COMP3008 - Big Data Analytics

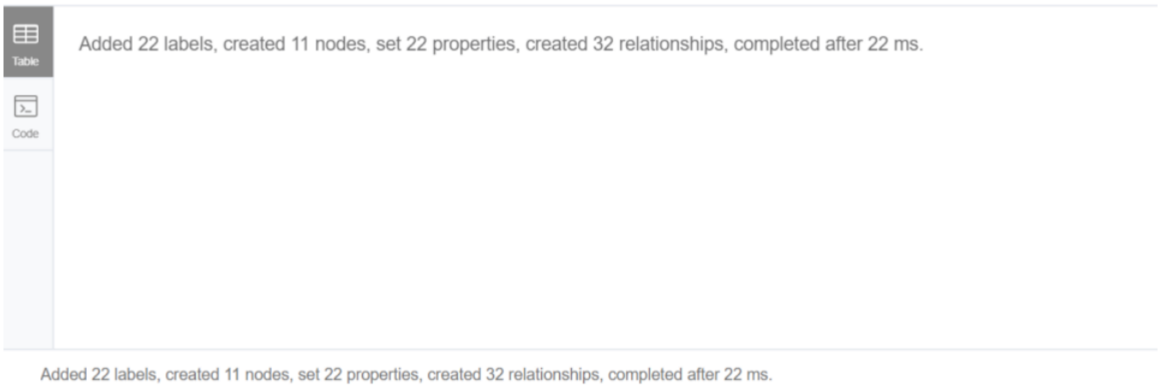
Workshop 5 - Influence Scoring

This practical constitutes our second approach to recommender systemscomputer systems that make recommendations are known as recommender systems. You will be implementing the *influence scoring* algorithm to identify friend recommendations for the characters appearing in the tragedy of Romeo and Juliet. Note that the practical employs the same collection of characters (11 in total) used in the previous practical to facilitate the testing of different options on a small dataset.Your solutions should match that which we calcualted during the lecture.

Uploading the Dataset

As in the case of the previous practical session, you will populate your database using a series of commands listed in the file `romeo_and_juliet_graph.txt`, which is available on the DLE. This file is known as a script, as it comprises multiple clauses involving the CREATE command. Make sure you understand how CREATE works and ask for help if you require clarification on this..

Click on the play button (*cypher run button*). Provided you copied and pasted the commands in the file correctly, you should be able to create 11 *characters*, 3 of them belonging to the *House of Montague*, 4 to the *House of Capulet*, 3 to the *House of Verona*, and 1 *Franciscan*. You should also create 32 `FRIENDS_WITH` relationships.



```
Added 22 labels, created 11 nodes, set 22 properties, created 32 relationships, completed after 22 ms.
```

```
Added 22 labels, created 11 nodes, set 22 properties, created 32 relationships, completed after 22 ms.
```

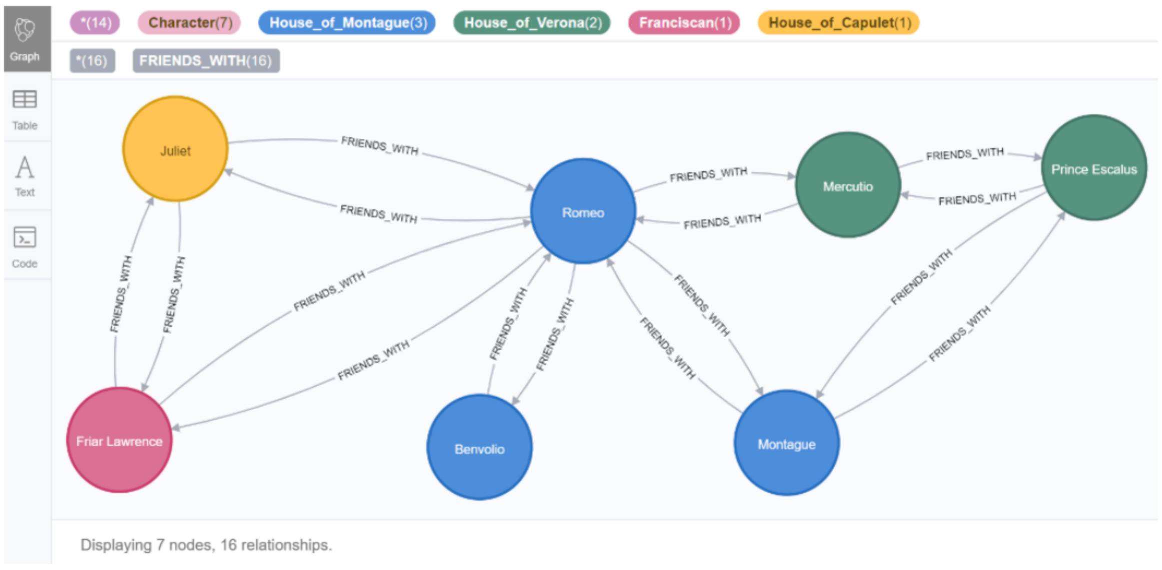
Familiarise yourself with the dataset and the relationships you have created. For example, count the total number of characters belonging to the House of Montague (3)you can do this by creating a collection composed by the characters in the House of Montague and then calculate the size of the collection:

```
MATCH (c:Character:House_of_Montague) RETURN COUNT(c)
```

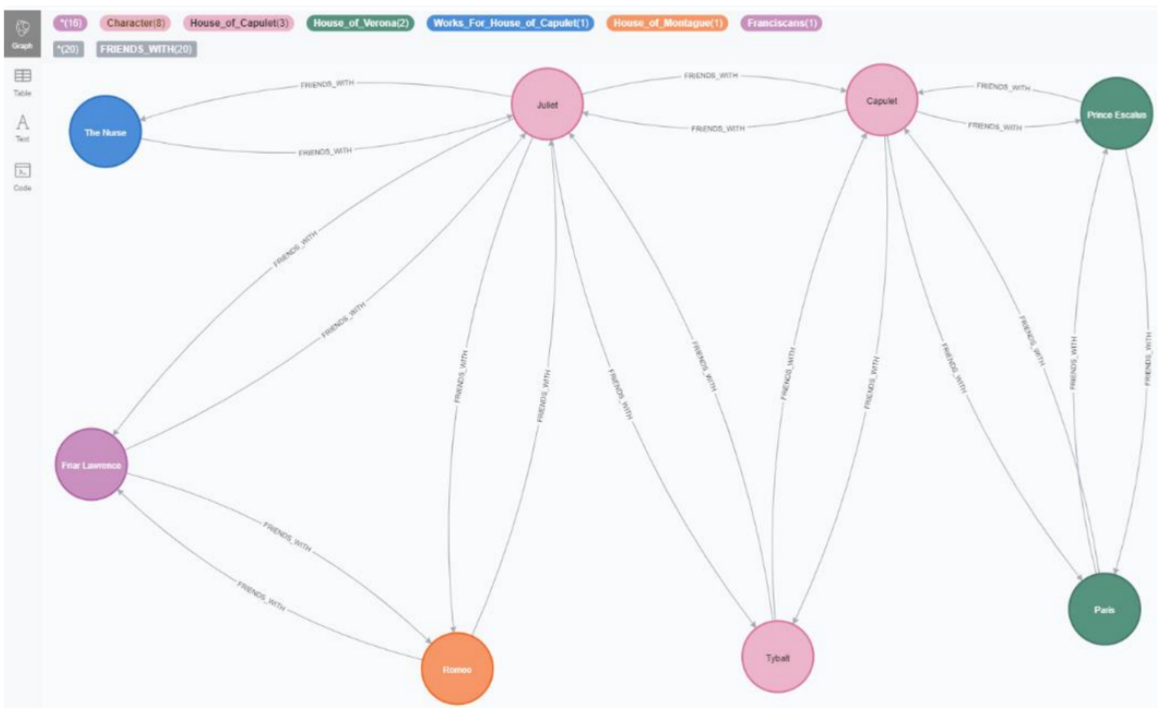
Can you explain why the following command produces the same result? If not, ask for clarification.

```
MATCH (c:House_of_Montague)
RETURN size(COLLECT(c))
```

Display on the screen a graph showing all the characters belonging to the House of Montague and all the relationships among them and the other characters in the graph. The results should be similar to the following picture.



Following the same approach you employed in the previous query, display a graph showing all the characters belonging to the House of Capulet, and all the relationships between them and the rest of the characters in the graph. Your results should look like the following picture.

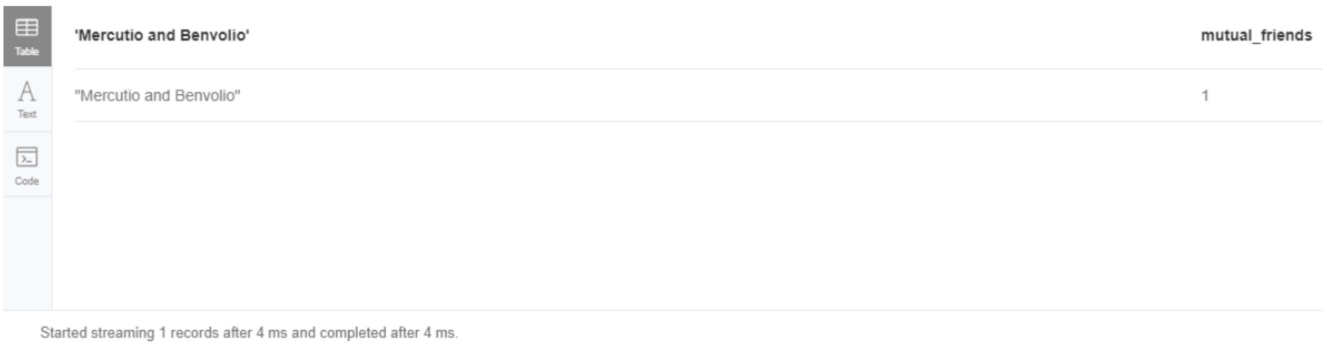


Let us incorporate the above idea into a friend recommendation algorithm. Here is the concrete way to do sonote that we call the technique influence scoring: Suppose that PersonA and PersonB have three friends in common: Friend1, Friend2, and Friend3. When recommending friends by influence, the score for PersonB as a friend of PersonA is

$$\frac{1}{\text{numfriends(Friend1)}} + \frac{1}{\text{numfriends(Friend2)}} + \frac{1}{\text{numfriends(Friend3)}}$$

where numfriends(f) is the number of friends that f has. In other words, each friend F of PersonA has a total influence score of 1 to contribute, and divides it equally among all of their friends.

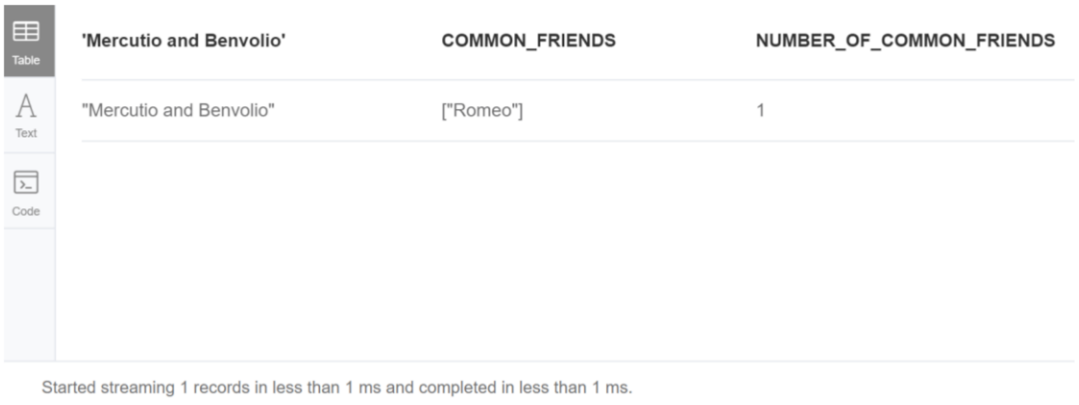
For the purpose of testing this algorithm in the practical session, start by counting the number of mutual friends between Mercutio and Benvolio. Make sure the names 'Mercutio and Benvolio' are followed by the number of mutual friends between them. Your results should look like the following picture.



'Mercutio and Benvolio'	mutual_friends
"Mercutio and Benvolio"	1

Started streaming 1 records after 4 ms and completed after 4 ms.

Produce a query to obtain not only the number of mutual friends between Mercutio and Benvolio (1), but also the names of the mutual friends (Romeo). Your results should look like the following picture.



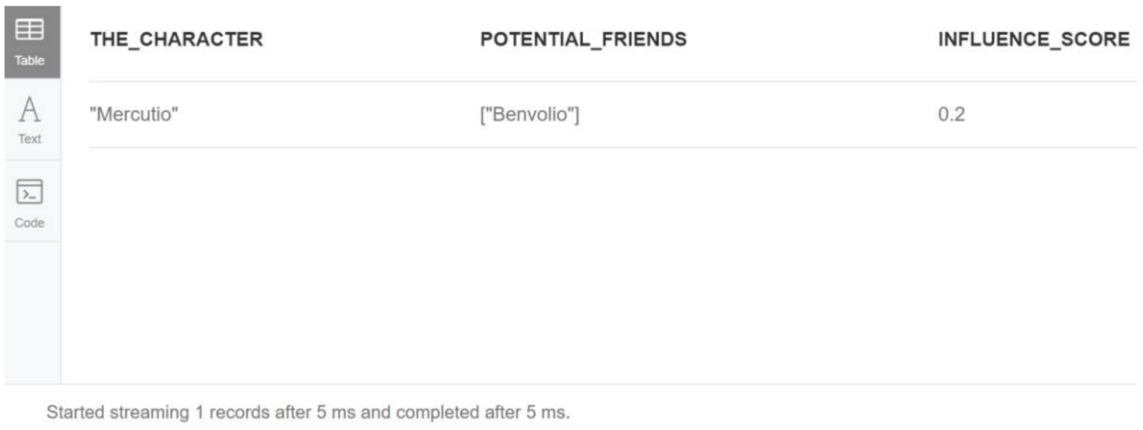
'Mercutio and Benvolio'	COMMON_FRIENDS	NUMBER_OF_COMMON_FRIENDS
"Mercutio and Benvolio"	["Romeo"]	1

Started streaming 1 records in less than 1 ms and completed in less than 1 ms.

Now that we have identified the mutual friends between Mercutio and Benvolio (Romeo), we can determine a value to indicate how good Benvolio is as a recommendation for Mercutio. The value is known as the influence score, and it is calculated as follows (review the lecture slides if you need clarification before proceeding):

$$\frac{1}{\text{numfriends(Romeo)}} = \frac{1}{5} = 0.2$$

Produce a query to calculate how good Benvolio is as a recommendation for Mercutio. Your results should look like the following picture.



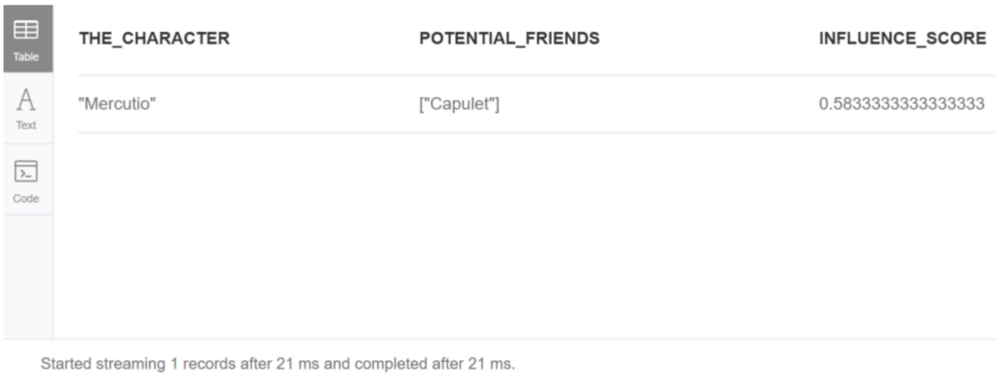
THE_CHARACTER	POTENTIAL_FRIENDS	INFLUENCE_SCORE
"Mercutio"	["Benvolio"]	0.2

Started streaming 1 records after 5 ms and completed after 5 ms.

Following the same procedure, we will calculate how good Capulet is as a recommendation for Mercutio. First, we have to identify the common friends between Mercutio and Capulet: Paris and Prince Escalus. Then, the influence score of Capulet as a recommendation for Mercutio is determined as

$$\frac{1}{\text{numfriends(Paris)}} + \frac{1}{\text{numfriends(Prince Escalus)}} = \frac{1}{3} + \frac{1}{4} \approx 0.5833$$

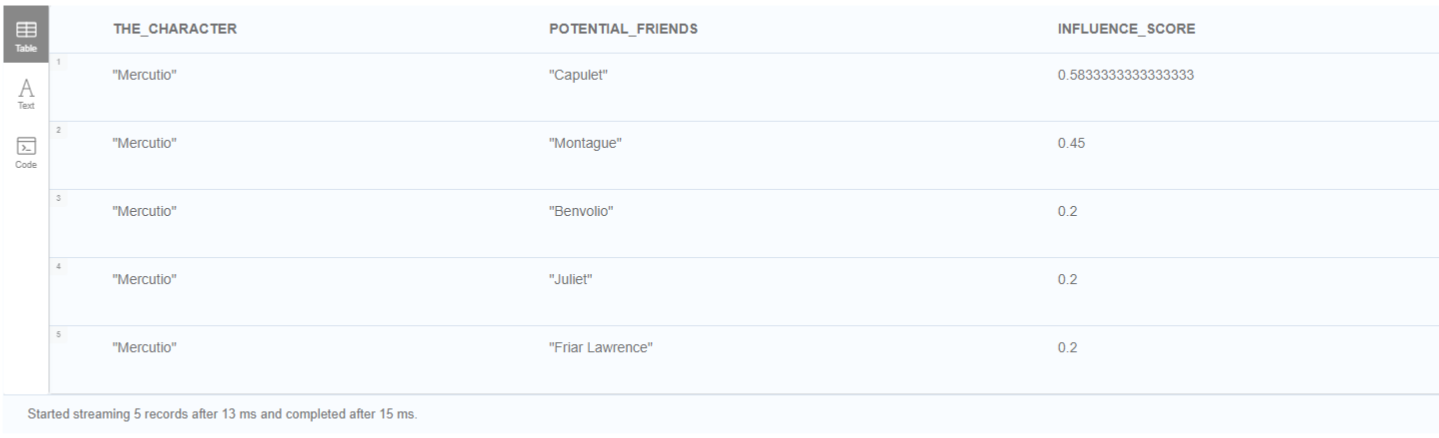
Produce a query to calculate how good Capulet is as a recommendation for Mercutio. Your results should look like the following picture.



THE_CHARACTER	POTENTIAL_FRIENDS	INFLUENCE_SCORE
"Mercutio"	["Capulet"]	0.5833333333333333

Started streaming 1 records after 21 ms and completed after 21 ms.

Finally, you should produce a generic CQL query to obtain the friends in common and the influence score for each character in Romeo and Juliet who is not yet a friend of Mercutio. It is not necessary to list the names of characters who do not have any mutual friends with Mercutio. However, make sure you sort the list by influence score in *descending* order (the characters with the largest influence should appear at the top). Your results should look similar to the following picture.



	THE_CHARACTER	POTENTIAL_FRIENDS	INFLUENCE_SCORE
1	"Mercutio"	"Capulet"	0.5833333333333333
2	"Mercutio"	"Montague"	0.45
3	"Mercutio"	"Benvolio"	0.2
4	"Mercutio"	"Juliet"	0.2
5	"Mercutio"	"Friar Lawrence"	0.2

Started streaming 5 records after 13 ms and completed after 15 ms.