Applied Databases

Neo4j I Exercise Sheet

Setup

- On the VM Neo4j is located here:

```
C:\Users\appDB\Documents\neo4j-community-5.3.0-windows\neo4j-community-5.3.0
```

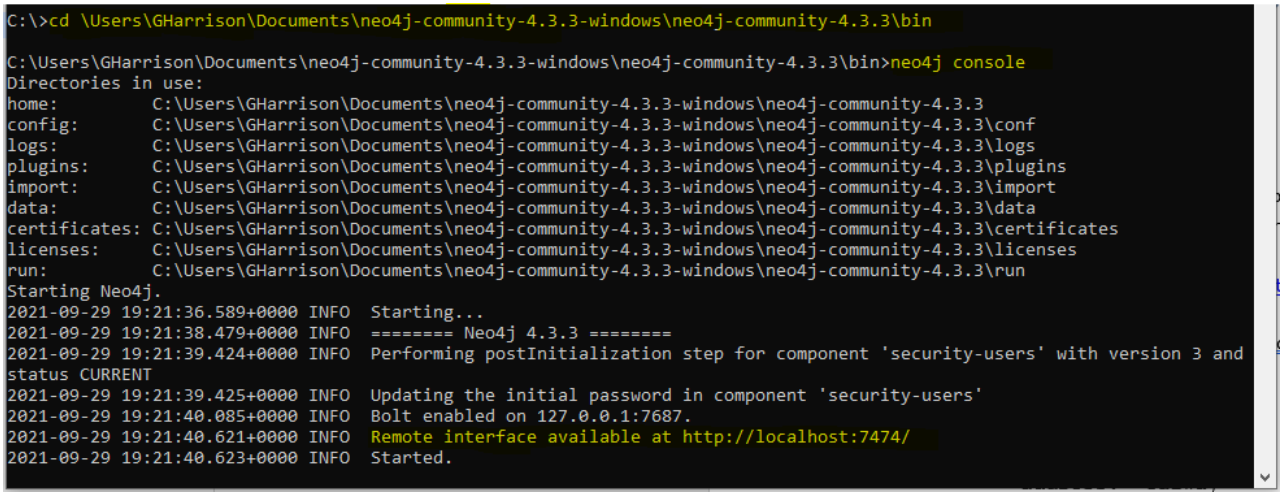
- *neo4j.conf* is located here:

```
C:\Users\appDB\Documents\neo4j-community-5.3.0-windows\neo4j-community-5.3.0\conf\neo4j.conf
```

- On the VM the Username and Password are:
  Username = `neo4j`
  Password = `neo4jneo4j`

Part 1

- Use a new database called *l6p1* (by updating *neo4j.conf*).
- Run Neo4j as follows:
  - Open a Windows Command prompt/PowerShell and change to the `bin` folder of the Neo4j installation.
  - Run `neo4j console`



- Then open a browser to http://localhost:7474 and select the database just created (*l6p1*).

- Create the following nodes with a label `:Student` with the following properties:
    - **name**: "Tom"
    - **sid**: "G001"
    - **age**: 23
    - **sex**: "M"
    - **address**: "Galway"
    - **hair**: "brown"
    - **email**: "tom@gmail.com"

    - **name**: "Sean"
    - **sid**: "G002"
    - **age**: 19
    - **sex**: "M"
    - **address**: "Galway"
    - **email**: sean@gmail.com

    - **name**: "Bob"
    - **sid**: "G003"
    - **age**: 22
    - **sex**: "M"
    - **address**: "Mayo"
    - **email**: "bob123@hotmail.com"
    - **twitter**: "@bob123"

    - **name**: "Mary"
    - **sid**: "G004"
    - **age**: 20
    - **sex**: "F"
    - **address**: "Mayo"
    - **hair**: "blonde"
    - **email**: "mary19@gmail.com"
    - **twitter**: "@mary19"
    - **snapchat**: "mary19"

    - **name**: "Alice"
    - **sid**: "G005"
    - **age**: 28
    - **sex**: "F"
    - **address**: "Roscommon"
    - **email**: "alice@hotmail.com"
    - **snapchat**: "alice123"

    - **name**: "Pat"
    - **sid**: "G006"
    - **age**: 24
    - **sex**: "M"
    - **address**: "Roscommon"
    - **email**: "pat@hotmail.com"
    - **twitter**: "patABC"

- Create the following nodes with a label `:Lecturer` with the following properties:
    - **name**: "Alan"
    - **sid**: "L001"
    - **age**: 57
    - **sex**: "M"
    - **address**: "Galway"
    - **email**: "alan@gmit.ie",
    - **twitter**: "@alan"

    - **name**: "Mary"
    - **sid**: "L002"
    - **age**: 47
    - **sex**: "F"
    - **address**: "Mayo"
    - **email**: "mary@gmit.ie
    - **hair**: "brown"

- Find the average age of Students, rounded to the nearest whole number.

- Show the name of each student and his/her age.

- Find the age of the youngest Student.

- Show the names of students who have a *twitter* attribute.

- Show the number of students who have a *twitter* attribute.

- Show the average of age of people in their 20s, 30s and 40s rounded to one decimal place.

- Show all the properties for the Student *Tom.*

- Increase everyone's age by 1.

- Return the name and age of all males living in Galway.

- Create the following nodes with both :Student and :Lecturer labels
    - `name: "Yvonne"`
    - `age: 37`
    - `sex: "F"`
    - `address: "Galway"`
    - `email: yvonne@gmit.ie`
    - `twitter: "@yv12"`

    - `name: "Walter"`
    - `age: 44`
    - `address: "Galway"`
    - `email: walter@gmit.ie`
    - `hair: "black"`

- Show the name, age and hair colour of everyone who is both a Student and a Lecturer.

- Update the *twitter* attribute of all lectures to have GMIT after their existing twitter name. E.g. "@alan" should become "@alanGMIT".

- Find the average age of Males and find the youngest Male(s).
  Then return the name (as Name) and age (as *Age*) of the youngest Male(s) as well as the average age of Males (as *AverageAge*) and the difference in age between the youngest Male(s) and the average age (as *Difference*).

  E.g., If the average age of Males was 30, and the youngest Male was called “Tony” aged 20, the following should be returned:

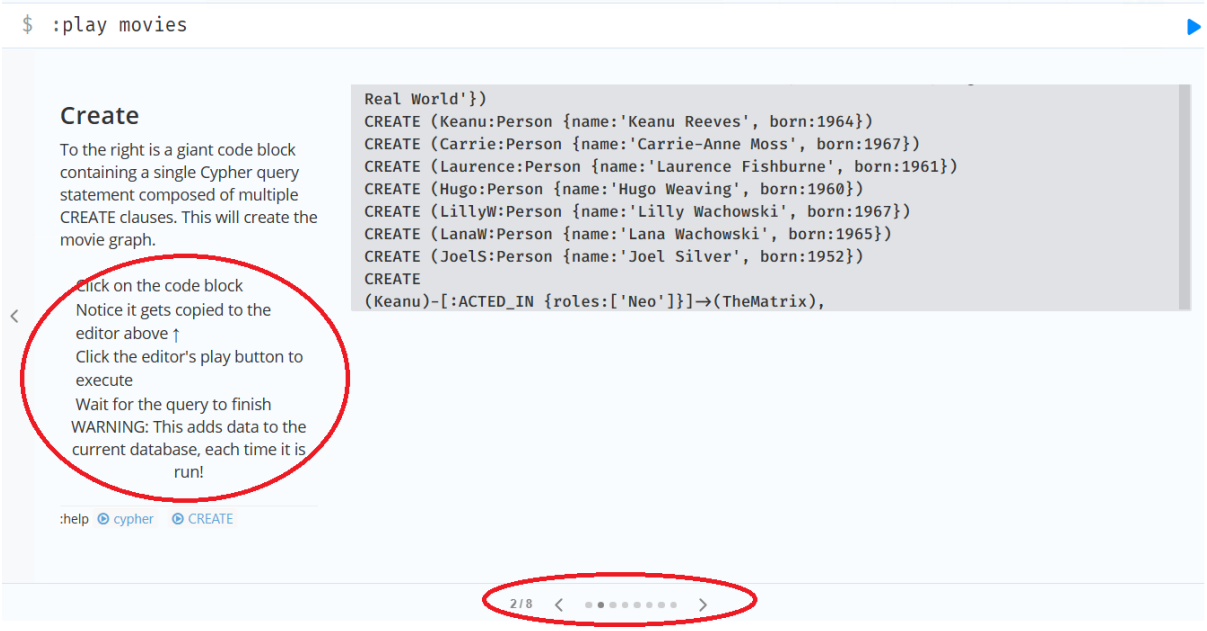
Name	**Age**	**AverageAge**	**Difference**
Tony	20	30	10

Part 2

- Use a new database, *I6p2*, (by updating neo4j.conf).
- In the Neo4j Browser, select the new database and type the following command:

```
tdb$ :play movies
```

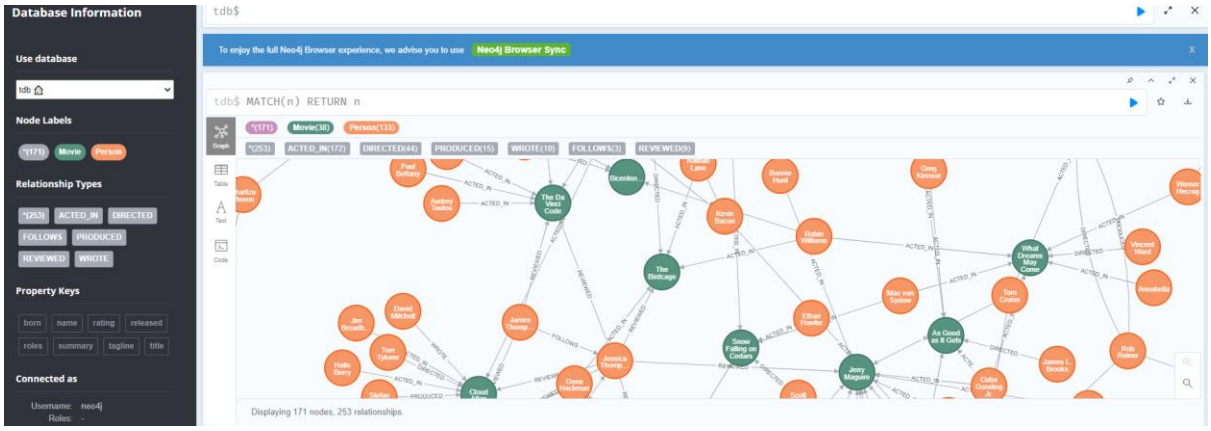
- This will return the following:



- Go to page 2 and follow steps circled above.

- A series of 171 nodes (representing Movies and People) and 253 relationships (such as ACTED_IN, DIRECTED etc.) between the nodes should now be created.

- Type MATCH(n) RETURN n to see all nodes and relationships:



- Show each movie node for movies that were released between 2000 and 2010

- Set an attribute called olderThan70 to true for all Persons born in the 1930s.

- Show the movie title and the year it was released for the first 10 movies in alphabetical order.

- Show the unique years in which movies were released in chronological order.

- Show the title and tagline for movies released in 1999.

- Show the names of the people (as *People)* and the year they were born (as *YOB*) for everyone older than “Robin Williams”.

- Show the number of movies released in 2006 (as *Releases_in_2006)*.

- Show the name (as *Name*) and born (as *YOB*) the youngest Person(s).