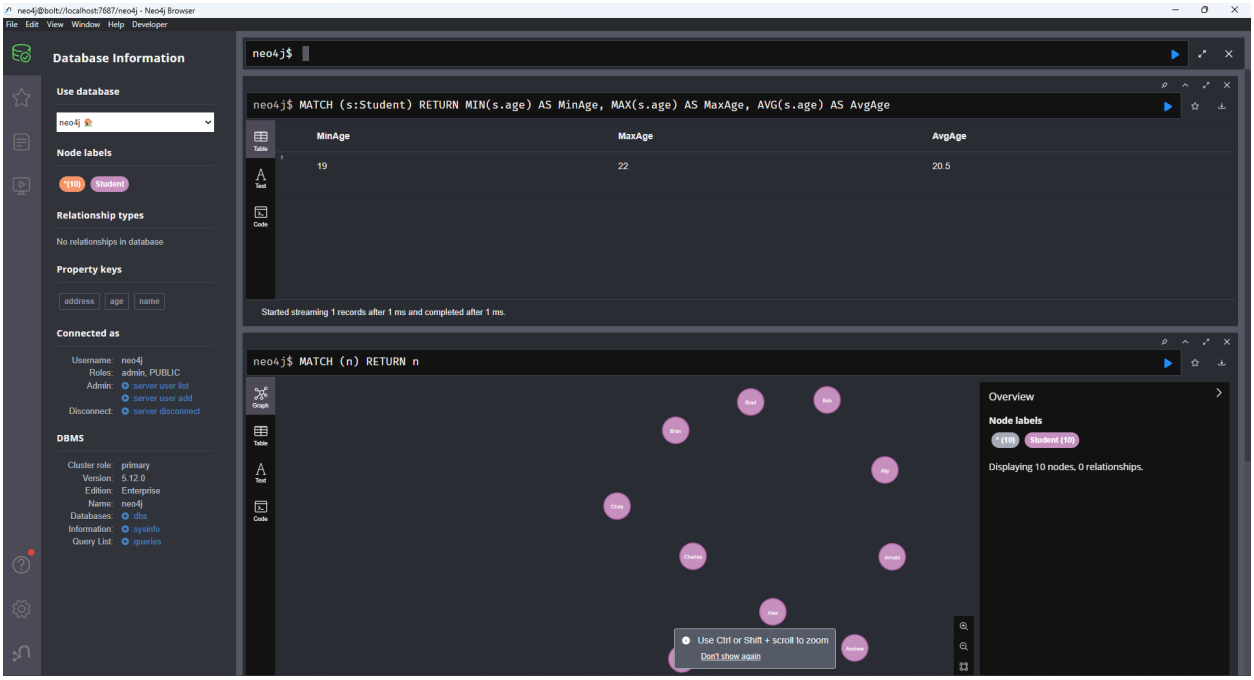
Query 1:

```
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:



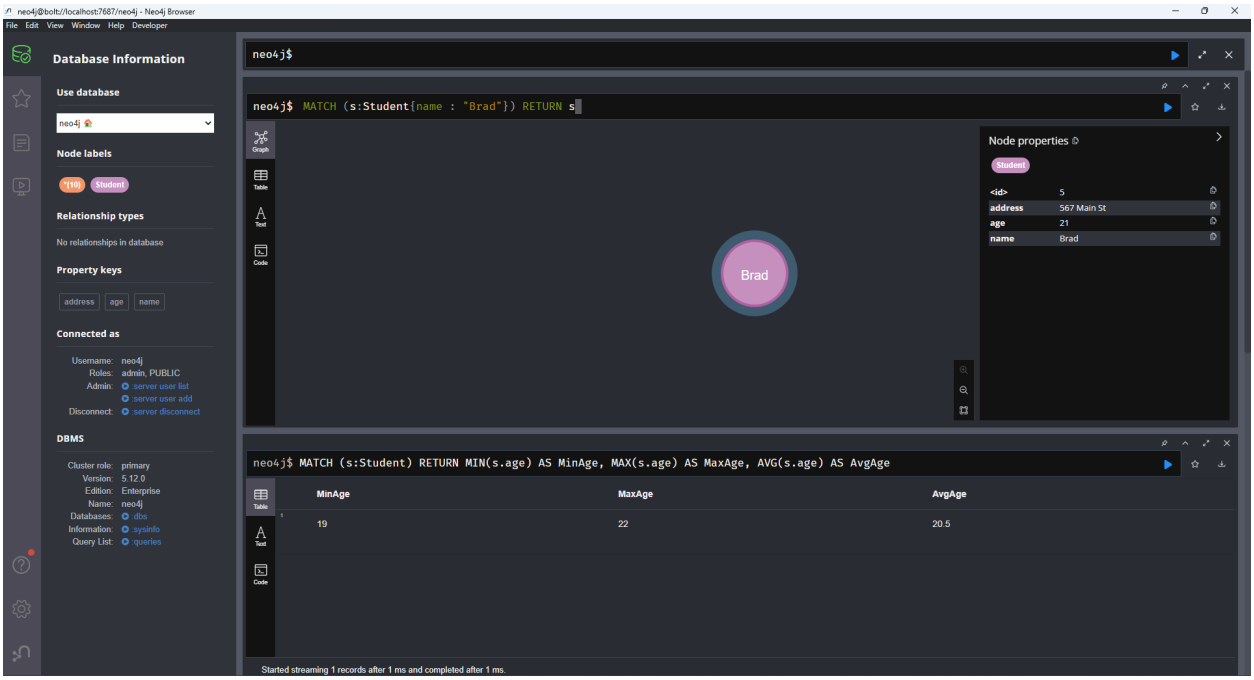
Query 2:

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:



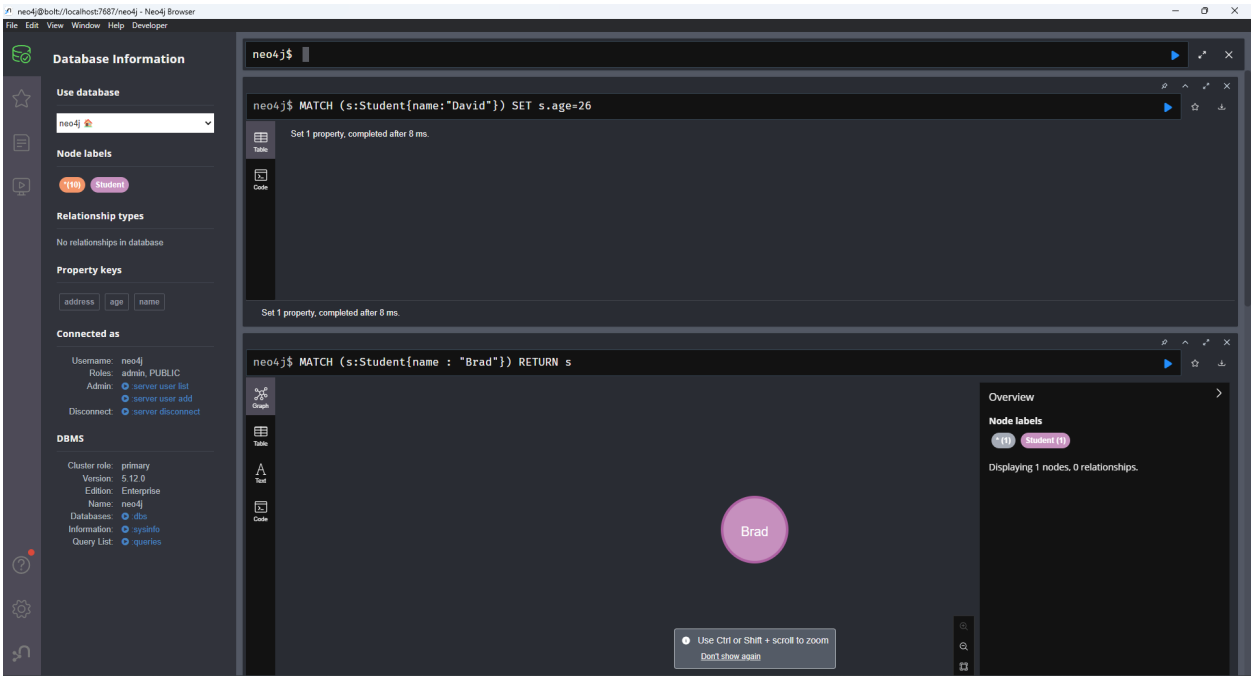
Query 3:

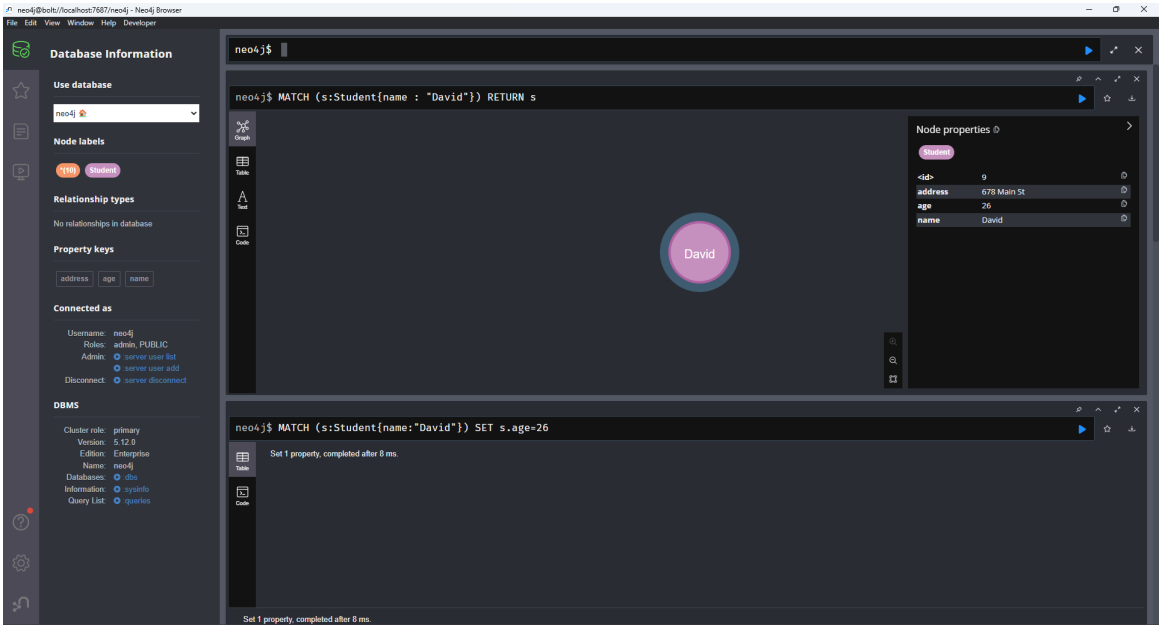
MATCH (s:Student{name : "Brad"}) RETURN s

Question 4:

Update age of David to be 26

Screenshot 4:





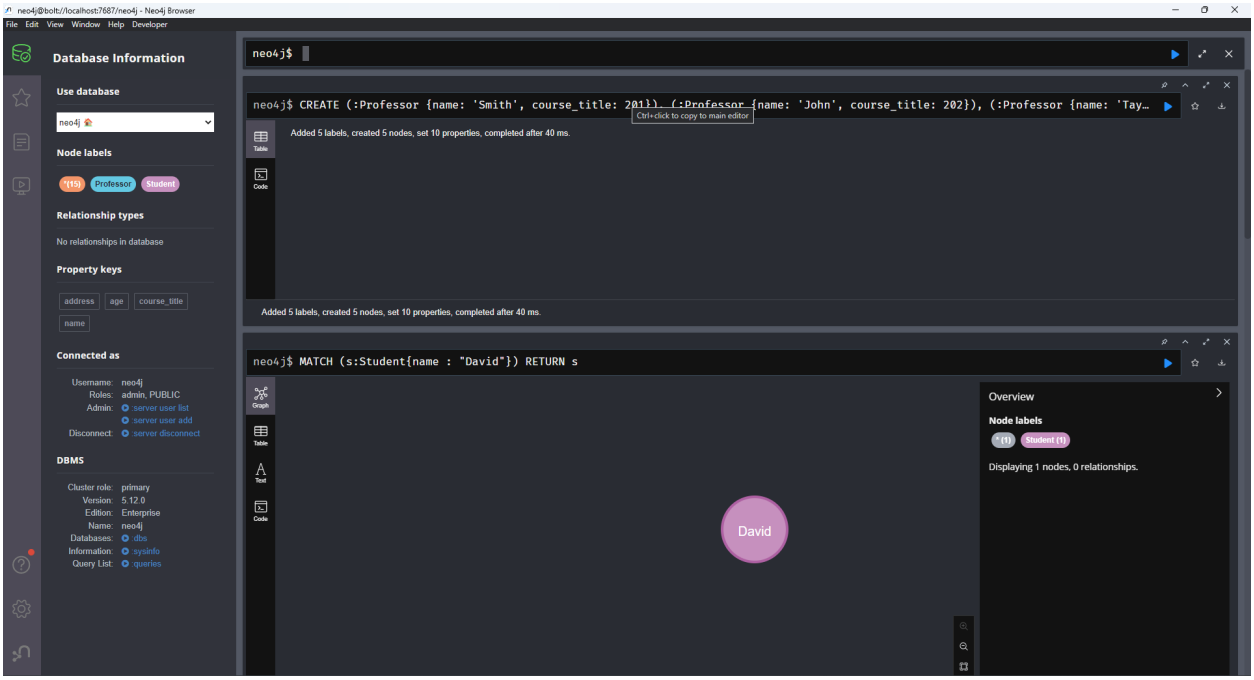
Query 4:

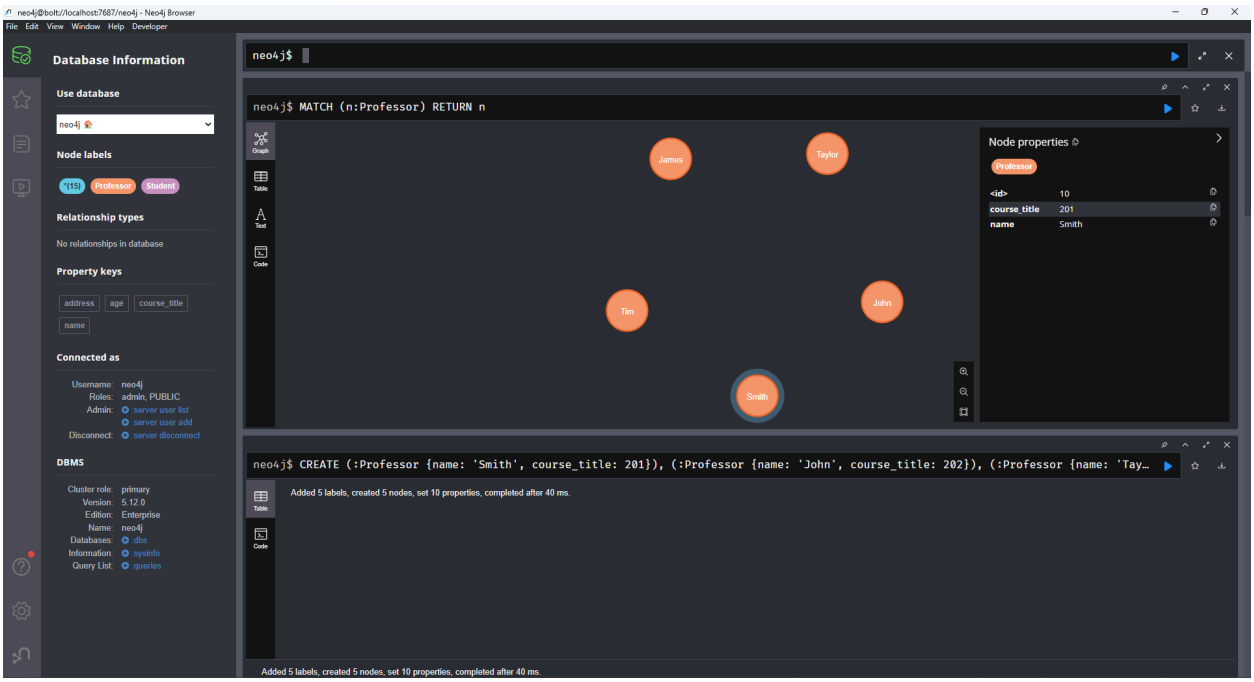
MATCH (s:Student{name:"David"}) SET s.age=26

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:

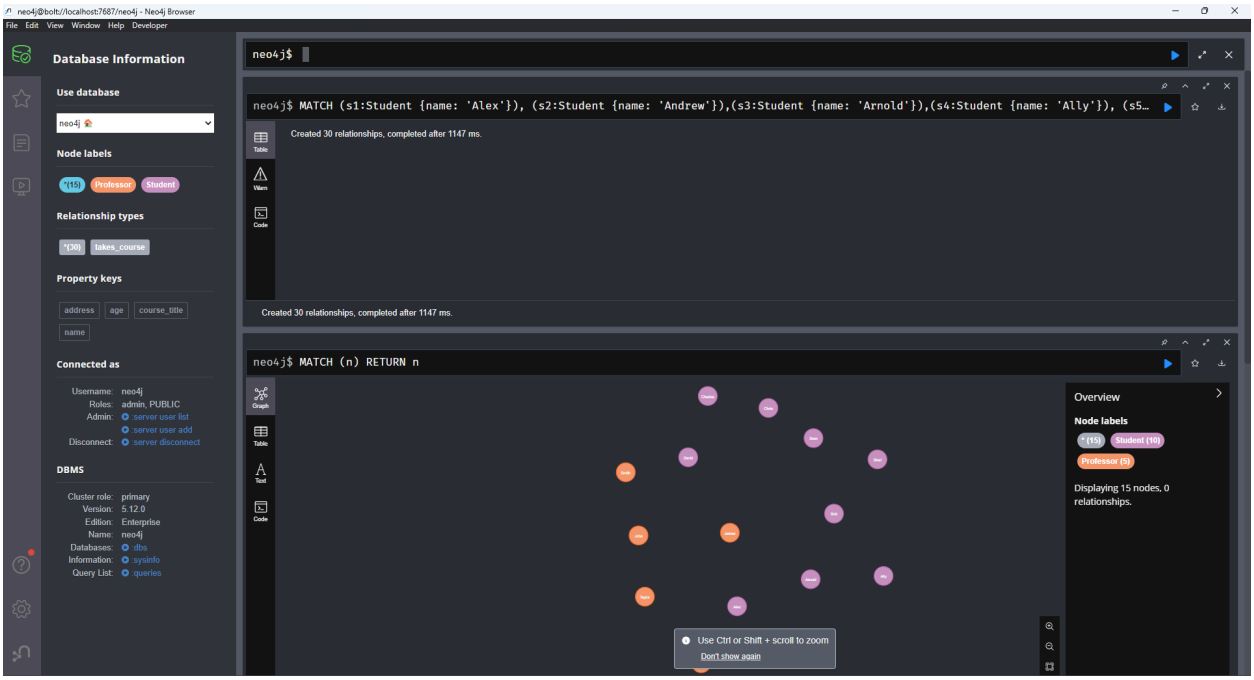
```
CREATE  (:Professor {name: 'Smith', course_title: 201}),
        (:Professor {name: 'John', course_title: 202}),
        (:Professor {name: 'Taylor', course_title: 203}),
        (:Professor {name: 'James', course_title: 204}),
        (:Professor {name: 'Tim', course_title: 205})
```

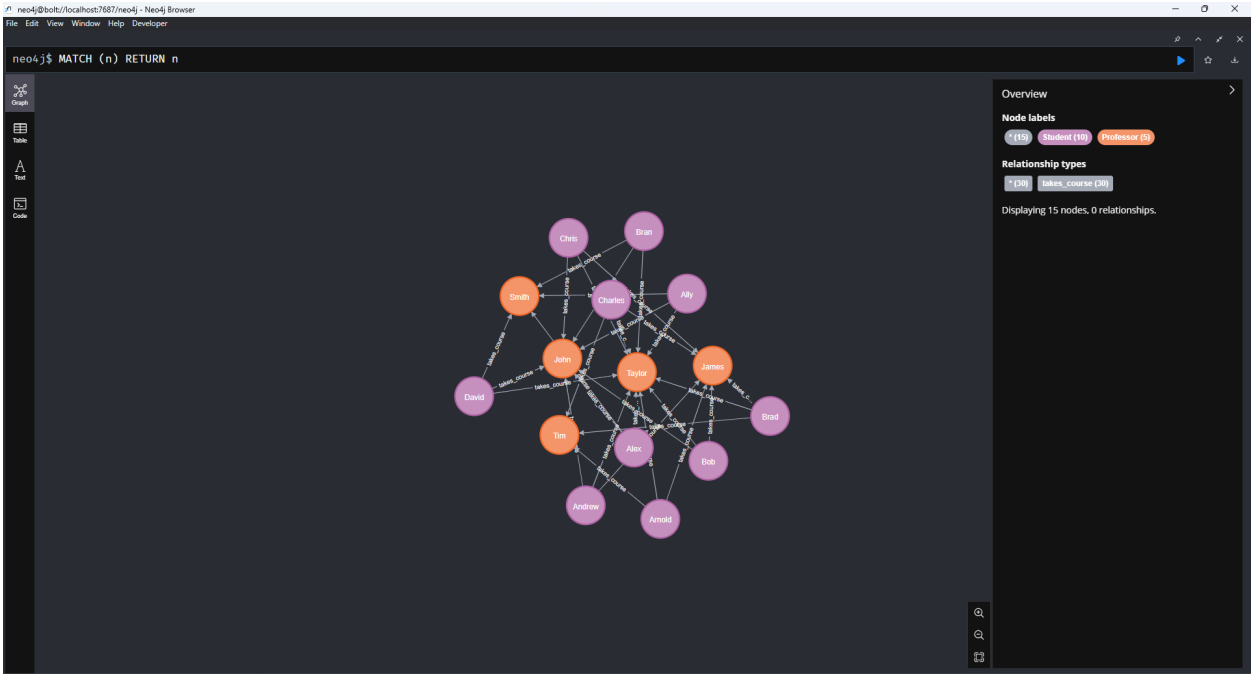
```
MATCH (n:Professor) RETURN n
```

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:





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),

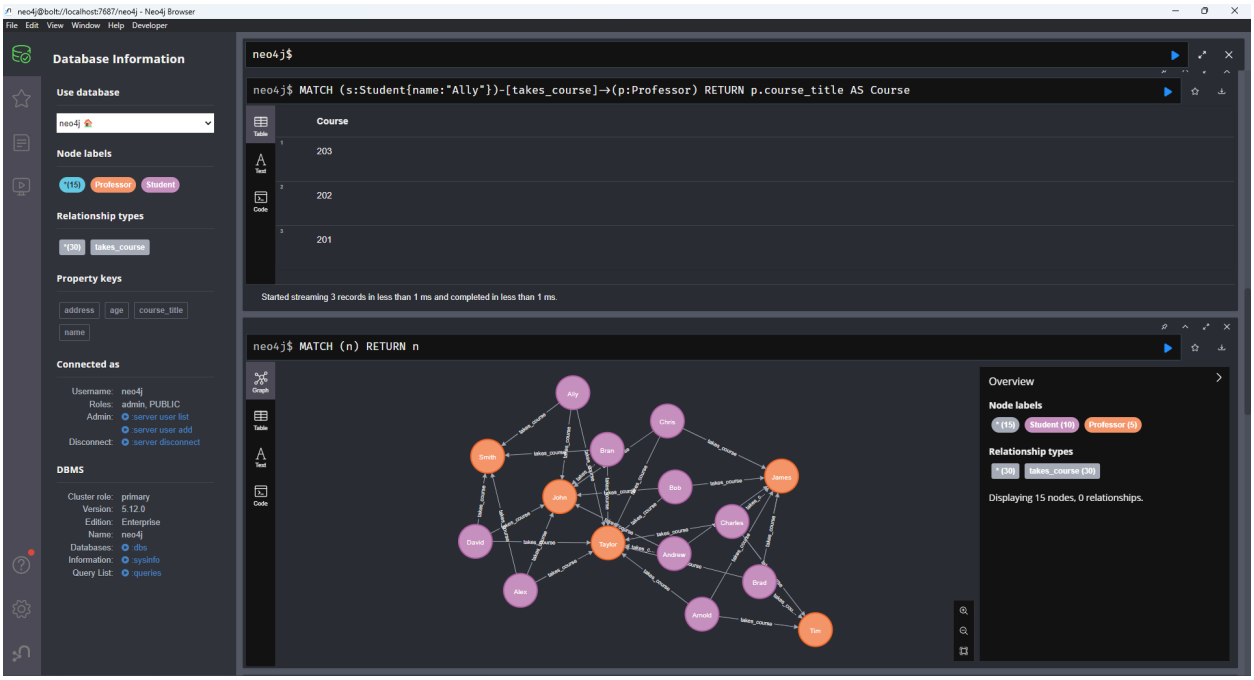
(s9)-[:takes_course]->(p3),(s9)-[:takes_course]->(p4),(s9)-[:takes_course]->(p5),

(s10)-[:takes_course]->(p1),(s10)-[:takes_course]->(p2),(s10)-[:takes_course]->(p3)

Question 7:

Display courses Ally took.

Screenshot 7:



Query 7:

MATCH

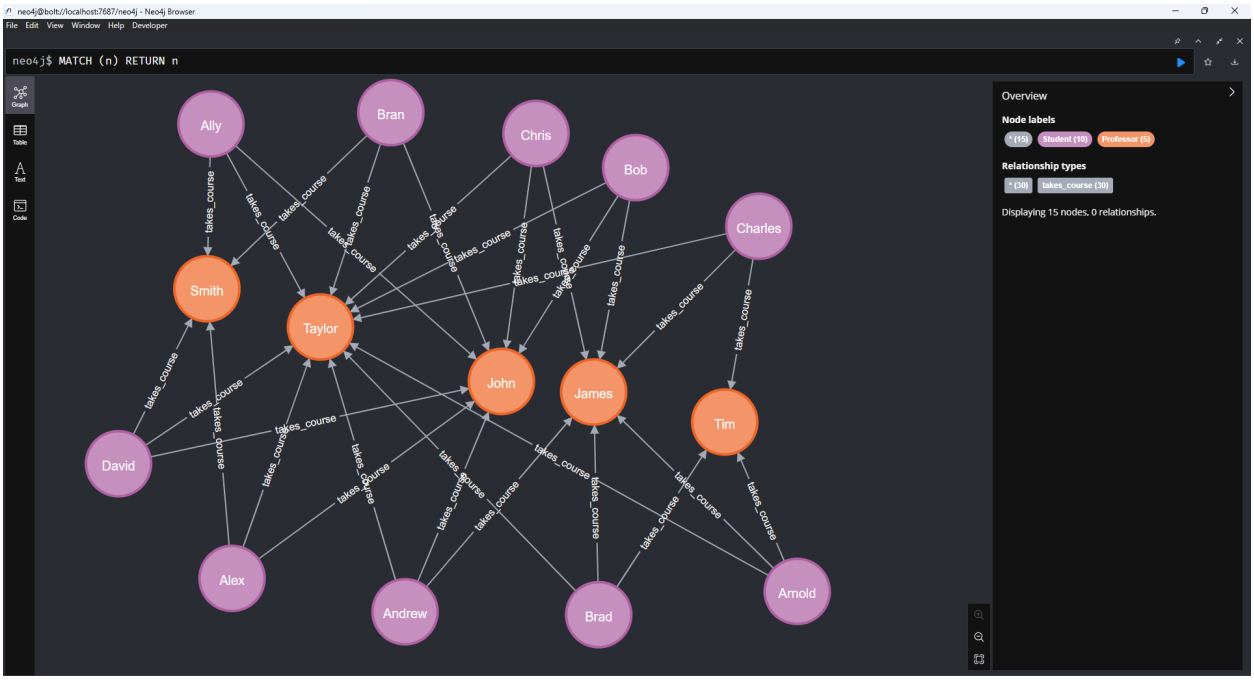
(s:Student{name:"Ally"})-[takes_course]->(p:Professor)

RETURN p.course_title AS Course

Question 8:

Display all 15 nodes with relationships

Screenshot 8:



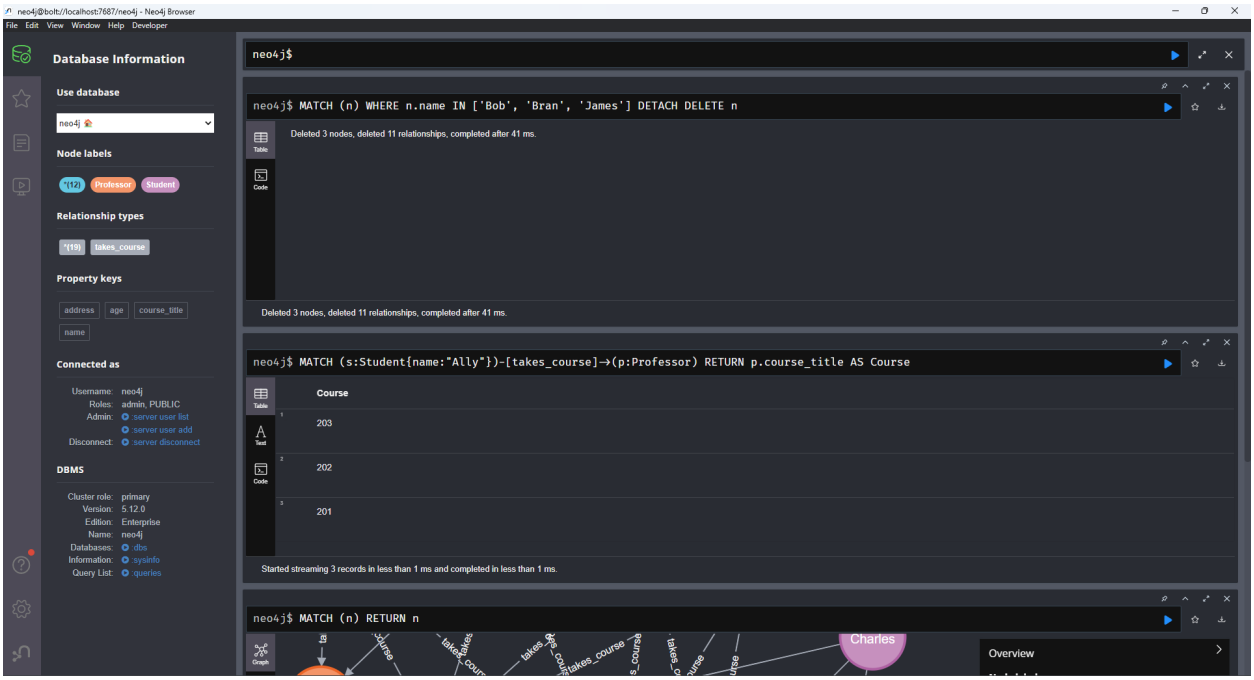
Query 8:

MATCH (n) RETURN n

Question 9:

Delete nodes Bob, Bran, James

Screenshot 9:



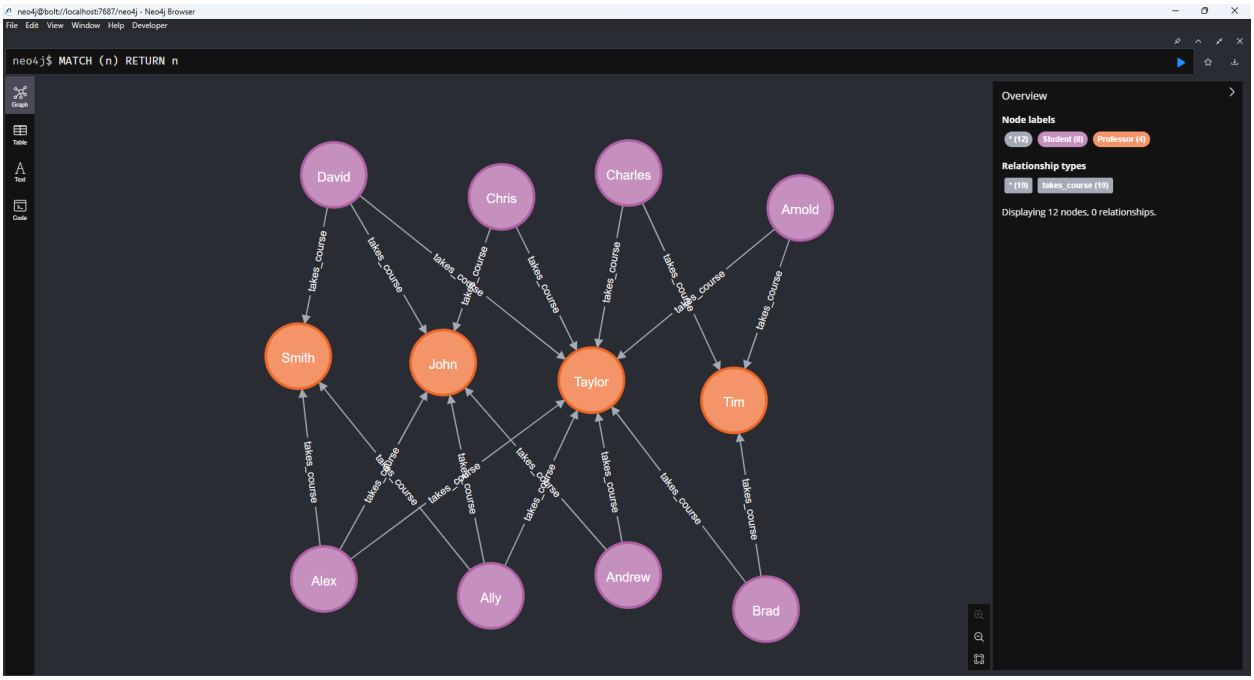
Query 9:

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