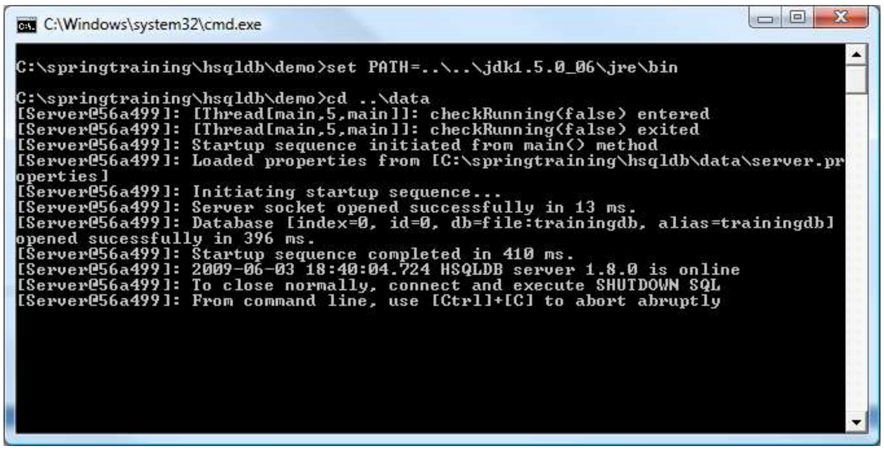
Lab 4

Part 1

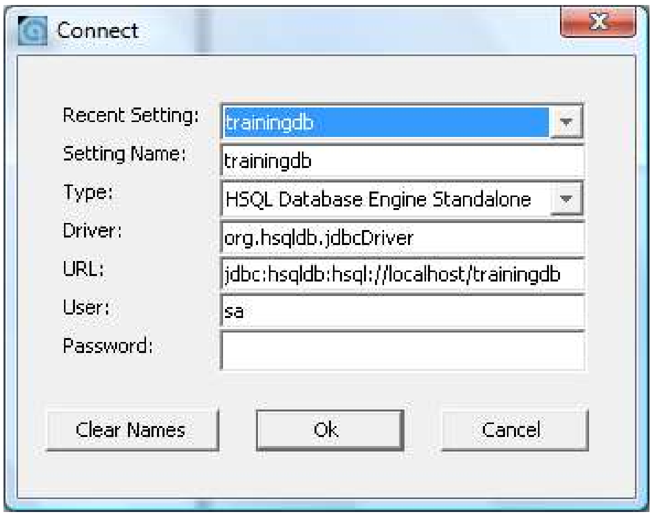
For this exercise, we will be using the HyperSQL database (HSQLDB)..

You can start the training database by double-clicking on **C:\SoftwareArchiteture \hsqldb\bin\runServerTrainingdb.bat.** This should open the following window to indicate that the database is running.



```
C:\springtraining\hsqldb\demo>set PATH=..\..\jdk1.5.0_06\jre\bin

C:\springtraining\hsqldb\demo>cd ..\data
[Server@56a499]: [Thread[main,5,main]]: checkRunning(false) entered
[Server@56a499]: [Thread[main,5,main]]: checkRunning(false) exited
[Server@56a499]: Startup sequence initiated from main() method
[Server@56a499]: Loaded properties from [C:\springtraining\hsqldb\data\server.pr
operties]
[Server@56a499]: Initiating startup sequence...
[Server@56a499]: Server socket opened successfully in 13 ms.
[Server@56a499]: Database [index=0, id=0, db=file:trainingdb, alias=trainingdb]
opened sucessfully in 396 ms.
[Server@56a499]: Startup sequence completed in 410 ms.
[Server@56a499]: 2009-06-03 18:40:04.724 HSQLDB server 1.8.0 is online
[Server@56a499]: To close normally, connect and execute SHUTDOWN SQL
[Server@56a499]: From command line, use [Ctrl]+[C] to abort abruptly
```

Then, start the database manager by double-clicking on **runManagerSwing.bat** in the same directory.



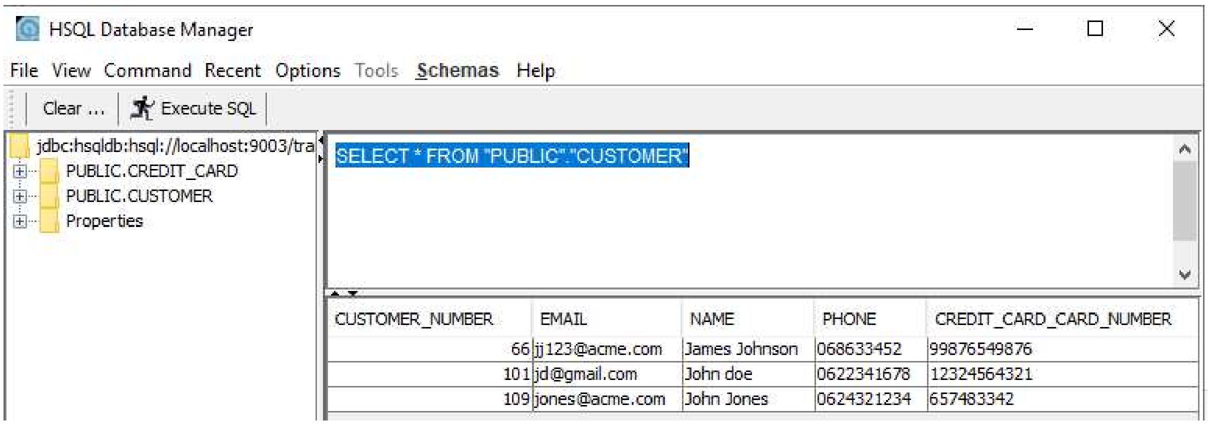
Recent Setting: trainingdb
Setting Name: trainingdb
Type: HSQL Database Engine Standalone
Driver: org.hsqldb.jdbcDriver
URL: jdbc:hsqldb:hsql://localhost/trainingdb
User: sa
Password:

When the manager asks for connection settings, and if this is the first time you start the database manager, then fill in the following information:

Setting Name: **trainingdb**
Type: **HSQL Database Engine Standalone**
Driver: **org.hsqldb.jdbcDriver**
URL: **jdbc:hsqldb:hsql://localhost:9003/trainingdb**
User: **SA**
Password:

And click the **OK** button.

Run the given **Lesson4SpringJPADemo** application and see that the application saves some customers with creditcards in the database.


HSQL Database Manager
File View Command Recent Options Tools Schemas Help
Clear ... Execute SQL
jdbc:hsqldb:hsql://localhost:9003/tra
PUBLIC.CREDIT_CARD
PUBLIC.CUSTOMER
Properties

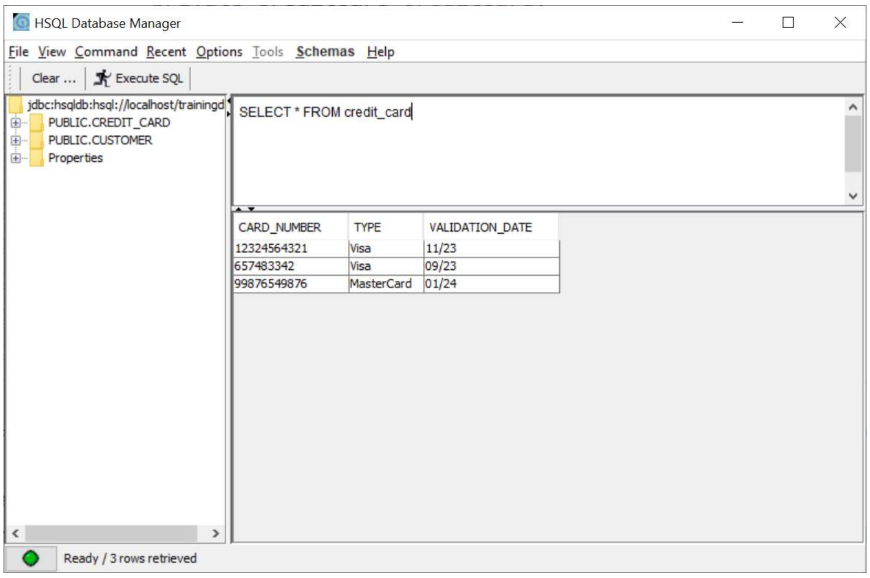
```
SELECT * FROM "PUBLIC"."CUSTOMER"
```

CUSTOMER_NUMBER	EMAIL	NAME	PHONE	CREDIT_CARD_CARD_NUMBER
66	jj123@acme.com	James Johnson	068633452	99876549876
101	jd@gmail.com	John doe	0622341678	12324564321
109	jones@acme.com	John Jones	0624321234	657483342

Modify the given **Lesson4SpringJPADemo** application so that the application stores students in the database
A Student contains the attributes name, phoneNumber and email.
A student has also an address.
Create a class Address with the attributes street, city and zip.
In the application add 5 students in the database.
Then perform the following queries:

- Get all students
- Get all students with a certain name
- Get a student with a certain phoneNumber
- Get all students from a certain city

Check the data in the database with the HSQLDatabase manager


HSQL Database Manager
File View Command Recent Options Tools Schemas Help
Clear ... Execute SQL
jdbc:hsqldb:hsql://localhost/trainingd
PUBLIC.CREDIT_CARD
PUBLIC.CUSTOMER
Properties

```
SELECT * FROM credit_card
```

CARD_NUMBER	TYPE	VALIDATION_DATE
12324564321	Visa	11/23
657483342	Visa	09/23
99876549876	MasterCard	01/24

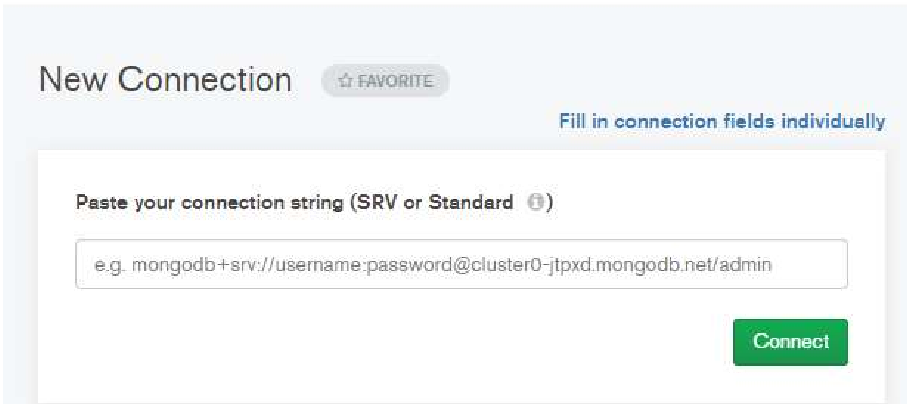
Ready / 3 rows retrieved

Part 2

First we have to start the mongo database by running

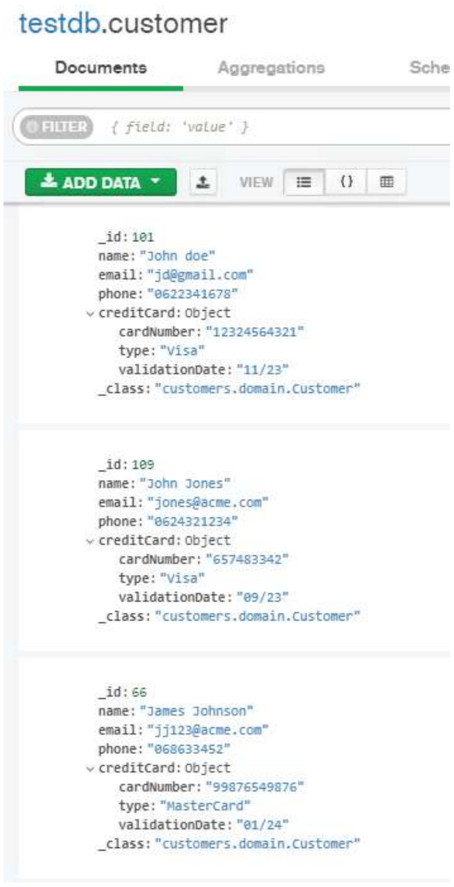
C:\SoftwareArchiteture\mongodb\bin\startmongo.bat

Then start **MongoCompass** by double clicking the file
C:\SoftwareArchiteture \mongocompass\MongoDBCompass.exe


New Connection
☆ FAVORITE
Fill in connection fields individually
Paste your connection string (SRV or Standard ⓘ)
e.g. mongodb+srv://username:password@cluster0-jtpxd.mongodb.net/admin
Connect

Click the **Connect** button.

Run the given **Lesson4SpringMongoDemo** application and see that the application saves some customers with creditcards in the database.



```
_id: 101
name: "John doe"
email: "jd@gmail.com"
phone: "0622341678"
creditCard: Object
   cardNumber: "12324564321"
   type: "Visa"
   validationDate: "11/23"
_class: "customers.domain.Customer"
```

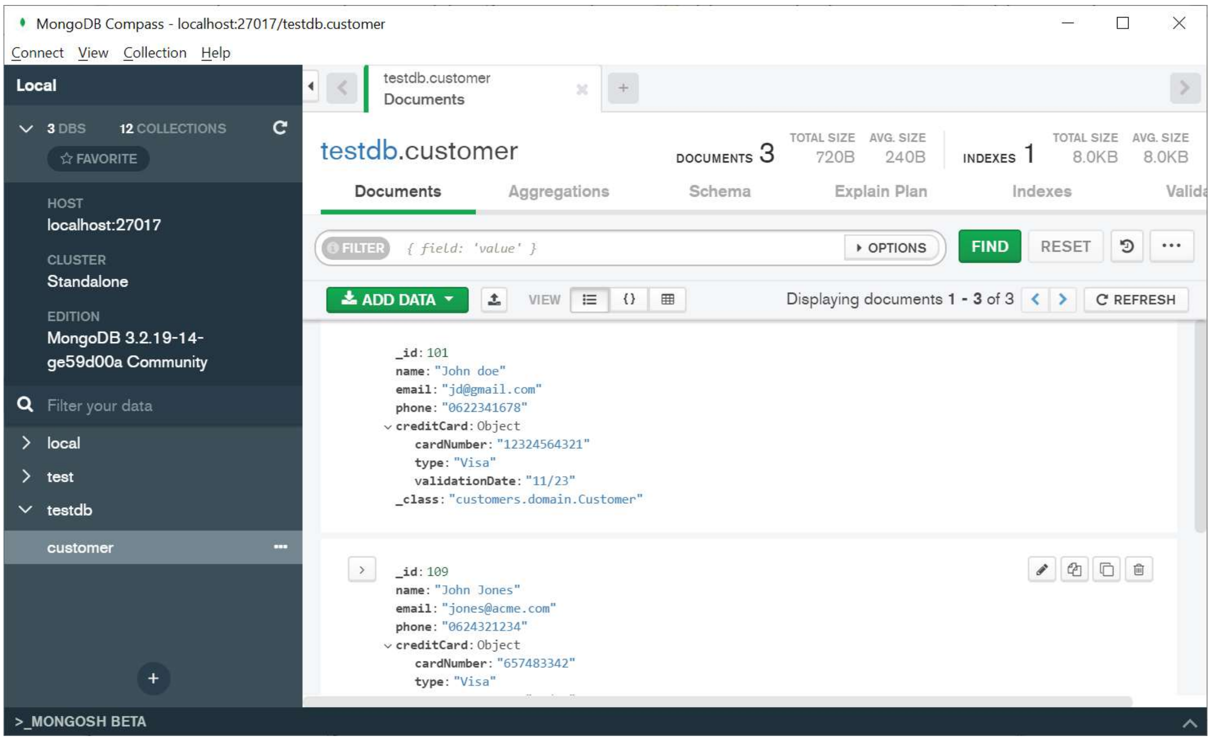
```
_id: 109
name: "John Jones"
email: "jones@acme.com"
phone: "0624321234"
creditCard: Object
   cardNumber: "657483342"
   type: "Visa"
   validationDate: "09/23"
_class: "customers.domain.Customer"
```

```
_id: 66
name: "James Johnson"
email: "jj123@acme.com"
phone: "068633452"
creditCard: Object
   cardNumber: "99876549876"
   type: "MasterCard"
   validationDate: "01/24"
_class: "customers.domain.Customer"
```

Modify the given **Lesson4SpringMongoDemo** application so that the application stores students in the database
A Student contains the attributes name, phoneNumber and email.
A student has also an address.
Create a class Address with the attributes street, city and zip.
In the application add 5 students in the database.
Then perform the following queries:

- Get all students
- Get all students with a certain name
- Get a student with a certain phoneNumber
- Get all students from a certain city

Check the data in the collection using Mongo compass.


MongoDB Compass - localhost:27017/testdb.customer
Connect View Collection Help
Local
3 DBS 12 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 3.2.19-14-ge59d00a Community
Filter your data
local
test
testdb
customer
testdb.customer Documents
testdb.customer
DOCUMENTS 3 TOTAL SIZE 720B AVG. SIZE 240B
INDEXES 1 TOTAL SIZE 8.0KB AVG. SIZE 8.0KB
Documents Aggregations Schema Explain Plan Indexes
FILTER { field: 'value' } OPTIONS FIND RESET
ADD DATA VIEW
Displaying documents 1 - 3 of 3
REFRESH
>_MONGOSH BETA

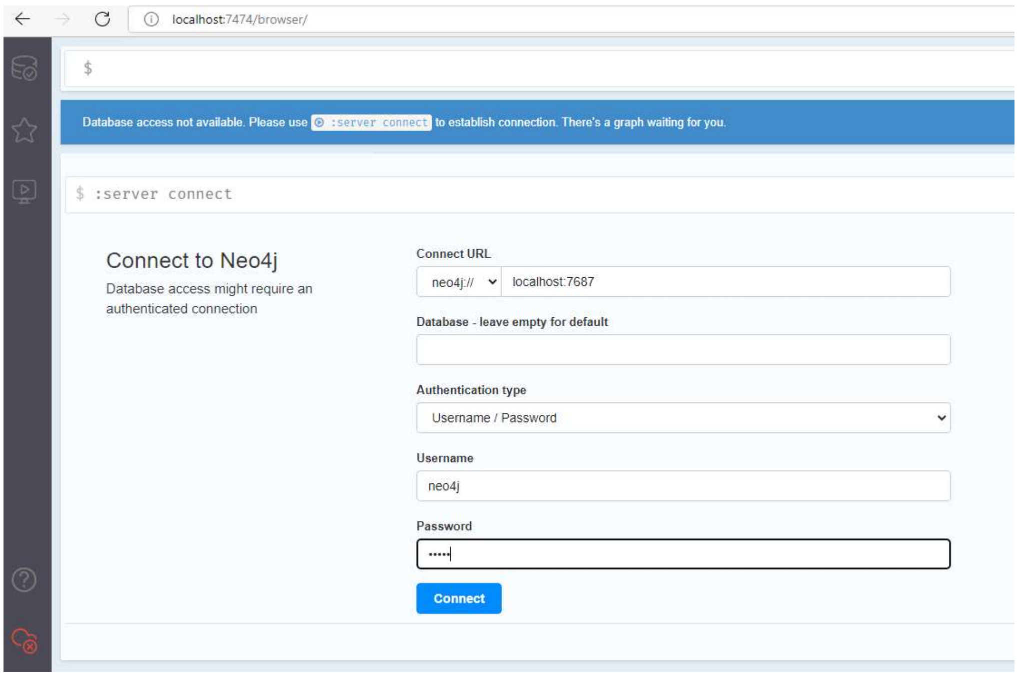
```
_id: 101
name: "John doe"
email: "jd@gmail.com"
phone: "0622341678"
creditCard: Object
    cardNumber: "12324564321"
    type: "Visa"
    validationDate: "11/23"
_class: "customers.domain.Customer"

_id: 109
name: "John Jones"
email: "jones@acme.com"
phone: "0624321234"
creditCard: Object
    cardNumber: "657483342"
    type: "Visa"
```

Part 3

Then start the neo4j database by double clicking the file **C:\SoftwareArchitecture\neo4j\bin\startNeo4j.bat.**

Then run the given project **Lesson4SpringNeo4j**.

Then open the browser at **localhost:7474/browser**


localhost:7474/browser/

$

Database access not available. Please use :server connect to establish connection. There's a graph waiting for you.

$:server connect

Connect to Neo4j

Database access might require an authenticated connection

Connect URL: neo4j:// localhost:7687

Database - leave empty for default:

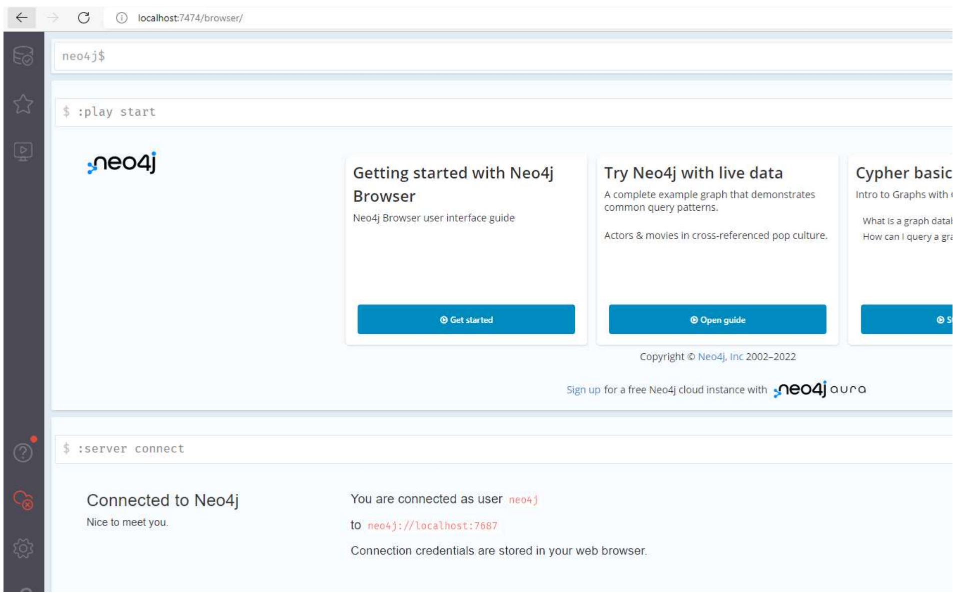
Authentication type: Username / Password

Username: neo4j

Password: •••••

Enter username **neo4j** and password **neo4j** and click **connect**.

It well then ask you to change the password. Change it **admin**


localhost:7474/browser/

neo4j$

$:play start

neo4j
Getting started with Neo4j Browser
Neo4j Browser user interface guide
Get started
Try Neo4j with live data
A complete example graph that demonstrates common query patterns.
Actors & movies in cross-referenced pop culture.
Open guide
Cypher basic
Intro to Graphs with
What is a graph datal
How can I query a gr

Copyright © Neo4j, Inc 2002–2022

Sign up for a free Neo4j cloud instance with neo4j aura

$:server connect

Connected to Neo4j

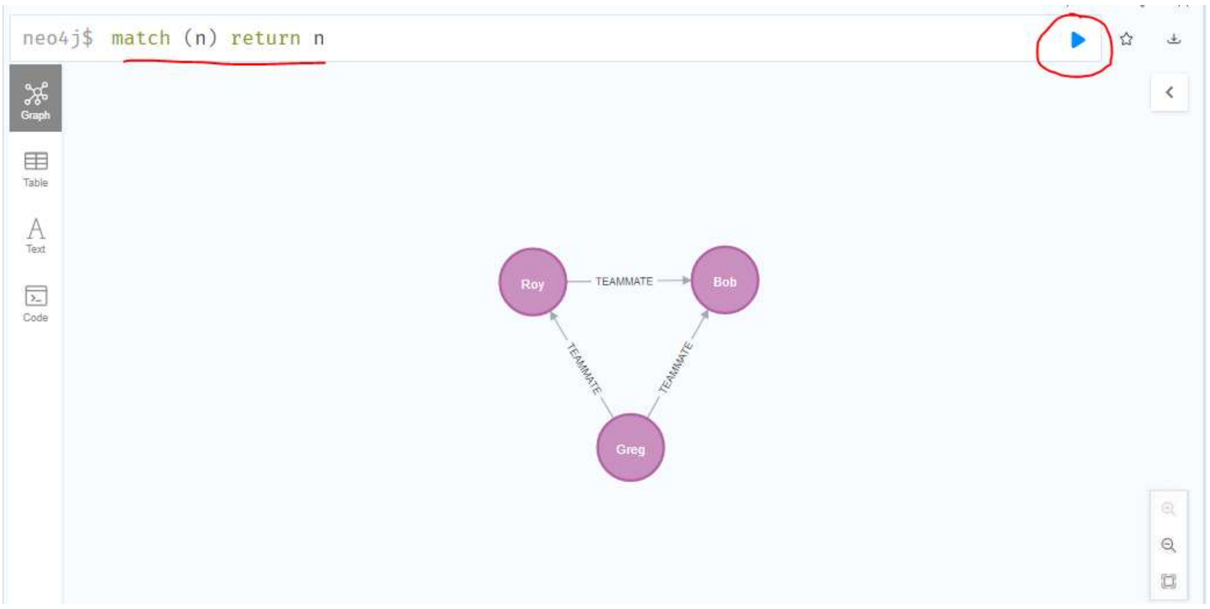
Nice to meet you.

You are connected as user neo4j

to neo4j://localhost:7687

Connection credentials are stored in your web browser.

In the query edit box, enter the query **match (n) return n** and click the **run** icon



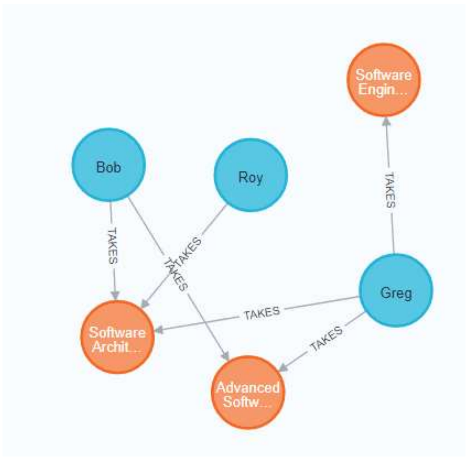
```
neo4j$ match (n) return n
```

You see now the created graph of persons

Modify the given **Lesson4SpringNeo4j** application so that the application stores students in the database including the courses they take.
A Student contains the attribute name.
A Course has the attributes courseNumber and name.

Then store some students with some courses in the database.

Check the data in the neo4j database



Now add the following method in the StudentRepository:

```
@Query("MATCH (s:Student) -[rel:TAKES] -> (c:Course {courseNumber: $coursenumber}) RETURN s")
List<Student> findByCourseNumber(String coursenumber);
```

Call the method and check if it works

Part 4

Suppose you need to store the following order in the database:

Ordernumber:122435
Orderdate 11/09/2021
Customer name: Frank Brown
Customer email: fbrown@gmail.com
Customer phone: 0623156543
Total price : 5160.00

quantity	Product number	Product name	price
2	A546	IPhone 12	980.00
4	S333	Samsung Galaxy 12S	800.00

1. Draw the tables including data that you need to store this order in a relational database.
2. Draw the collections including data that you need to store this order in a mongo database.
3. Draw the tables including data that you need to store this order in a cassandra database if you are interested in orders by customer.
4. Draw the database structure including data that you need to store this order in a neo4j database.

What to hand in?

1. A zip file of part 1
2. A zip file of part 2
3. A zip file of part 3
4. A PDF of part 4