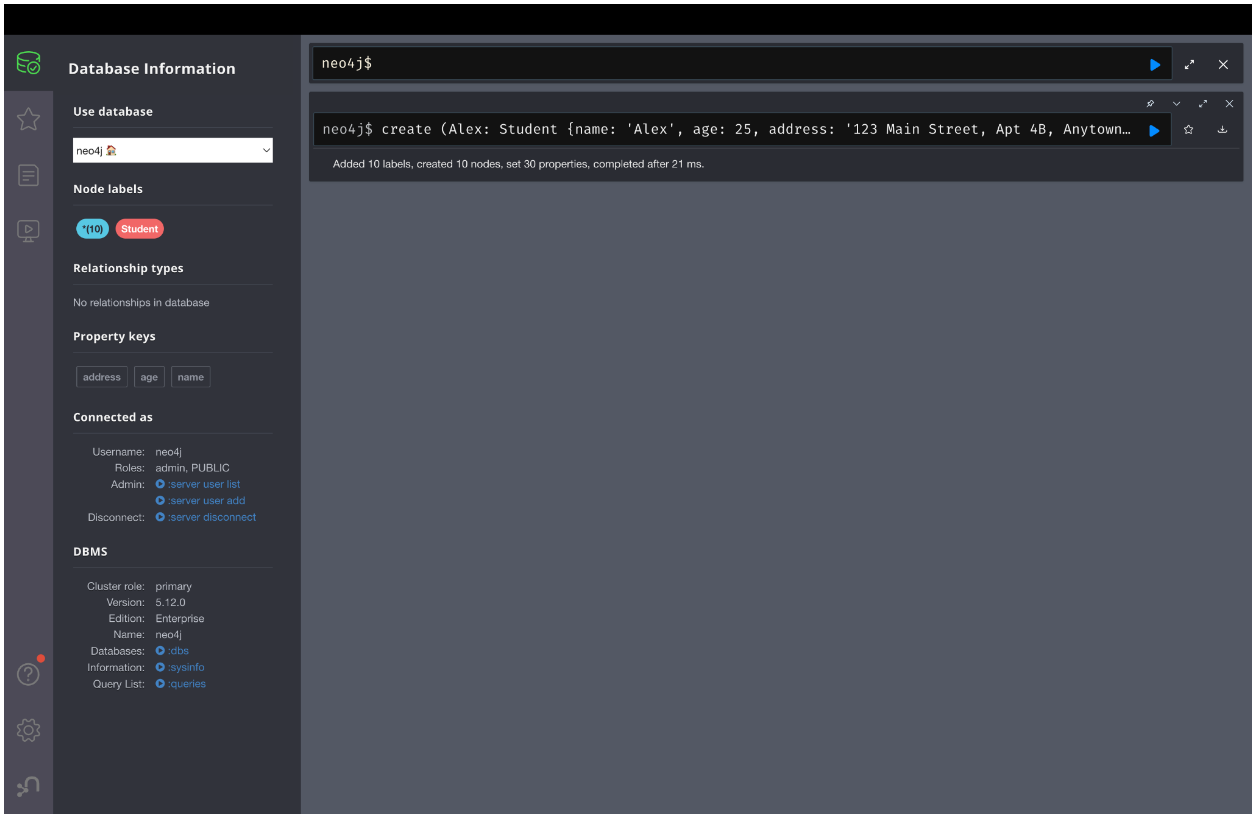
Homework 8

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

```
CREATE (Alex:Student {name: 'Alex', age: 25, address: '123 Main
Street, Apt 4B, Anytown'}),
       (Andrew:Student {name: 'Andrew', age: 22, address: '456 Elm
Street, Apt 3A, Othertown'}),
       (Arnold:Student {name: 'Arnold', age: 21, address: '789 Oak
Street, Apt 2C, Another town'}),
       (Ally:Student {name: 'Ally', age: 23, address: '321 Pine
Street, Apt 1D, Differenttown'}),
       (Bob:Student {name: 'Bob', age: 24, address: '654 Birch
Street, Apt 5E, Newtown'}),
       (Brad:Student {name: 'Brad', age: 26, address: '987 Cedar
Street, Apt 6F, Nearbytown'}),
       (Bran:Student {name: 'Bran', age: 20, address: '234 Maple
Street, Apt 7G, Nearbytown'}),
       (Chris:Student {name: 'Chris', age: 27, address: '567 Walnut
Street, Apt 8H, Nearbytown'}),
       (Charles:Student {name: 'Charles', age: 22, address: '876
Redwood Street, Apt 9I, Nearbytown'}),
       (David:Student {name: 'David', age: 28, address: '123 Spruce
Street, Apt 10J, Nearbytown'});
```



Database Information

Use database

neo4j

Node labels

*(10) 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
Disconnect: :server disconnect
```

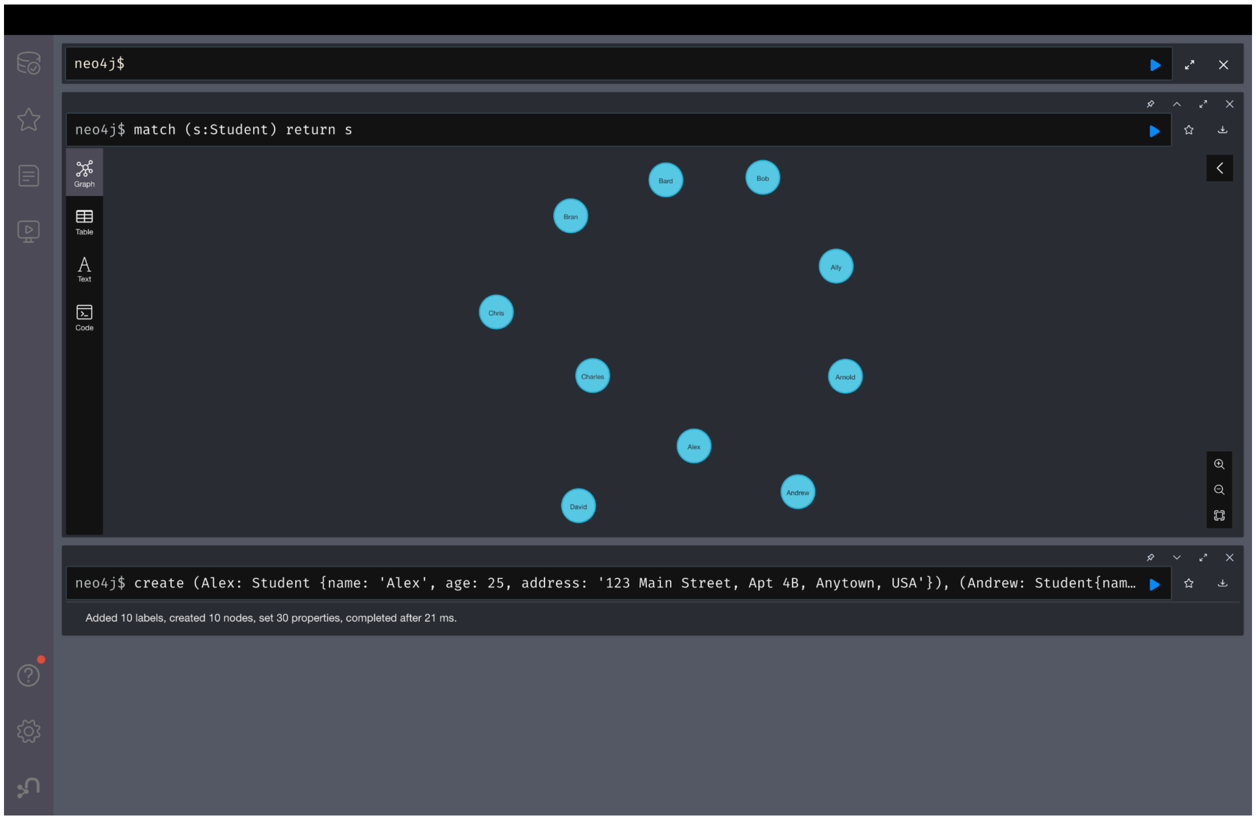
DBMS

```
Cluster role: primary
     Version: 5.12.0
     Edition: Enterprise
        Name: neo4j
   Databases: :dbs
 Information: :sysinfo
  Query List: :queries
```

```
neo4j$
```

```
neo4j$ create (Alex: Student {name: 'Alex', age: 25, address: '123 Main Street, Apt 4B, Anytown…
```

Added 10 labels, created 10 nodes, set 30 properties, completed after 21 ms.



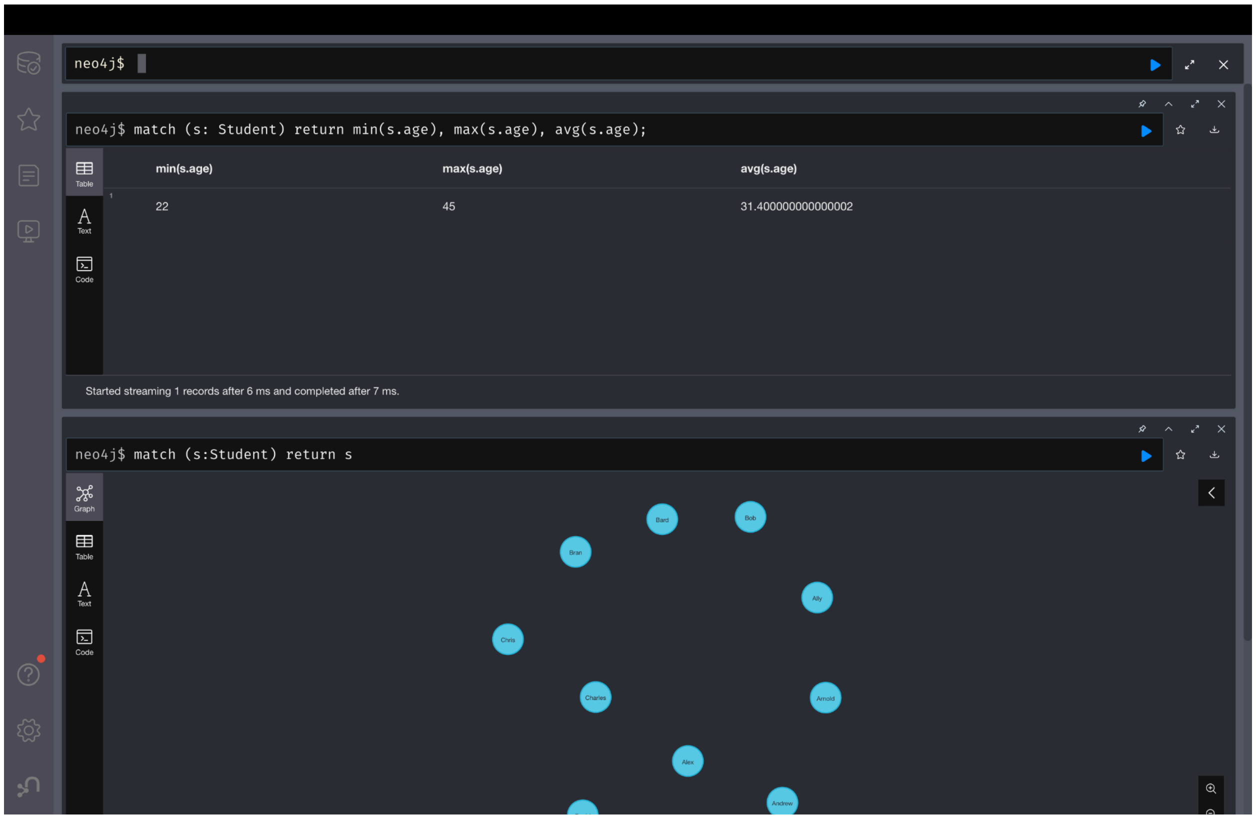
```
neo4j$
```

```
neo4j$ match (s:Student) return s
```

```
neo4j$ create (Alex: Student {name: 'Alex', age: 25, address: '123 Main Street, Apt 4B, Anytown, USA'}), (Andrew: Student{nam…
```

Added 10 labels, created 10 nodes, set 30 properties, completed after 21 ms.

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



```
neo4j$
```

```
neo4j$ match (s: Student) return min(s.age), max(s.age), avg(s.age);
```

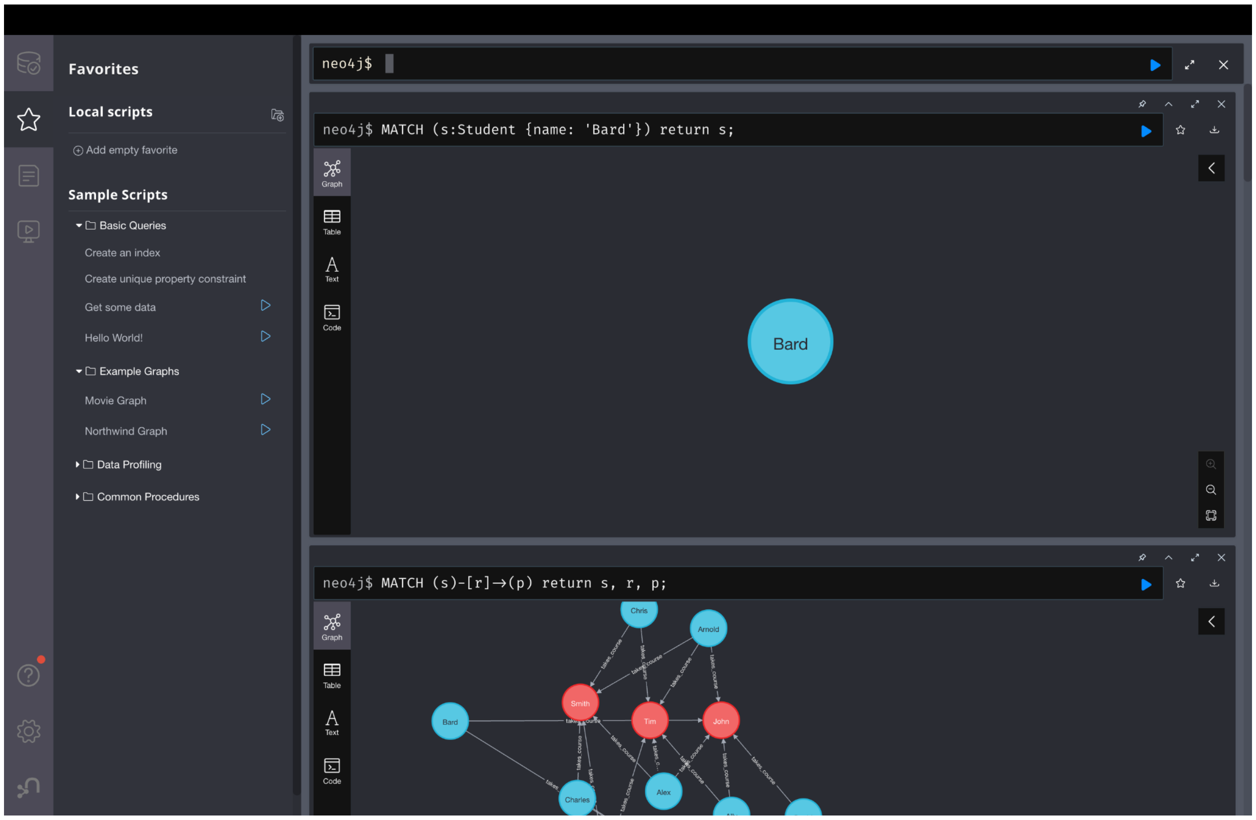
min(s.age)	max(s.age)	avg(s.age)
22	45	31.400000000000002

Started streaming 1 records after 6 ms and completed after 7 ms.

```
neo4j$ match (s:Student) return s
```

3. Display only Brad node.

```
MATCH (s:Student {name: 'Bard'}) return s;
```



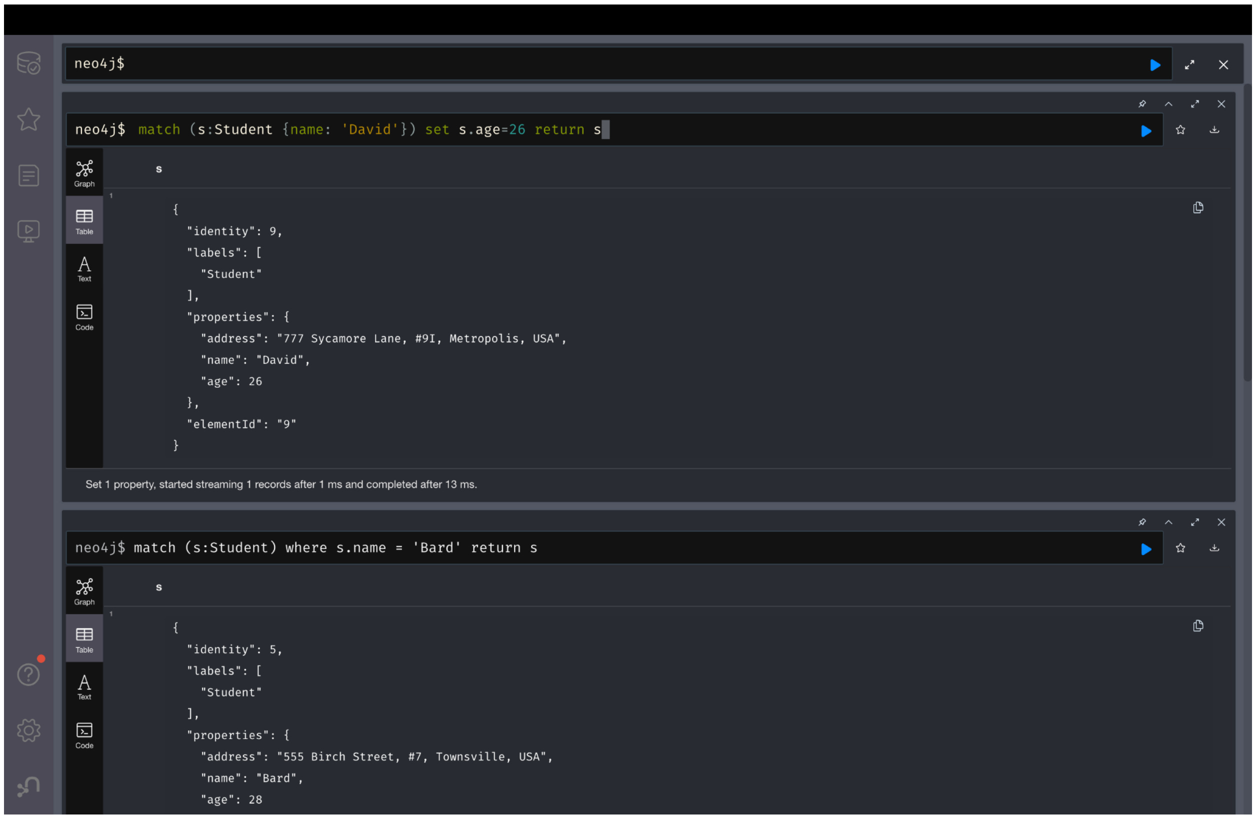
```
neo4j$
```

```
neo4j$ MATCH (s:Student {name: 'Bard'}) return s;
```

```
neo4j$ MATCH (s)-[r]→(p) return s, r, p;
```

4. Update age of David to be 26

```
match (s:Student {name: 'David'}) set s.age=26 return s;
```



```
neo4j$
```

```
neo4j$ match (s:Student {name: 'David'}) set s.age=26 return s
```

```
s
1
  {
    "identity": 9,
    "labels": [
      "Student"
    ],
    "properties": {
      "address": "777 Sycamore Lane, #9I, Metropolis, USA",
      "name": "David",
      "age": 26
    },
    "elementId": "9"
  }
```

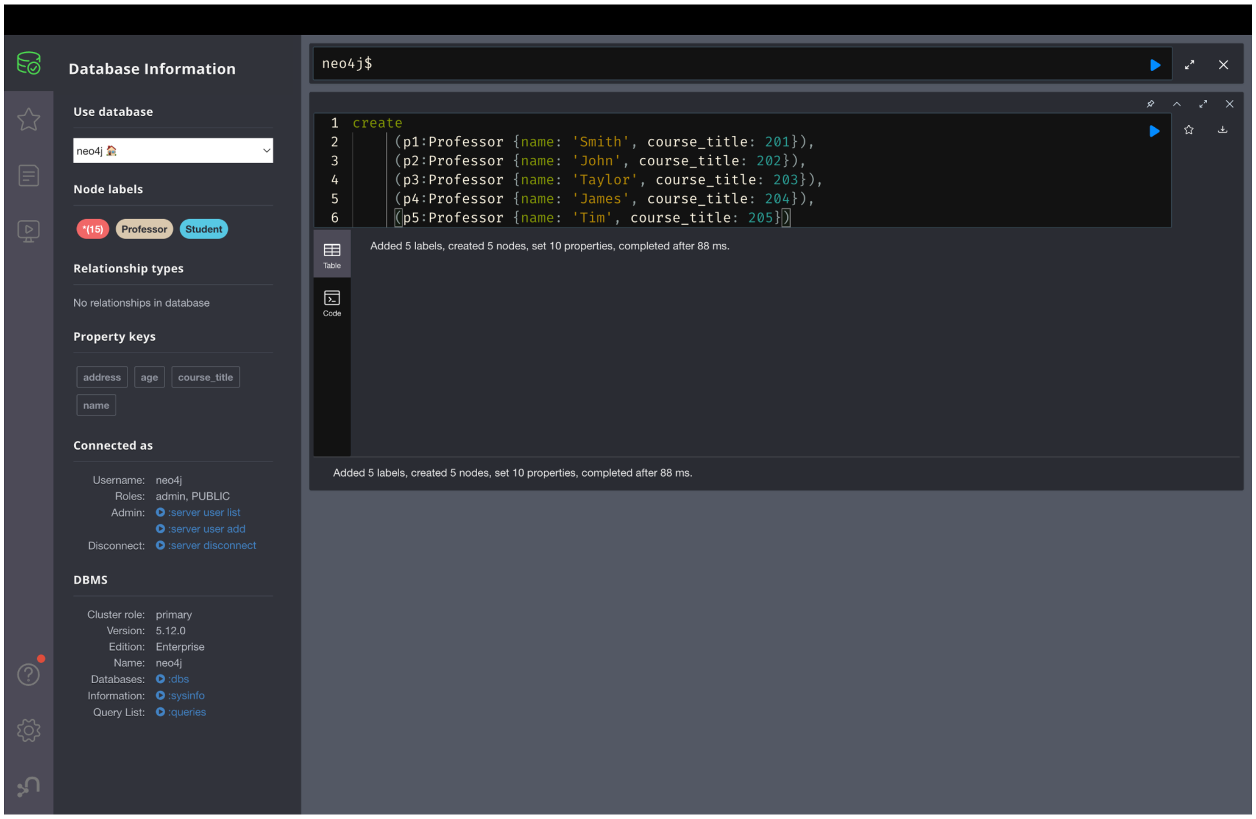
Set 1 property, started streaming 1 records after 1 ms and completed after 13 ms.

```
neo4j$ match (s:Student) where s.name = 'Bard' return s
```

```
s
1
  {
    "identity": 5,
    "labels": [
      "Student"
    ],
    "properties": {
      "address": "555 Birch Street, #7, Townsville, USA",
      "name": "Bard",
      "age": 28
```

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

```
create
(p1:Professor {name: 'Smith', course_title: 201}),
(p2:Professor {name: 'John', course_title: 202}),
(p3:Professor {name: 'Taylor', course_title: 203}),
(p4:Professor {name: 'James', course_title: 204}),
(p5:Professor {name: 'Tim', course_title: 205})
```



Database Information

Use database

neo4j

Node labels

*(15) Professor Student

Relationship types

No relationships in database

Property keys

address age course_title name

Connected as

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

DBMS

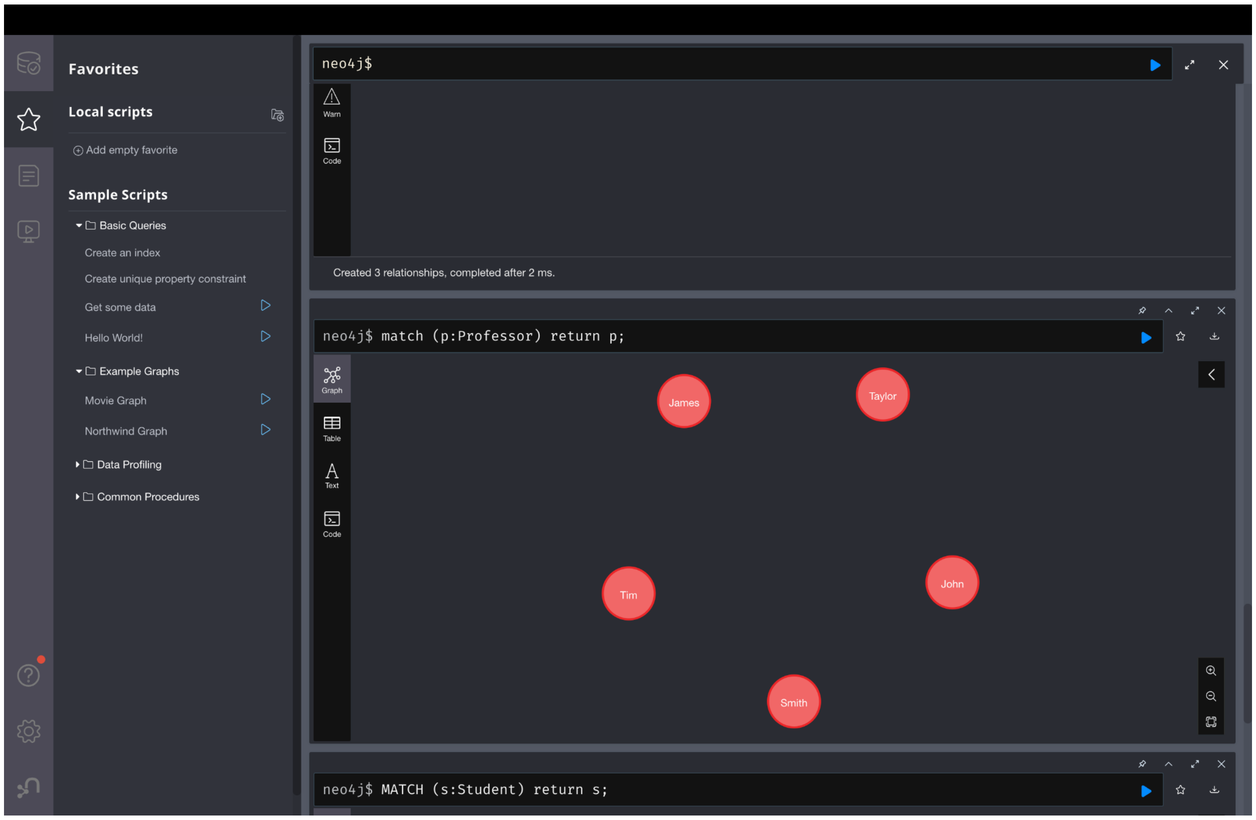
Cluster role: primary
Version: 5.12.0
Edition: Enterprise
Name: neo4j
Databases: :dbs
Information: :sysinfo
Query List: :queries

```
neo4j$
```

```
create
    (p1:Professor {name: 'Smith', course_title: 201}),
    (p2:Professor {name: 'John', course_title: 202}),
    (p3:Professor {name: 'Taylor', course_title: 203}),
    (p4:Professor {name: 'James', course_title: 204}),
    (p5:Professor {name: 'Tim', course_title: 205})
```

```
Added 5 labels, created 5 nodes, set 10 properties, completed after 88 ms.
```

```
Added 5 labels, created 5 nodes, set 10 properties, completed after 88 ms.
```



Favorites

Local scripts

Add empty favorite

Sample Scripts

Basic Queries
Create an index
Create unique property constraint
Get some data
Hello World!
Example Graphs
Movie Graph
Northwind Graph
Data Profiling
Common Procedures

```
neo4j$
```

```
Created 3 relationships, completed after 2 ms.
```

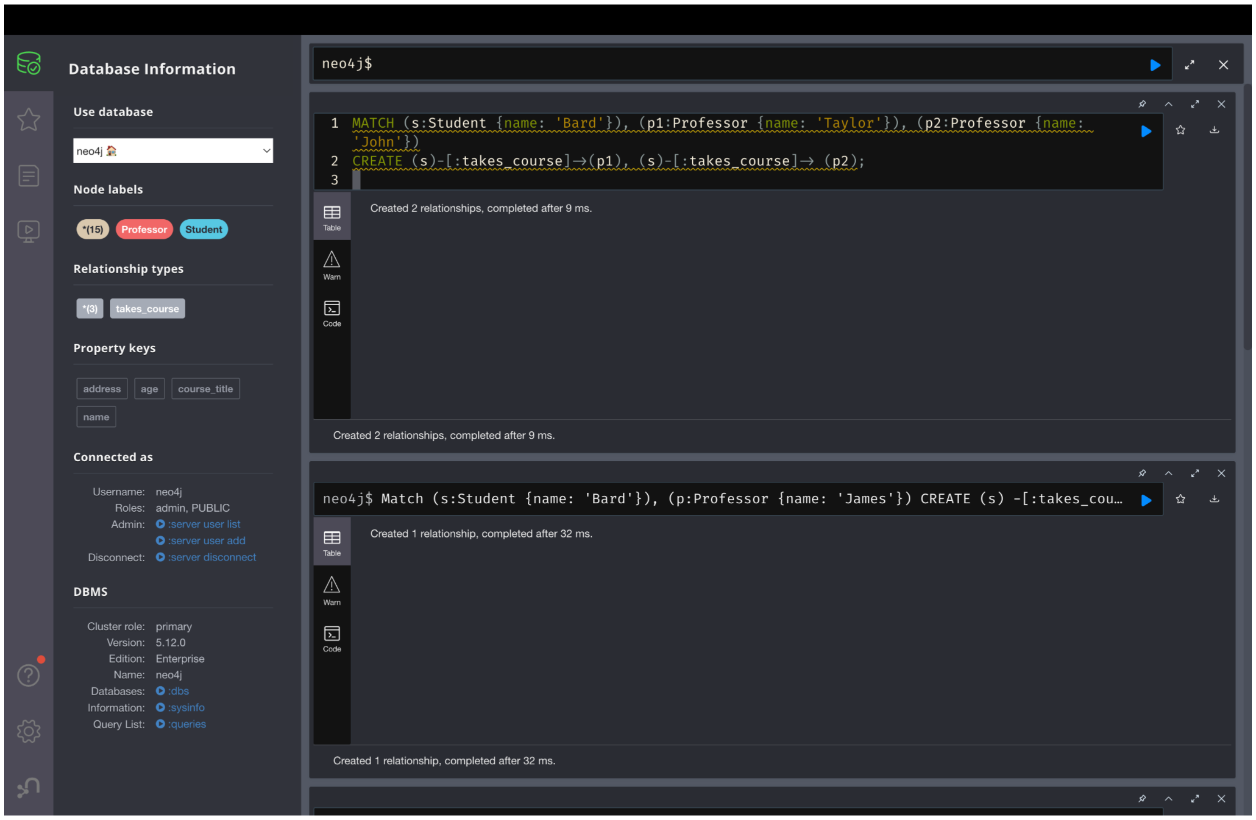
```
neo4j$ match (p:Professor) return p;
```

```
neo4j$ MATCH (s:Student) return s;
```

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)

```
MATCH
(s: Student {name: 'Bran'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'Smith'})
CREATE (s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_course]->(p3)

MATCH
(s: Student {name: 'Chris'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'James'}),
(p3:Professor {name: 'Smith'})
CREATE (s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_course]->(p3)

MATCH
(s: Student {name: 'Charles'}),
(p1:Professor {name: 'Taylor'}),
(p2:Professor {name: 'James'}),
(p3:Professor {name: 'Smith'})
CREATE (s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_course]->(p3)

MATCH
(s: Student {name: 'David'}),
(p1:Professor {name: 'James'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'Taylor'})
CREATE (s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_course]->(p3)
```

```
MATCH
(s: Student {name: 'Alex'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'Smith'})
CREATE
(s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_cours
e]->(p3)

MATCH
(s: Student {name: 'Andrew'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'Taylor'}),
(p3:Professor {name: 'Smith'})
CREATE
(s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_cours
e]->(p3)

MATCH
(s: Student {name: 'Arnold'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'Smith'})
CREATE
(s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_cours
e]->(p3)

MATCH
(s: Student {name: 'Ally'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'James'})
CREATE
(s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_cours
e]->(p3)

MATCH
(s: Student {name: 'Bob'}),
(p1:Professor {name: 'James'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'Smith'})
CREATE
(s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_cours
e]->(p3)

MATCH
(s: Student {name:'Bran'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'James'})
CREATE
(s)-[:takes_course]->(p1),(s)-[:takes_course]->(p2),(s)-[:takes_cours
e]->(p3)
```



Database Information

Use database: neo4j

Node labels: *(15) Professor Student

Relationship types: *(3) takes_course

Property keys: address age course_title name

Connected as

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

DBMS

Cluster role:	primary
Version:	5.12.0
Edition:	Enterprise
Name:	neo4j
Databases:	:dbs
Information:	:sysinfo
Query List:	:queries

```
neo4j$
```

```
MATCH (s:Student {name: 'Bard'}), (p1:Professor {name: 'Taylor'}), (p2:Professor {name:
'John'})
CREATE (s)-[:takes_course]→(p1), (s)-[:takes_course]→ (p2);
```

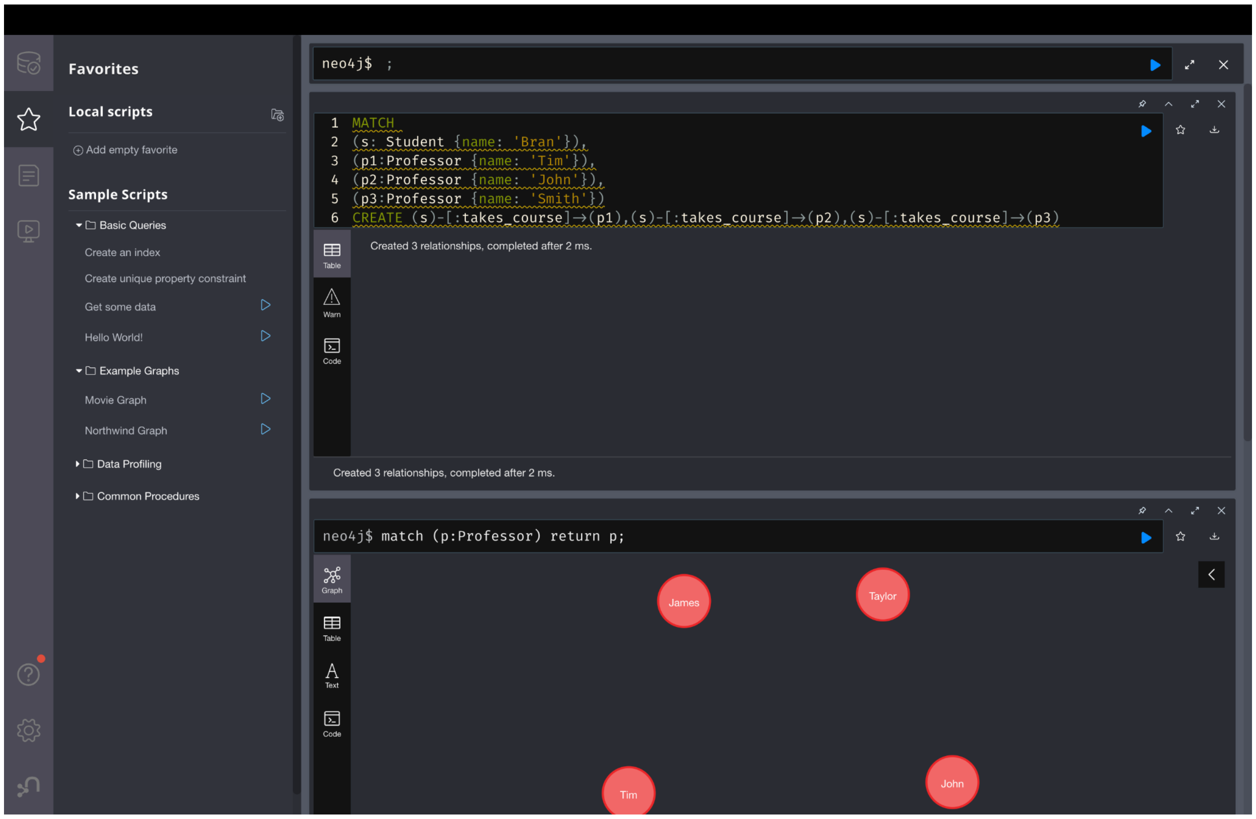
Created 2 relationships, completed after 9 ms.

Created 2 relationships, completed after 9 ms.

```
neo4j$ Match (s:Student {name: 'Bard'}), (p:Professor {name: 'James'}) CREATE (s) -[:takes_cou…
```

Created 1 relationship, completed after 32 ms.

Created 1 relationship, completed after 32 ms.



```
neo4j$ ;
```

```
MATCH
(s: Student {name: 'Bran'}),
(p1:Professor {name: 'Tim'}),
(p2:Professor {name: 'John'}),
(p3:Professor {name: 'Smith'})
CREATE (s)-[:takes_course]→(p1),(s)-[:takes_course]→(p2),(s)-[:takes_course]→(p3)
```

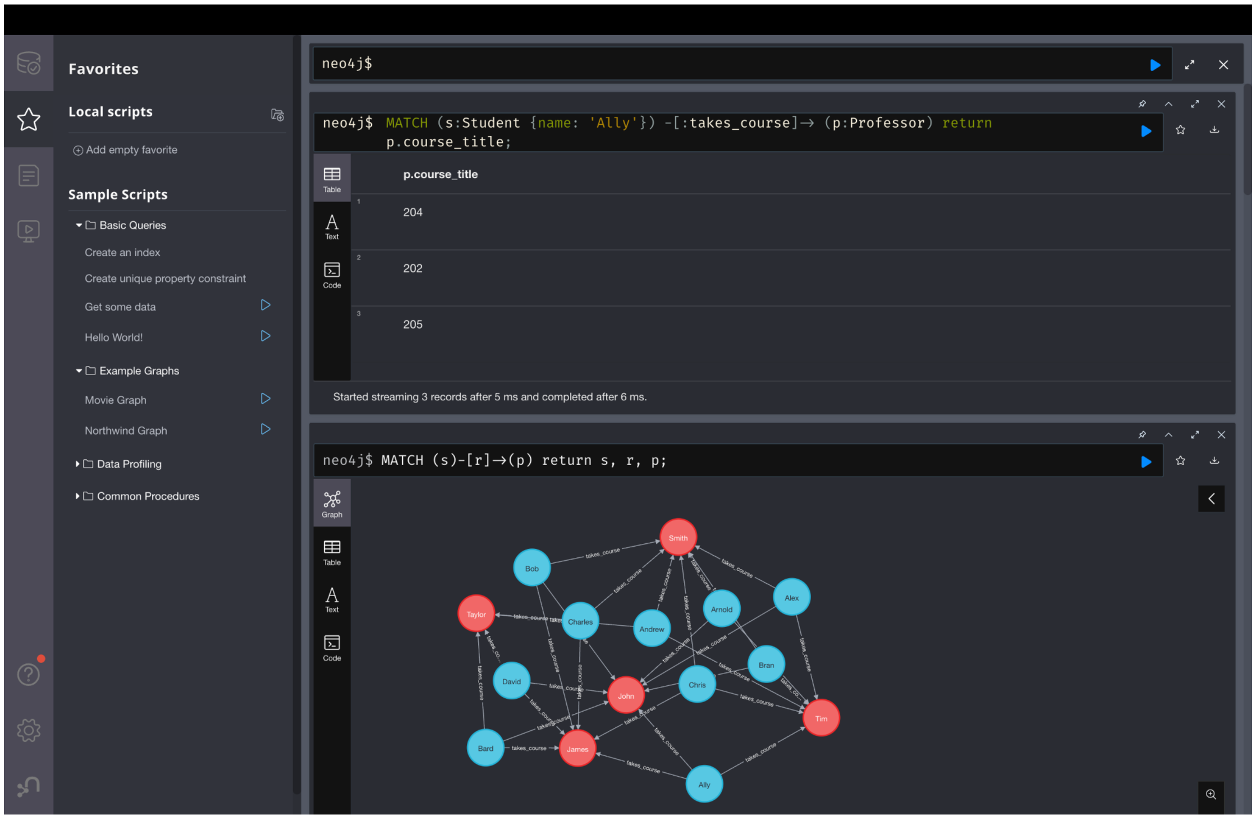
```
Created 3 relationships, completed after 2 ms.
```

```
Created 3 relationships, completed after 2 ms.
```

```
neo4j$ match (p:Professor) return p;
```

7. Display courses Ally took.

```
MATCH (s:Student {name: 'Ally'}) -[:takes_course]-> (p:Professor) return
p.course_title;
```



```
neo4j$
```

```
neo4j$ MATCH (s:Student {name: 'Ally'}) -[:takes_course]→ (p:Professor) return
       p.course_title;
```

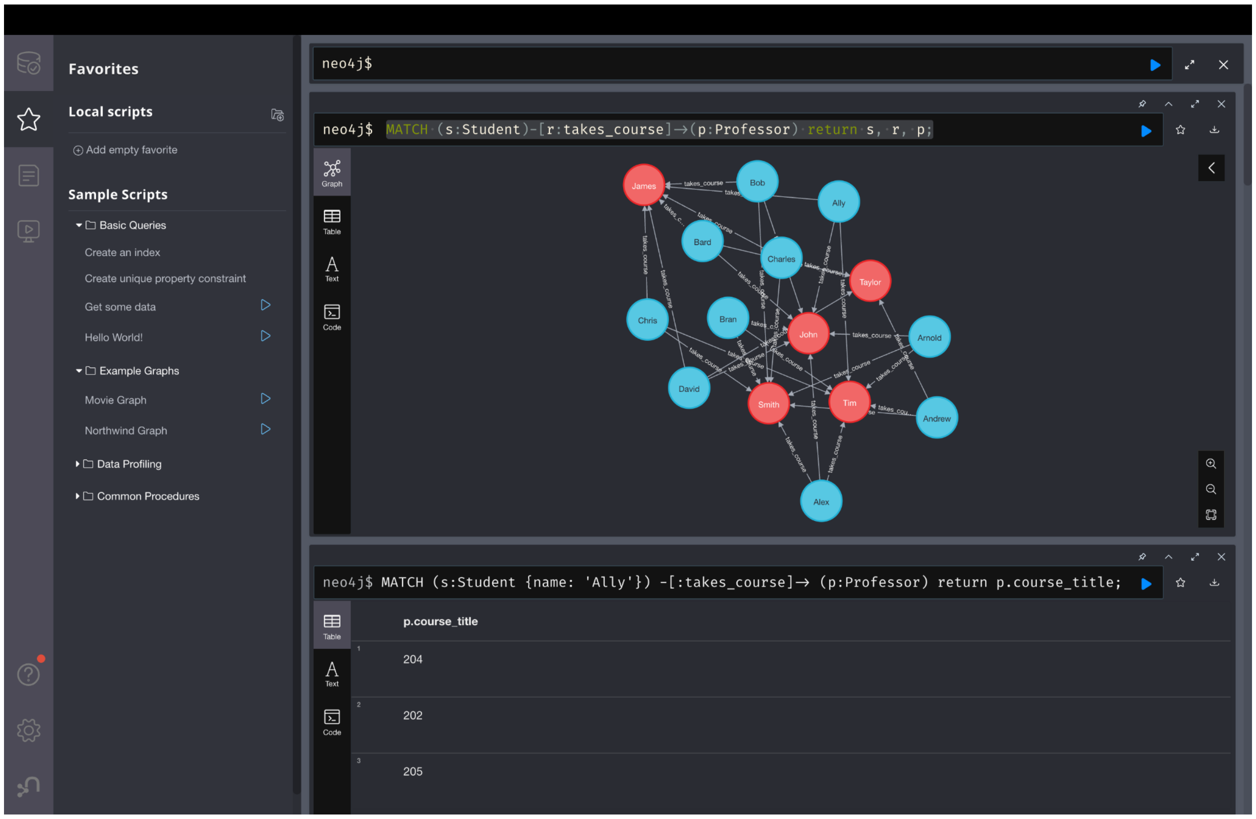
	p.course_title
1	204
2	202
3	205

Started streaming 3 records after 5 ms and completed after 6 ms.

```
neo4j$ MATCH (s)-[r]→(p) return s, r, p;
```

8. Display all 15 nodes with relationships

```
MATCH (s:Student)-[r:takes_course]->(p:Professor) return s, r, p;
```



```
neo4j$
```

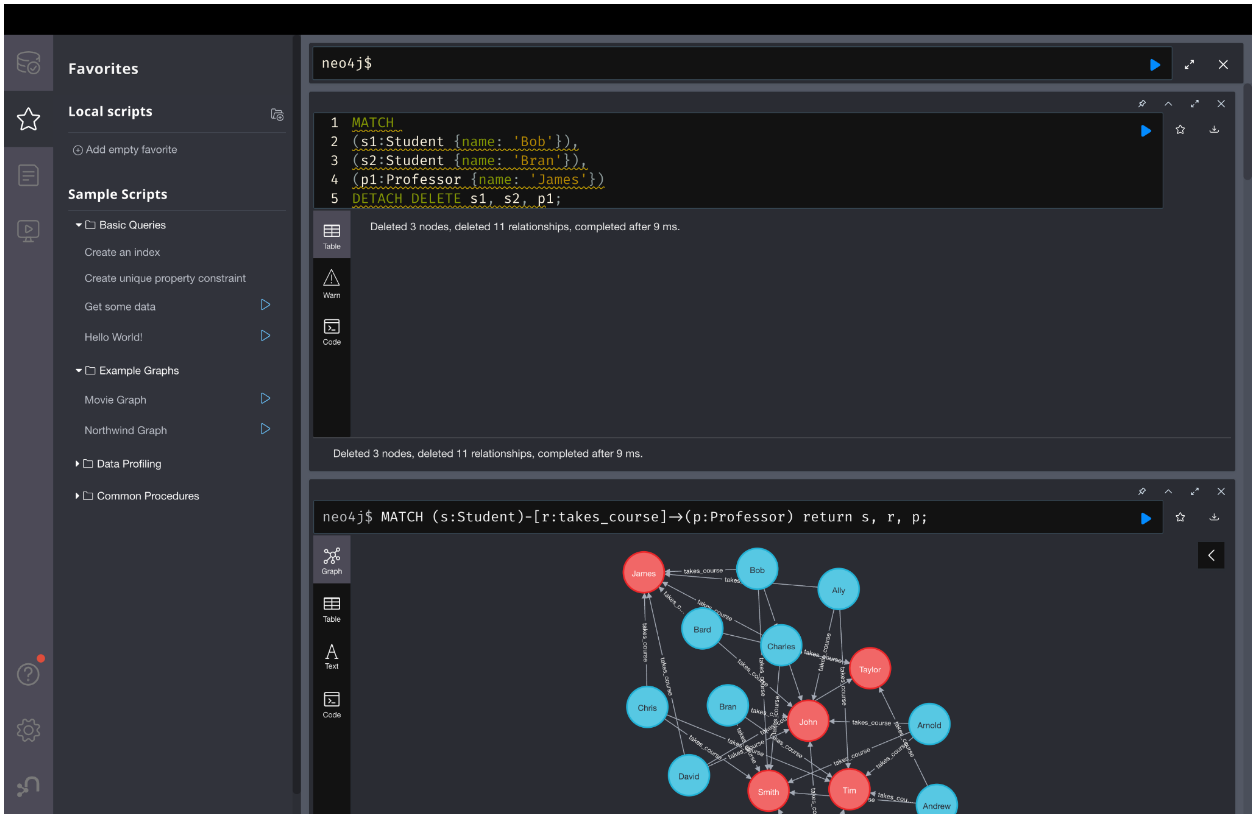
```
neo4j$ MATCH (s:Student)-[r:takes_course]→(p:Professor) return s, r, p;
```

```
neo4j$ MATCH (s:Student {name: 'Ally'}) -[:takes_course]→ (p:Professor) return p.course_title;
```

	p.course_title
1	204
2	202
3	205

9. Delete nodes Bob, Bran, James

```
MATCH
(s1:Student {name: 'Bob'}),
(s2:Student {name: 'Bran'}),
(p1:Professor {name: 'James'})
DETACH DELETE s1, s2, p1;
```



```
neo4j$
```

```
MATCH
(s1:Student {name: 'Bob'}),
(s2:Student {name: 'Bran'}),
(p1:Professor {name: 'James'})
DETACH DELETE s1, s2, p1;
```

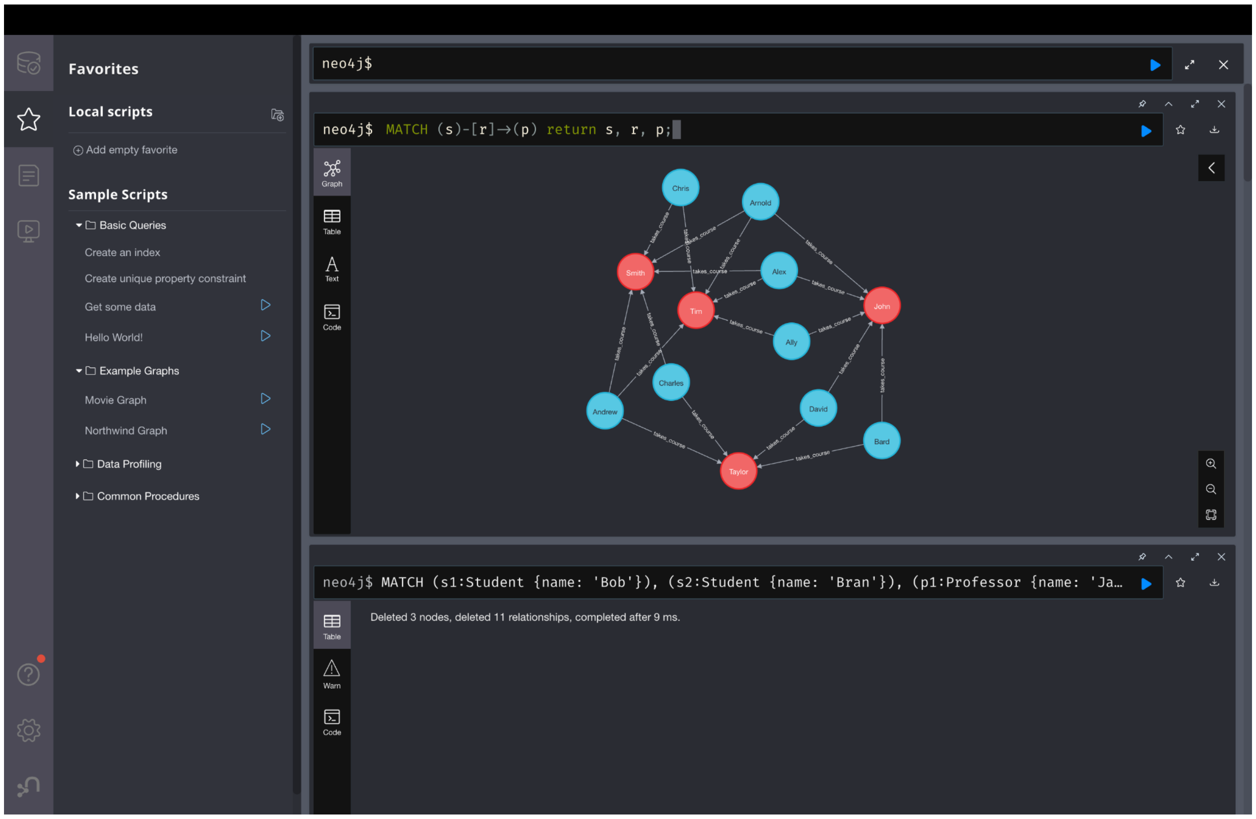
Deleted 3 nodes, deleted 11 relationships, completed after 9 ms.

Deleted 3 nodes, deleted 11 relationships, completed after 9 ms.

```
neo4j$ MATCH (s:Student)-[r:takes_course]→(p:Professor) return s, r, p;
```

10. Display the graph

```
MATCH (s)-[r]->(p) return s, r, p;
```



```
neo4j$
```

```
neo4j$  MATCH (s)-[r]→(p) return s, r, p;
```

```
neo4j$ MATCH (s1:Student {name: 'Bob'}), (s2:Student {name: 'Bran'}), (p1:Professor {name: 'Ja…
```

```
Deleted 3 nodes, deleted 11 relationships, completed after 9 ms.
```