

## Data 225 data base systems Homework 8

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

Query:



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

The image shows the Neo4j Cypher query interface. The query entered is `neo4j$ match(n:student) return avg(n.age),max(n.age),min(n.age)`. The result is displayed in a table with three columns: `avg(n.age)`, `max(n.age)`, and `min(n.age)`. The table has one data row showing the values 14.3, 18, and 10 respectively.

	<code>avg(n.age)</code>	<code>max(n.age)</code>	<code>min(n.age)</code>
1	14.3	18	10

## Data 225 data base systems Homework 8

3) Display only Brad node

The screenshot shows the Neo4j web interface. The command bar at the top contains the Cypher query: `neo4j$ match (n:student {name:"Brad"}) return n`. The left sidebar has icons for Graph, Table, Text, and Code. The main canvas displays a single purple circular node labeled "Brad". On the right, the "Overview" panel shows "Node labels" with `^(1)` and `student (1)`, and a status message: "Displaying 1 nodes, 0 relationships."

4) Update age of David to be 26

The screenshot shows the Neo4j web interface after two operations. The top command bar contains the query: `neo4j$ match (n:student {name:"David"}) set n.age=26`. Below the command bar, a status message reads: "Set 1 property, completed after 6 ms." The bottom command bar contains the query: `neo4j$ match(n) return n limit 25`. The main canvas displays a graph with several purple circular nodes and a central hub-and-spoke structure. On the right, the "Node properties" panel for a `student` node shows the following data:

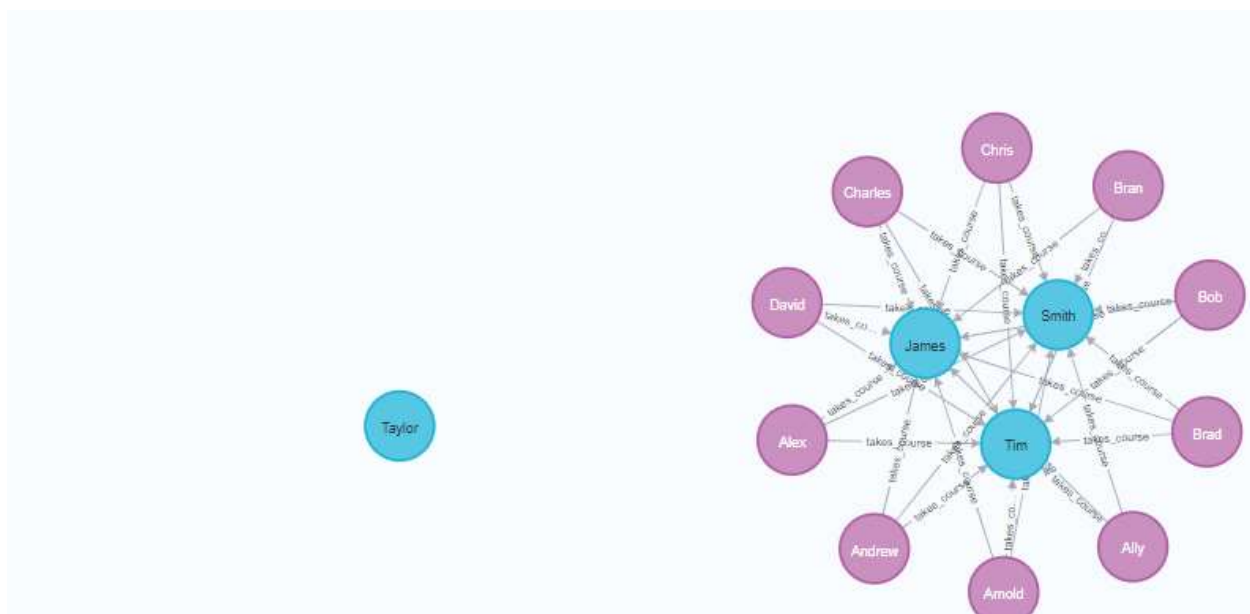
Property	Value
<code>&lt;id&gt;</code>	0
<code>address</code>	nizambad
<code>age</code>	26
<code>name</code>	David

## Data 225 data base systems Homework 8

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



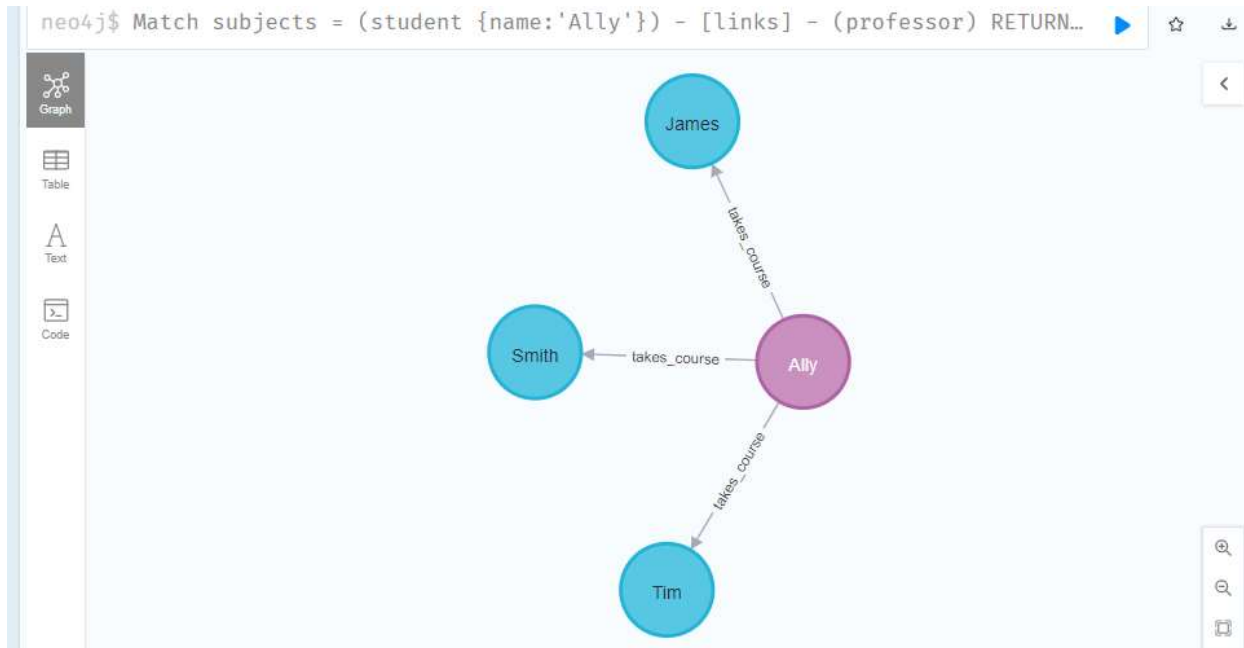
6) Create 'takes\_course' relationships (eg; **from:** Semester) between all students and any professor. (Make sure each student takes at least 3 courses) (Eg: Arnold takes\_course John) (Arnold takes\_course Tim)



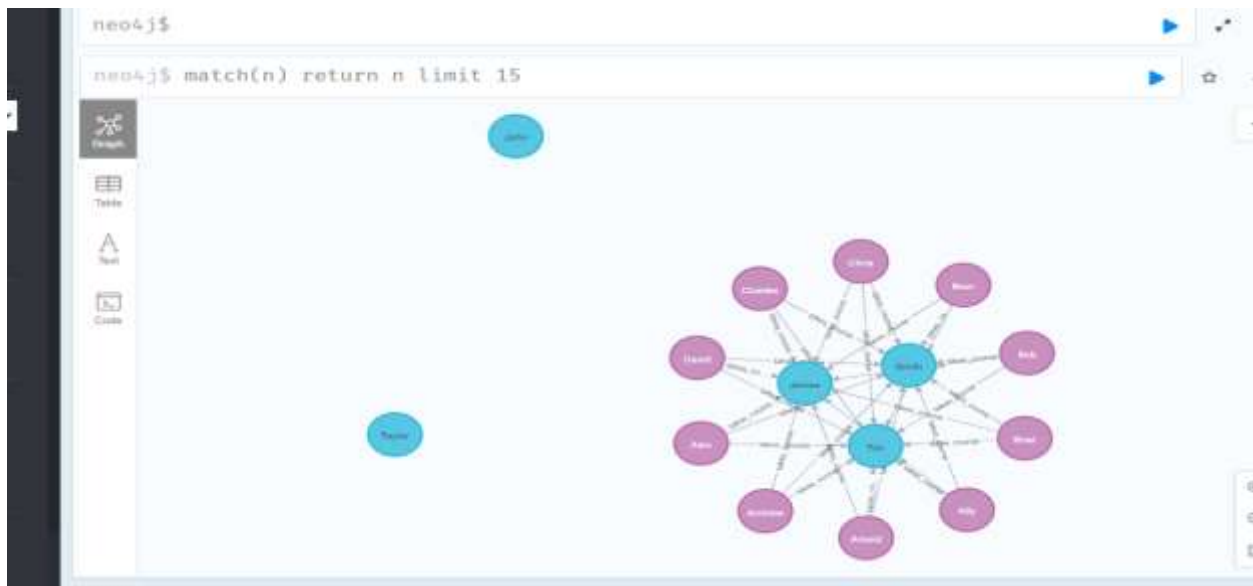
## Data 225 data base systems Homework 8

7) Display courses Ally took

Match subjects = (student {name:'Ally'}) - [links] - (professor) RETURN \*




8) Display all 15 nodes with relationships




## Data 225 data base systems Homework 8

9) Delete nodes Bob, Bran, James

```
neo4j$ MATCH (n {name: 'James'}) DETACH DELETE n
```


  
Table


  
Code

Deleted 1 node, deleted 8 relationships, completed after 7 ms.

Deleted 1 node, deleted 8 relationships, completed after 7 ms.

```
neo4j$ MATCH (n {name: 'Bran'}) DETACH DELETE n
```

  
Table

  
Code

Deleted 1 node, deleted 3 relationships, completed after 7 ms.

## Data 225 data base systems Homework 8

```
neo4j$ MATCH (n {name: 'Bob'}) DETACH DELETE n
```

Deleted 1 node, deleted 3 relationships, completed after 18 ms.

Table

Code

- 10) Display the graph

```
neo4j$ match(n) return n limit 15
```

Graph

Table

Text

Code

John

Taylor

Charles

Chris

David

Brad

Tim

Smith

Alex

Andrew

Arnold

Ally