LAB TASK 5

Firstly, I have setup account on neo4j sandbox and chose the pre-loaded dataset of movie.

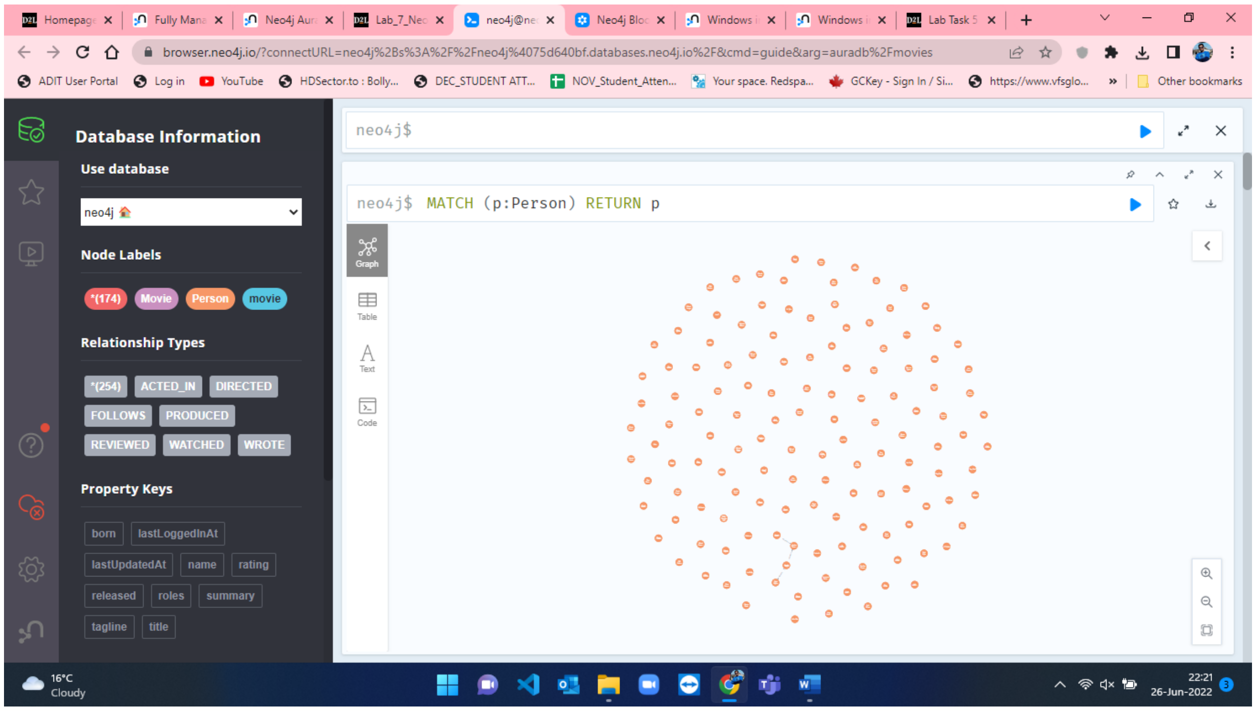
The default dataset is imported with all the required Nodes, Labels, Properties, and Relationships.

I have explored through the in-built tutorial and run some sample queries.

Then, I have use some of the assumptions and created and run my own queries.

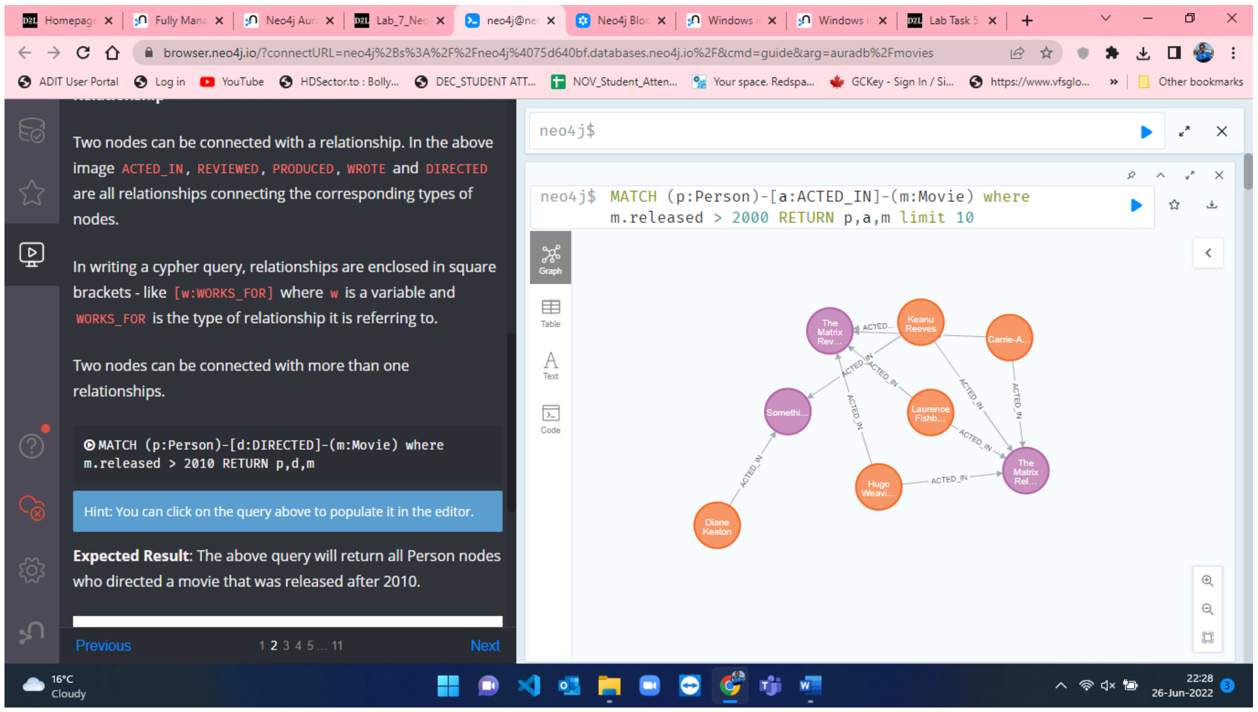
The sample queries along with the output screenshot is mentioned below.

1) This cypher query represents all the nodes in the database.
   MATCH (p:Person) RETURN p



```
neo4j$ MATCH (p:Person) RETURN p
```

2) This cypher query represents all the Person nodes who directed a movie that was released after 2000. I have limited the number of result to 10.
   MATCH (p:Person)-[a:ACTED_IN]-(m:Movie) where m.released > 2000 RETURN p,a,m limit 10



Two nodes can be connected with a relationship. In the above image `ACTED_IN`, `REVIEWED`, `PRODUCED`, `WROTE` and `DIRECTED` are all relationships connecting the corresponding types of nodes.

In writing a cypher query, relationships are enclosed in square brackets - like `[w:WORKS_FOR]` where `w` is a variable and `WORKS_FOR` is the type of relationship it is referring to.

Two nodes can be connected with more than one relationships.

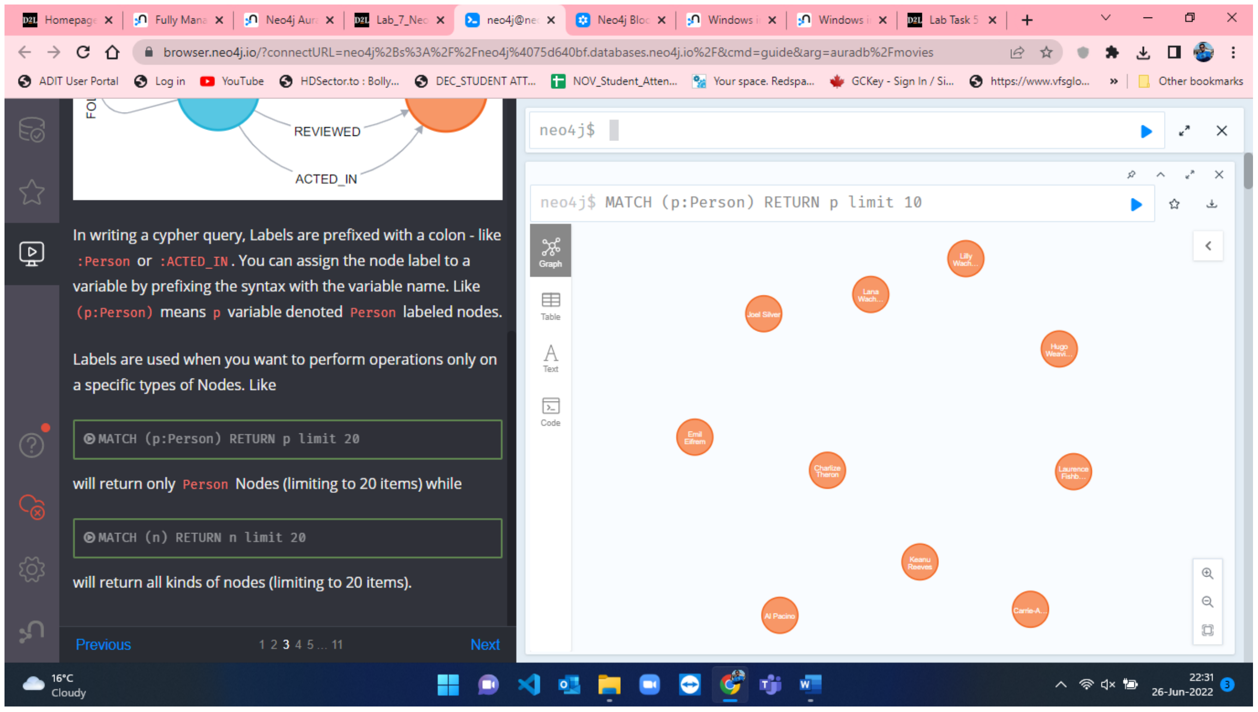
```
MATCH (p:Person)-[d:DIRECTED]-(m:Movie) where
m.released > 2010 RETURN p,d,m
```

Hint: You can click on the query above to populate it in the editor.

Expected Result: The above query will return all Person nodes who directed a movie that was released after 2010.

```
neo4j$ MATCH (p:Person)-[a:ACTED_IN]-(m:Movie) where
m.released > 2000 RETURN p,a,m limit 10
```

3) This cypher query represents all the person nodes with limit of 10 results.
MATCH (p:Person) RETURN p limit 10



In writing a cypher query, Labels are prefixed with a colon - like `:Person` or `:ACTED_IN`. You can assign the node label to a variable by prefixing the syntax with the variable name. Like `(p:Person)` means `p` variable denoted `Person` labeled nodes.

Labels are used when you want to perform operations only on a specific types of Nodes. Like

```
MATCH (p:Person) RETURN p limit 20
```

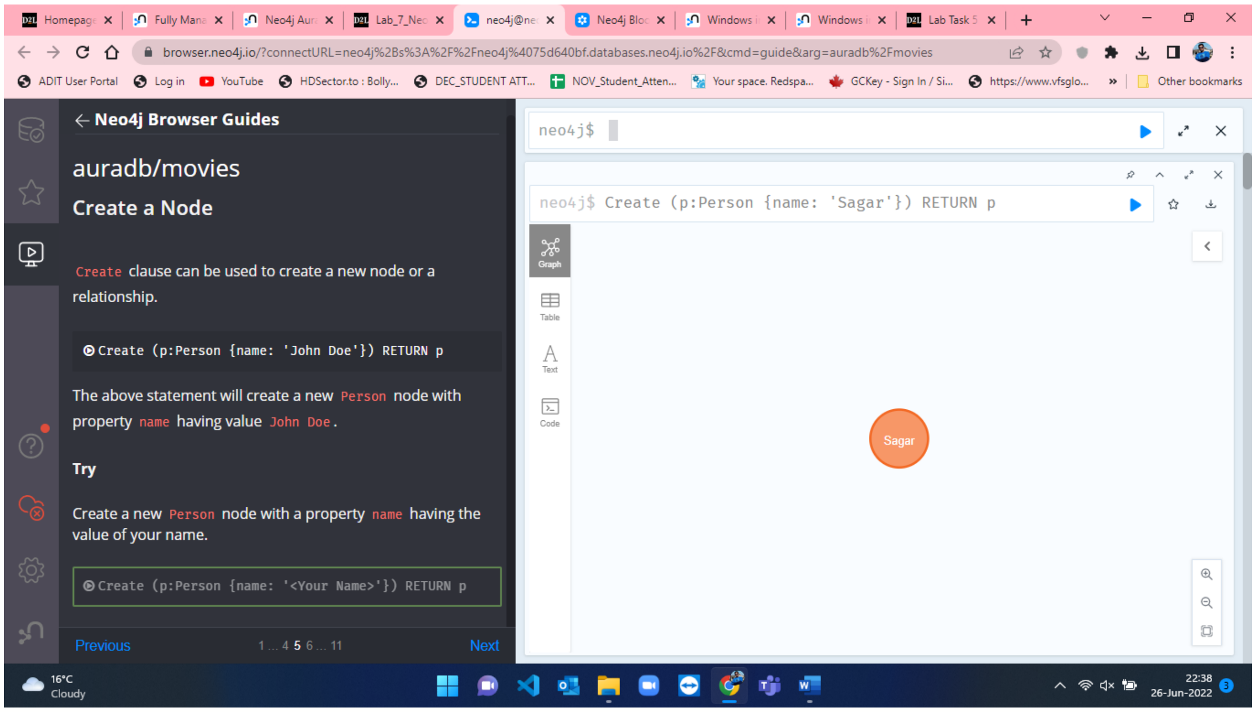
will return only `Person` Nodes (limiting to 20 items) while

```
MATCH (n) RETURN n limit 20
```

will return all kinds of nodes (limiting to 20 items).

```
neo4j$ MATCH (p:Person) RETURN p limit 10
```

4) This query creates the Person node of name sagar
CREATE (p:Person {name: 'Sagar'}) RETURN p



auradb/movies

Create a Node

`Create` clause can be used to create a new node or a relationship.

```
Create (p:Person {name: 'John Doe'}) RETURN p
```

The above statement will create a new `Person` node with property `name` having value `John Doe`.

Try

Create a new `Person` node with a property `name` having the value of your name.

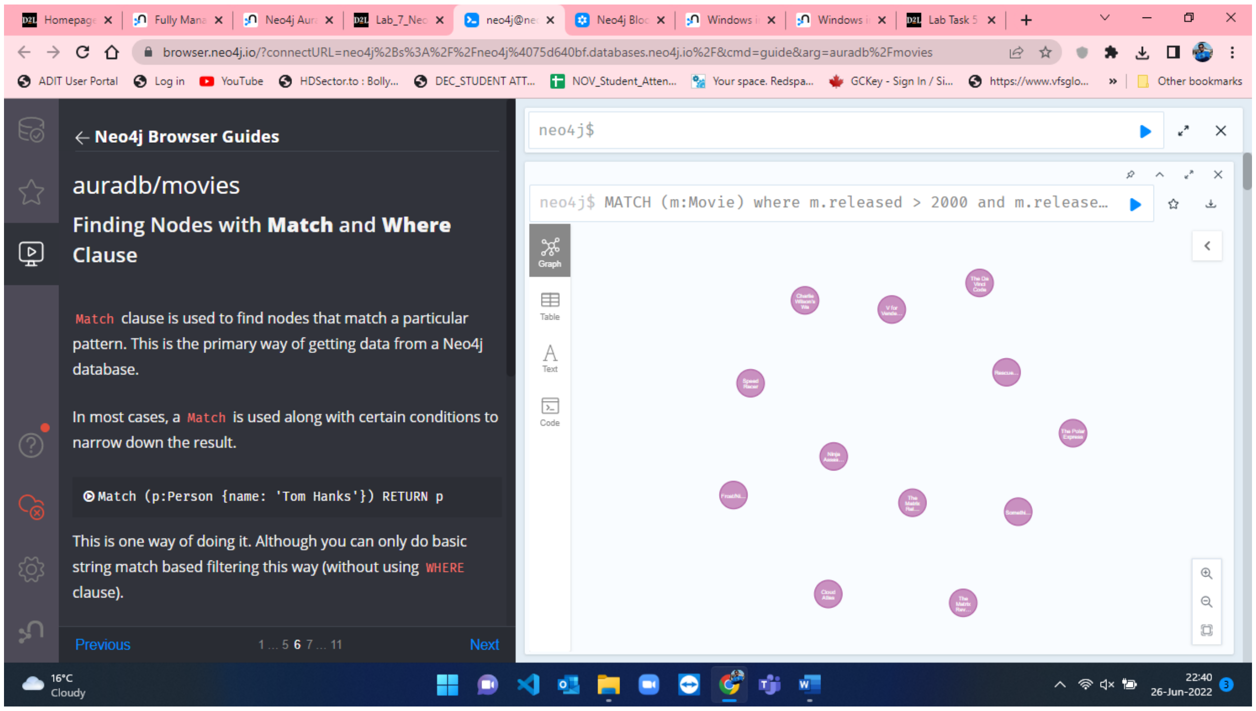
```
Create (p:Person {name: '<Your Name>'}) RETURN p
```

```
neo4j$ Create (p:Person {name: 'Sagar'}) RETURN p
```

5) Finding nodes by using MATCH

This query will get all the movies that were released between year 2000 and 2020.

MATCH (m:Movie) where m.released > 2000 and m.released < 2020 RETURN m



auradb/movies

Finding Nodes with Match and Where Clause

`Match` clause is used to find nodes that match a particular pattern. This is the primary way of getting data from a Neo4j database.

In most cases, a `Match` is used along with certain conditions to narrow down the result.

```
Match (p:Person {name: 'Tom Hanks'}) RETURN p
```

This is one way of doing it. Although you can only do basic string match based filtering this way (without using `WHERE` clause).

```
neo4j$ MATCH (m:Movie) where m.released > 2000 and m.release…
```