# DIFFERENT WAYS YOU CAN USE NEO4J DB

Below are given three different ways that you can start working with neo4j;

* Sandbox:

Neo4j Sandbox is a cloud-based environment that allows users to try out and experiment with Neo4j, a popular graph database, without the need for local installation. It provides a pre-configured instance of Neo4j, along with various tools and resources, such as tutorials, guides, and sample datasets.

Though it is a one of the great options yet it has some limitations such as larger datasets or more complex queries may not perform as well in the sandbox environment and users have limited options for customizing the settings and configuration of their Neo4j instance.

* Desktop:

Neo4j Desktop is a desktop application that provides a graphical user interface (GUI) for working with Neo4j, a popular graph database system. It is designed to make it easier for developers, data scientists, and other users to create, manage, and monitor Neo4j databases and applications. The problem with it is its complexity of usage.

* Aura DB:

Aura DB is a cloud-based graph database service that allows users to create and manage their own graph databases without having to worry about infrastructure management, scaling, or maintenance.

Aura DB is one of the most powerful and convenient platform to work with neo4j on cloud and it is being used widely in database management.

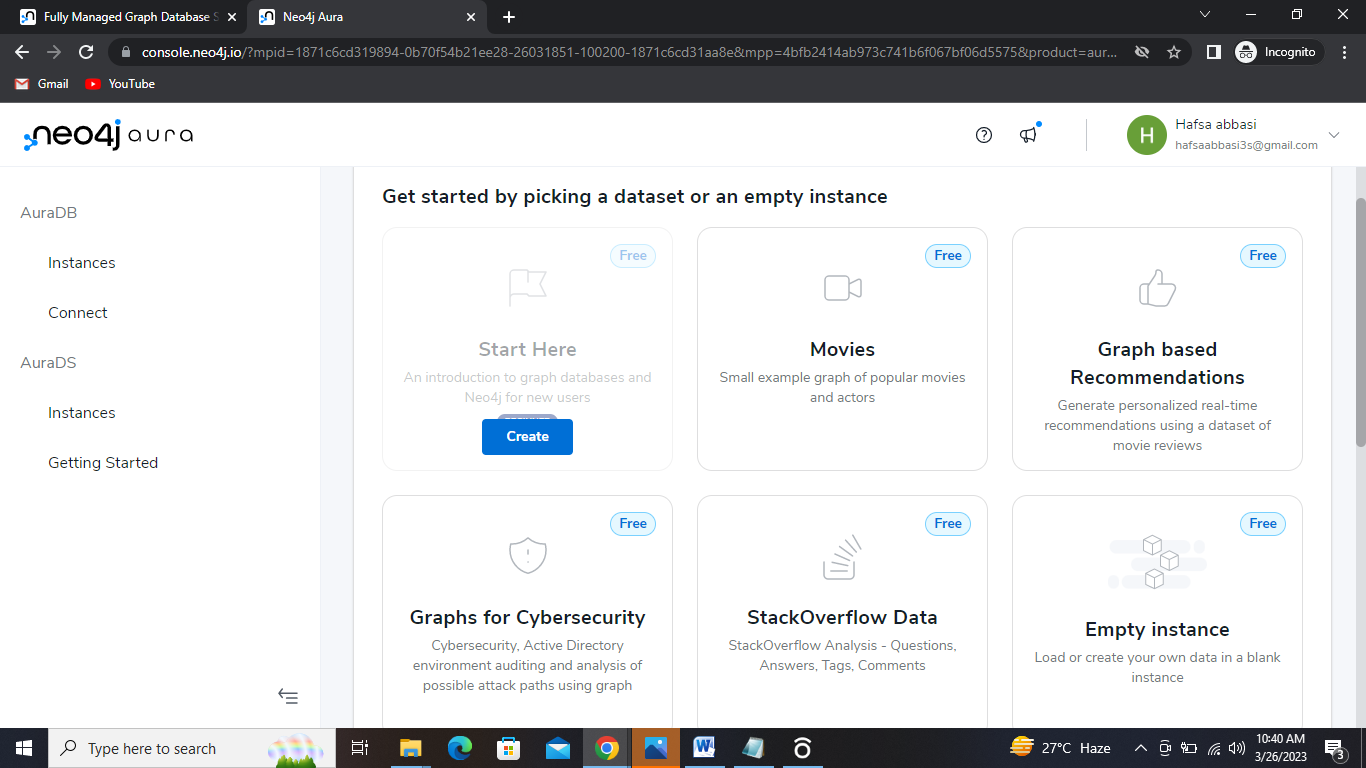
# USING NEO4J AURA DB

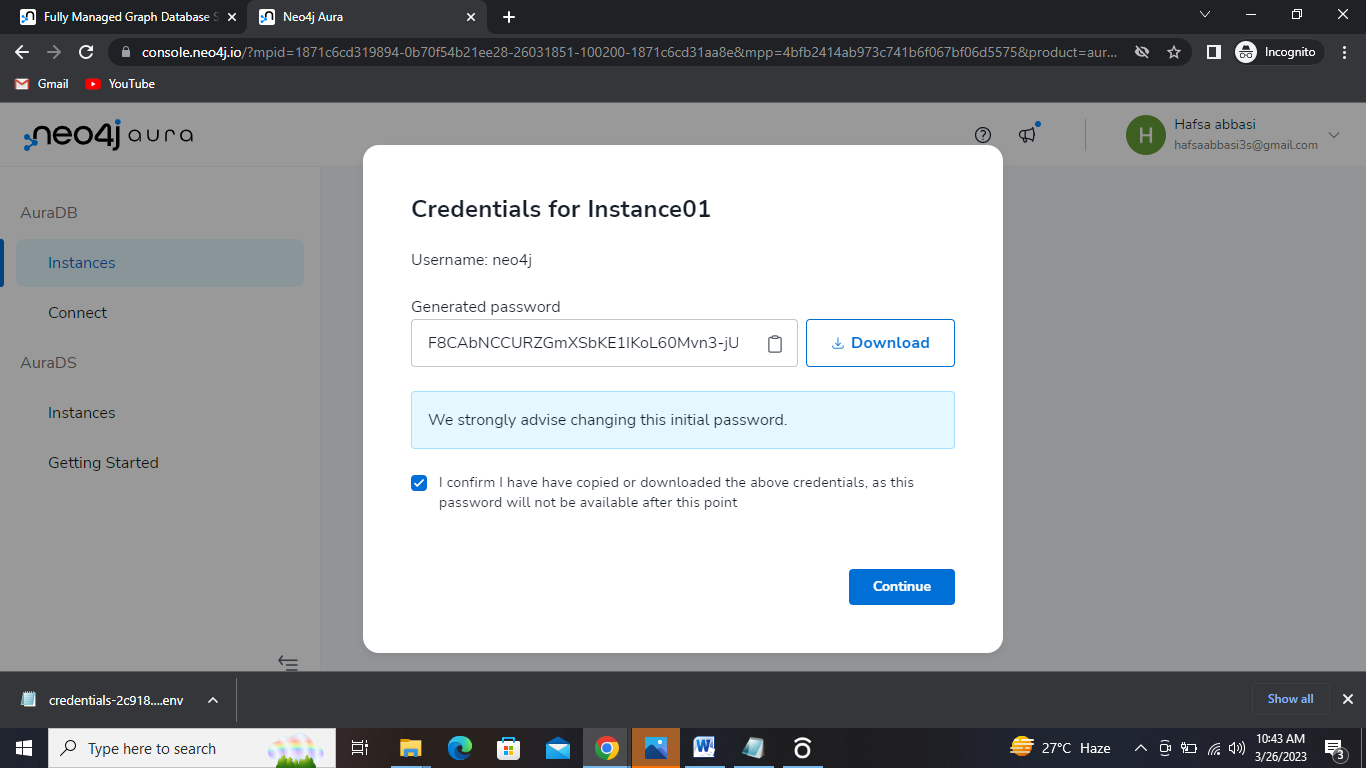
* **Step 01: To create your account**

1. Go to Neo4j aura DB official website <https://neo4j.com/cloud/aura/>
2. Click on “Start Free”
3. Signup with your email and password.
4. Select I agree once you have read the Terms of Service and Privacy Policy.

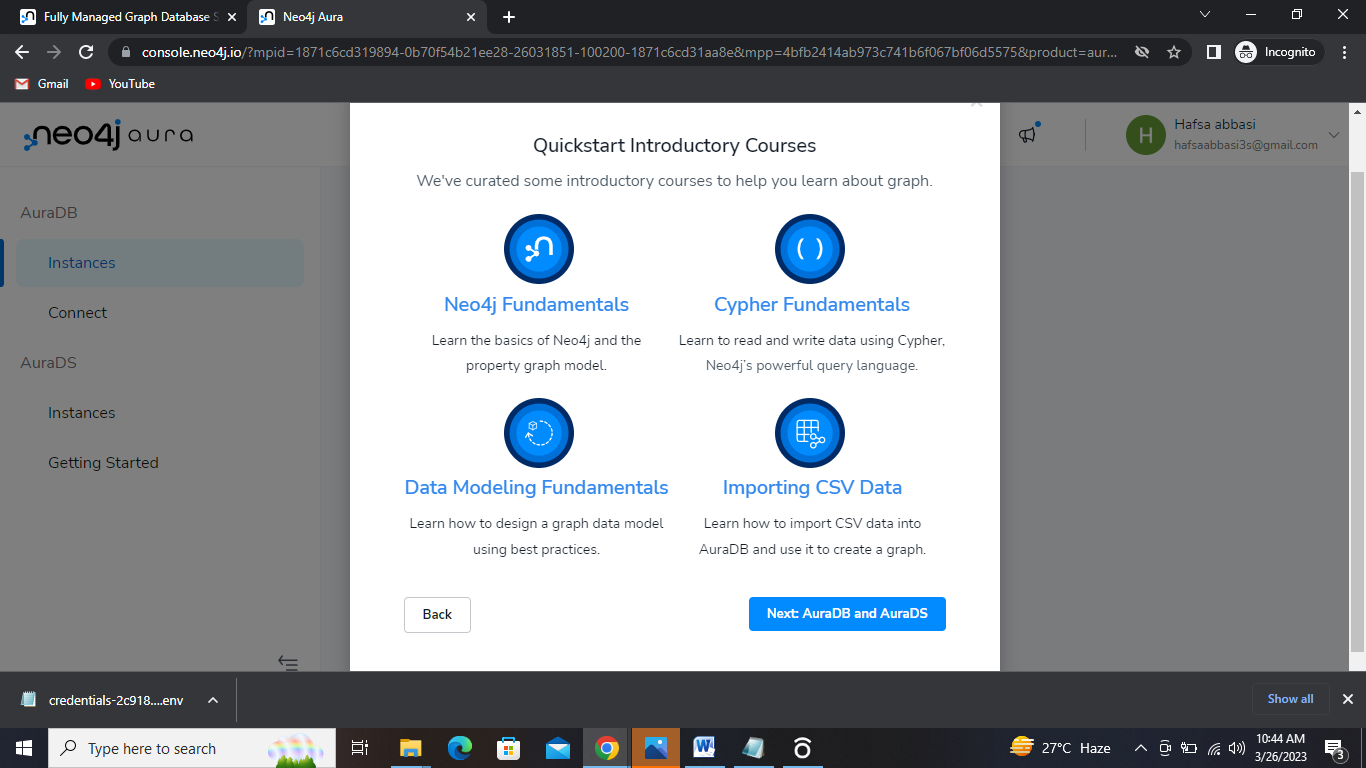
* **Step 02 :Set up your First instance**

1. You can either select the given dataset or empty instance to create your own.
2. You should download the credential file (it contains password) and continue.

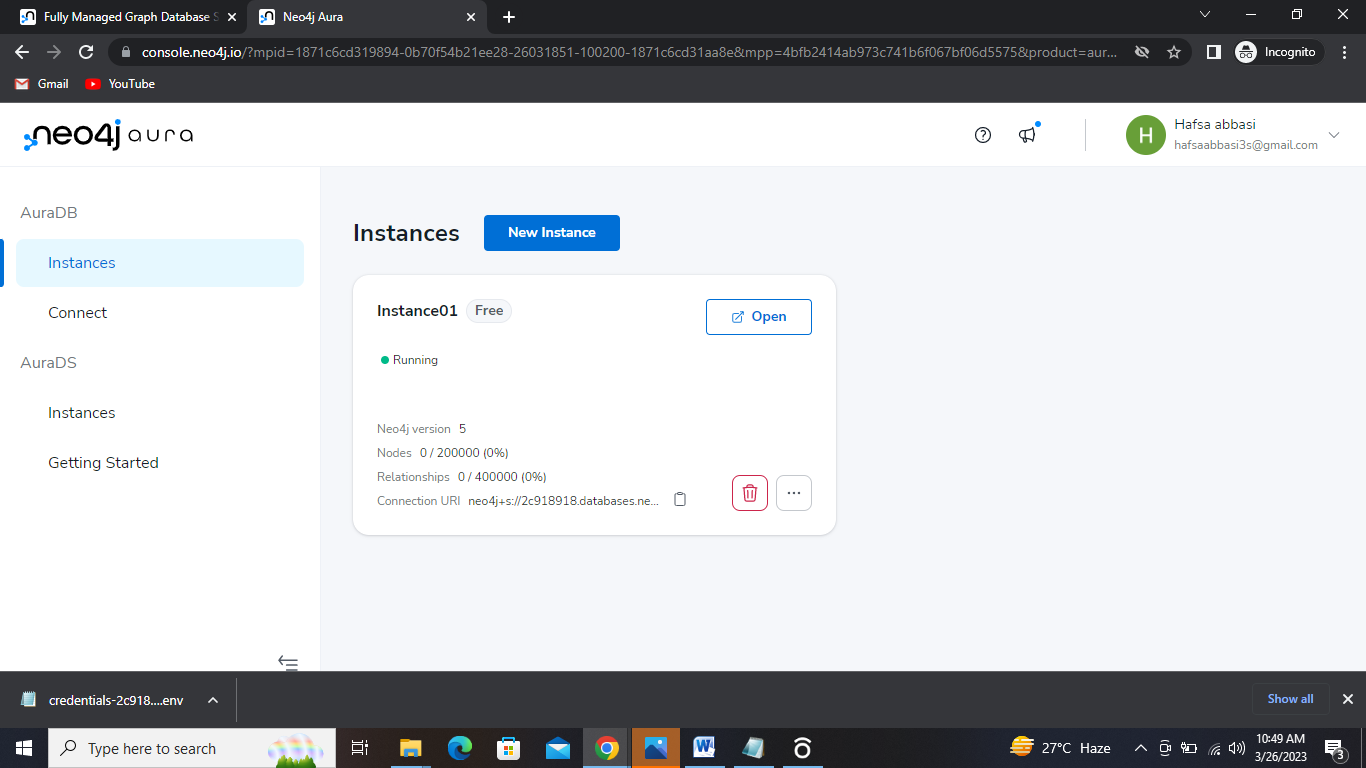




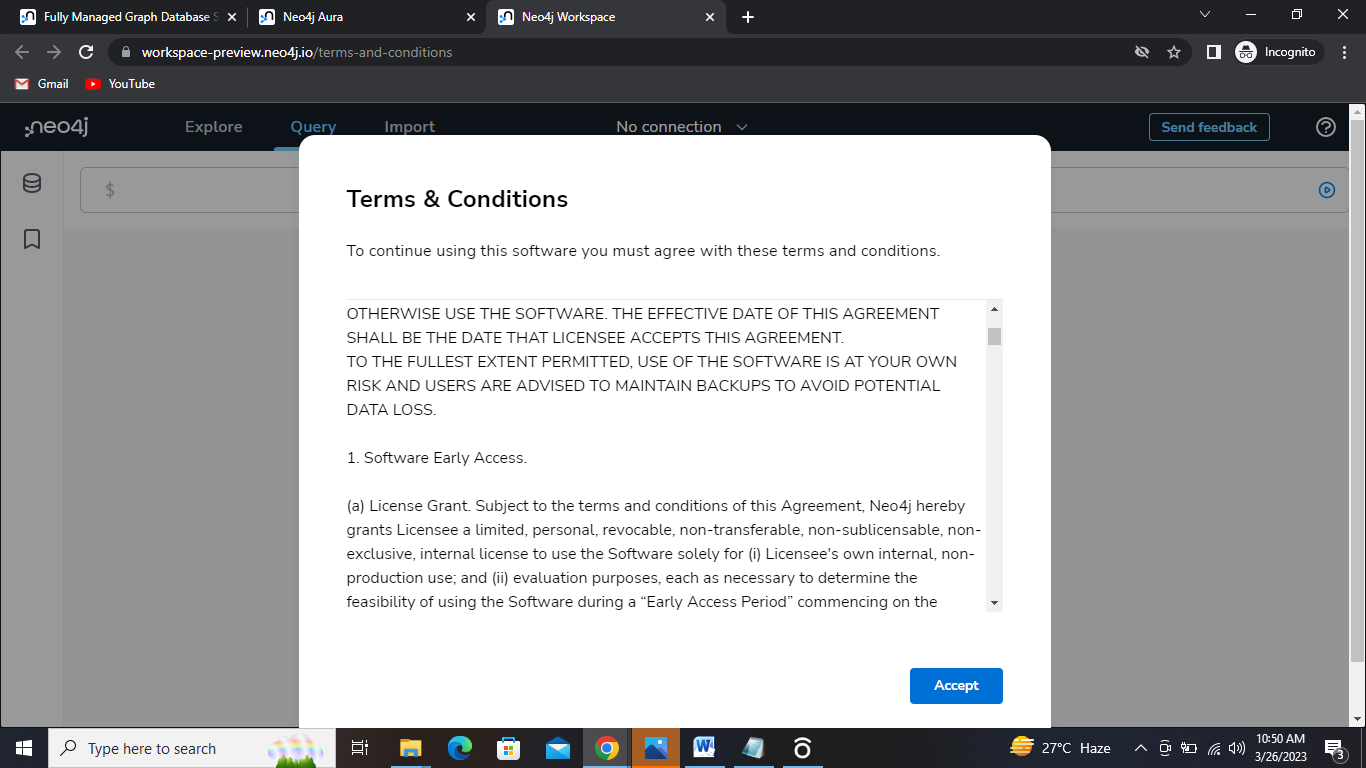
1. You can take introductory courses



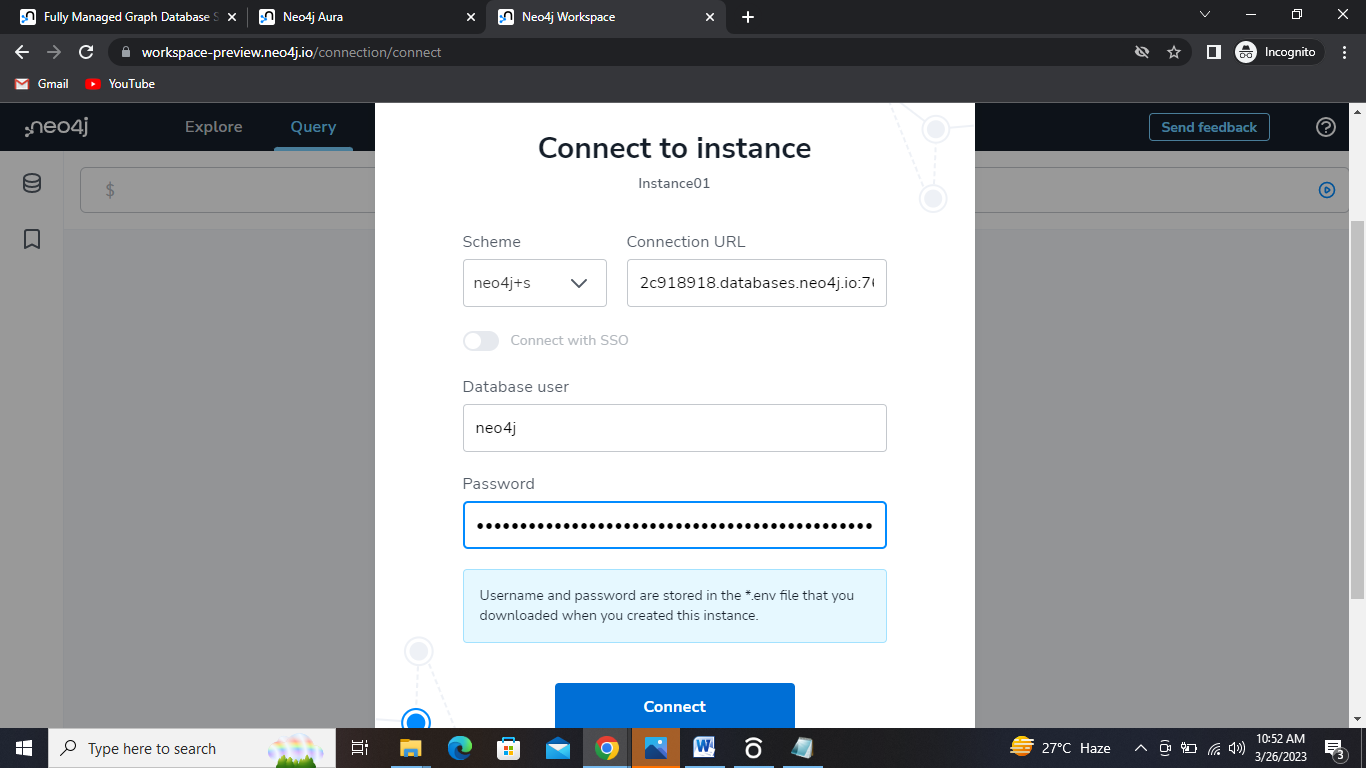
1. Your instance is created. Click on open button and you will go to aura workspace page.



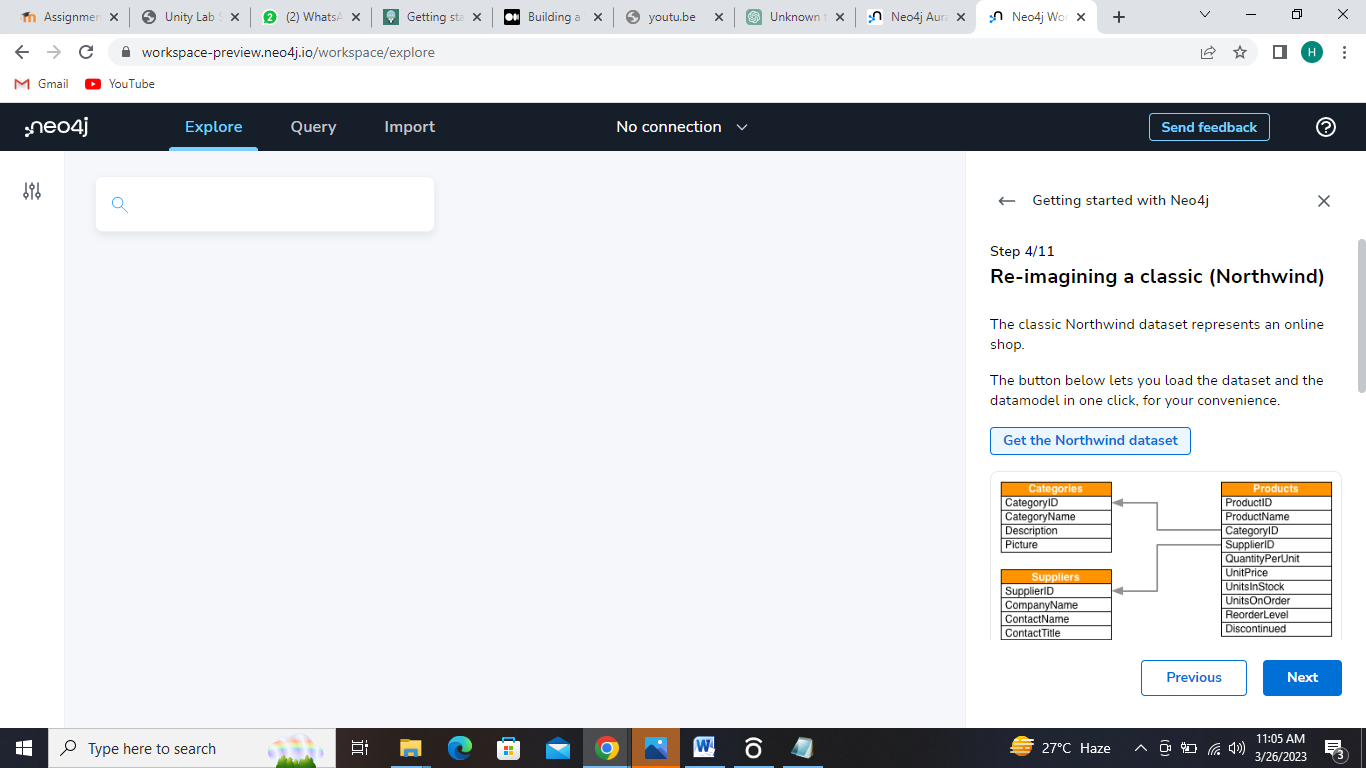
1. Agree with the terms and condition after reading it.



1. Paste the password from the credential file and connect.



1. As a beginner they provide a default dataset to work with it. You can also Drag and drop your csv file ,or create your own.



1. Then click on Run Import
2. You have three panels.
   * + - Explore: to view your graph
       - Query: To run your Cypher query
       - Import: To import your file or create the database with nodes and relationship.

**Cypher language queries:**

1. Match all nodes and relationships in the graph:

MATCH (n)-[r]->(m)

RETURN n, r, this query uses the **MATCH** clause to match all nodes labeled "n" that have any outgoing relationship "r" to any node labeled "m". It then returns all three elements in the match.

Cypher is a declarative query language used for querying graph databases such as Neo4j Aura DB. Here are some examples of Cypher queries:

1. Match all nodes and relationships in the graph:

MATCH (n)-[r]->(m)

RETURN n, r, m

This query uses the **MATCH** clause to match all nodes labeled "n" that have any outgoing relationship "r" to any node labeled "m". It then returns all three elements in the match.

1. Create a new node:

CREATE (:Person {name: 'John', age: 30})

This query creates a new node labeled "Person" with the properties "name" and "age" set to 'John' and 30, respectively.

1. Delete a node:

MATCH (n:Person {name: 'John', age: 30})

DELETE n

This query matches the node labeled "Person" with the properties "name" and "age" set to 'John' and 30, respectively. It then deletes the matched node.

1. Find all nodes with a certain label and property value:

MATCH (n:Person)

WHERE n.age > 25

RETURN n.nameThis query matches all nodes labeled "Person" that have an "age" property greater than 25. It then returns the "name" property of each matched node.

1. Create a relationship between nodes:

MATCH (n:Person {name: 'John'}), (m:Person {name: 'Jane'})

CREATE (n)-[:FRIENDS]->(m)

This query matches the node labeled "Person" with the property "name" set to 'John' and the node labeled "Person" with the property "name" set to 'Jane'. It then creates a new "FRIENDS" relationship between these two nodes.

1. Update a node property:

MATCH (n:Person {name: 'John'})

SET n.age = 31

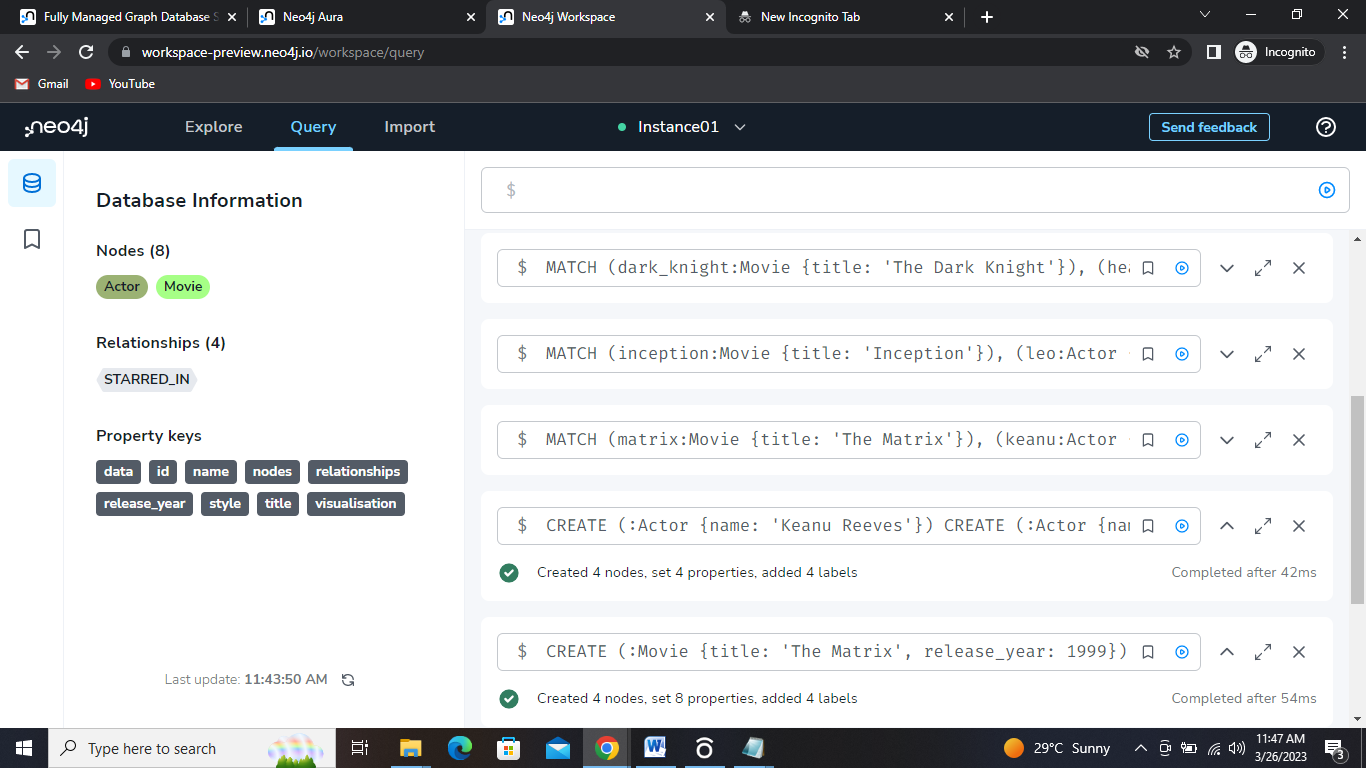
This query matches the node labeled "Person" with the property "name" set to 'John'. It then sets the value of the "age" property to 31.

These are just a few examples of the types of queries you can run in Cypher. Cypher is a very powerful and expressive language, and can be used to query and manipulate even very large and complex graph databases.

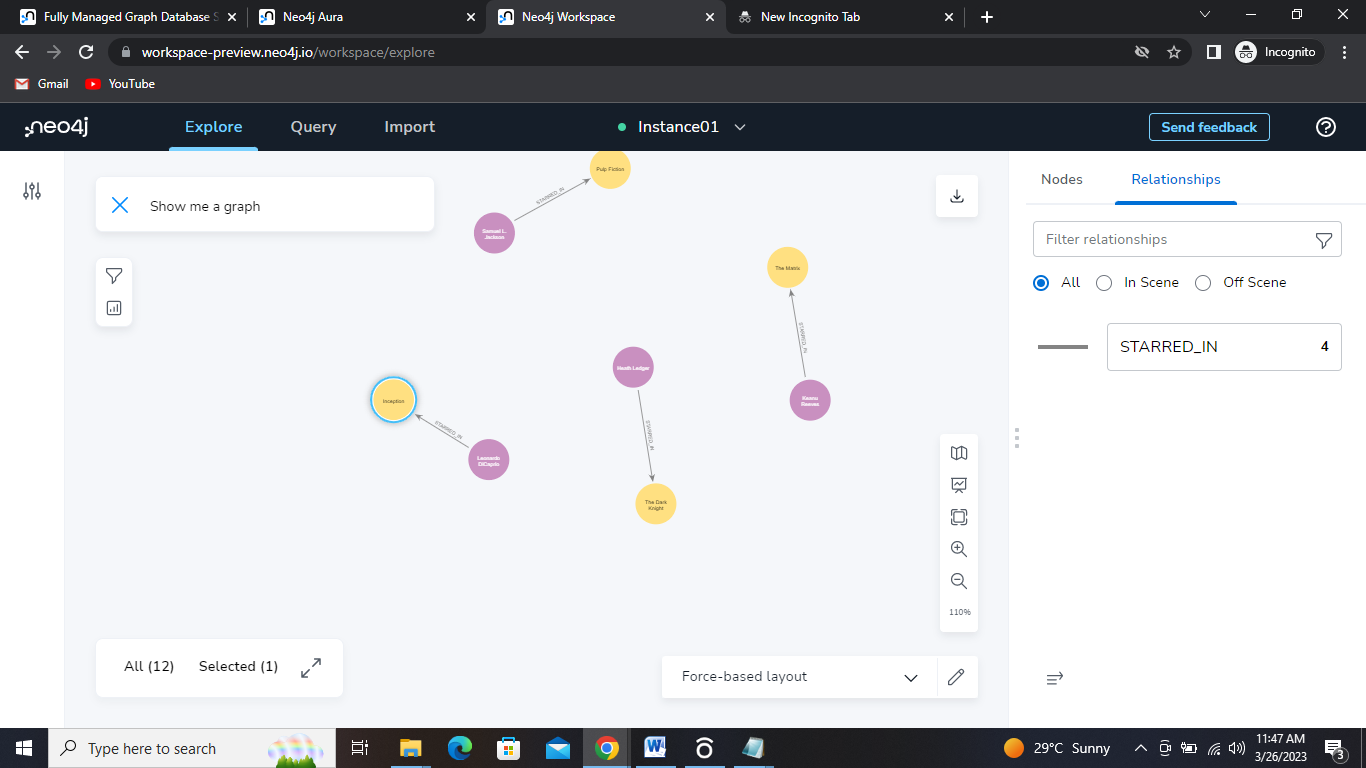
Top of Form

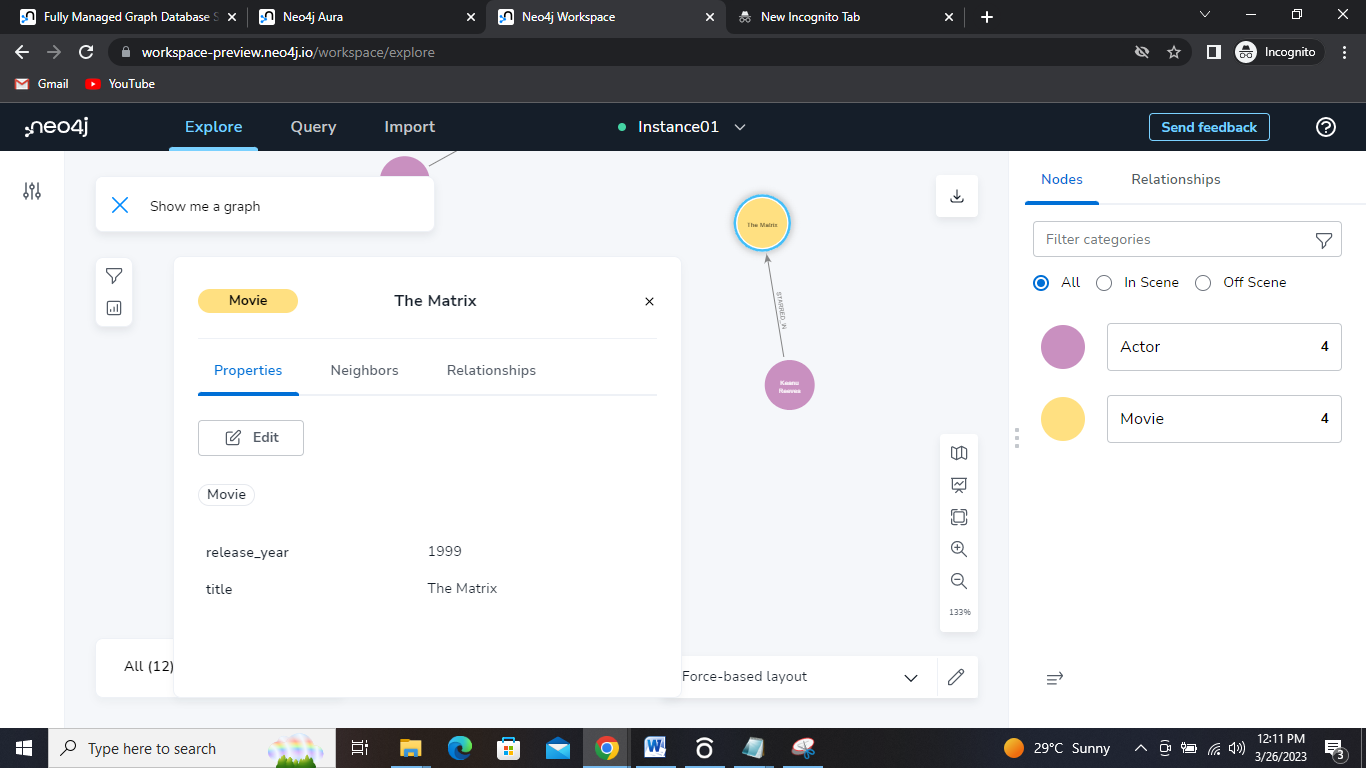
**For understanding import csv file and run panel queries**

1. Import CSV file of movie db and then run the queries in the query panel.



1. Visualize it in Explore panel.





You can also view the properties, neighbors etc. of the nodes

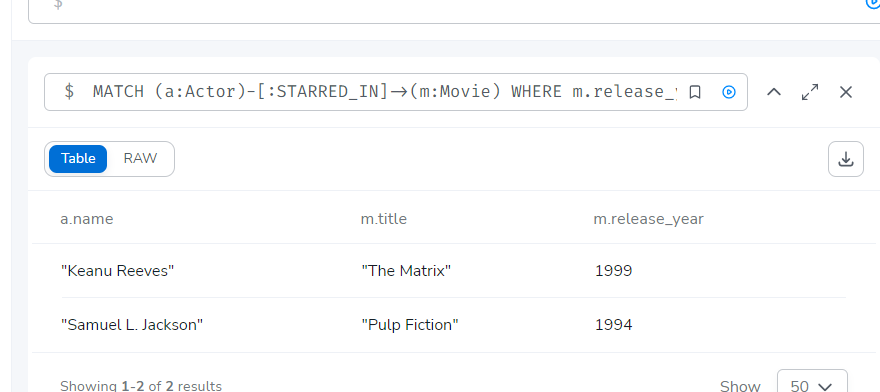
**Some Queries:**

* **For searching**

MATCH (a:Actor)-[:STARRED\_IN]->(m:Movie)

WHERE m.release\_year >= 1990 AND m.release\_year <= 1999

RETURN a.name, m.title, m.release\_year



* **For delete nodes**

**MATCH (m:Movie)-[r:STARRED\_IN]->(a:Actor)**

**DELETE m, r**

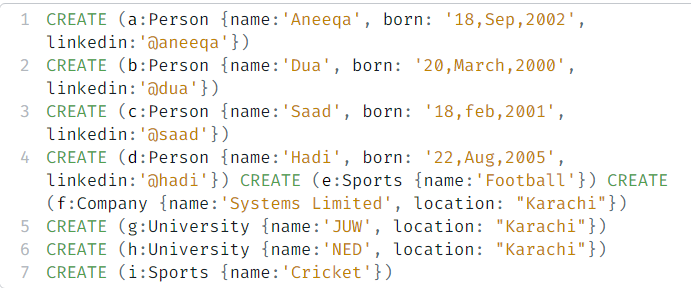
* **For deletion of particular node**

**MATCH (m:Movie {title: 'The Matrix'})-[r:STARRED\_IN]->(a:Actor {name: 'Keanu Reeves'})**

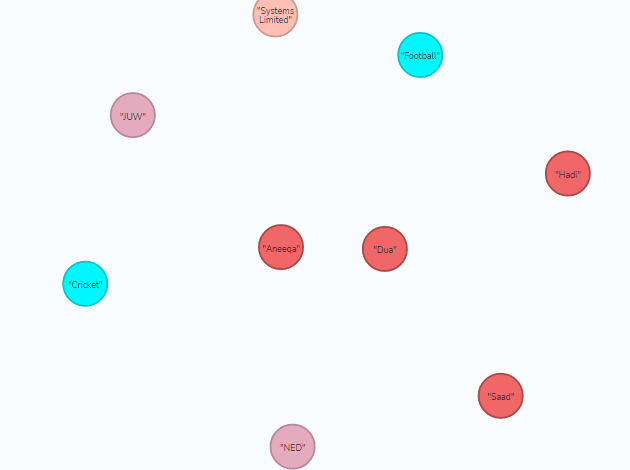
**DELETE m, r**

**Now create your own graph database using Aura db**

Start creating nodes with CREATE clause of CQL

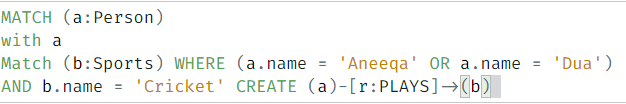


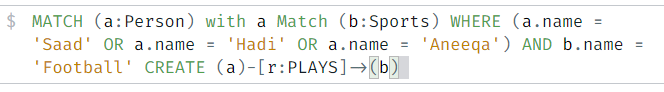
After running above query following nodes are generated:

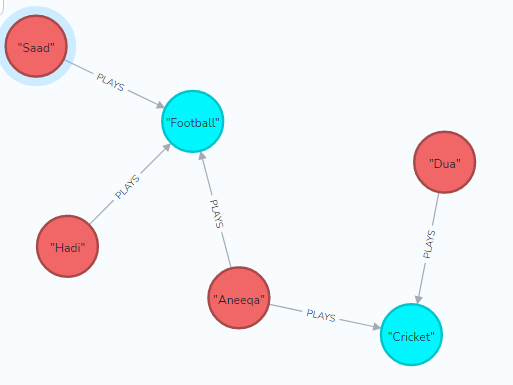


Now create relationships among Nodes

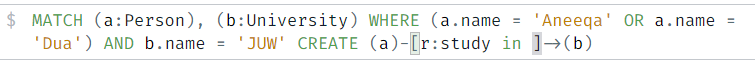
Creating play relationship between person and sport nodes

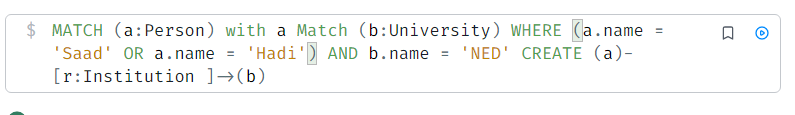


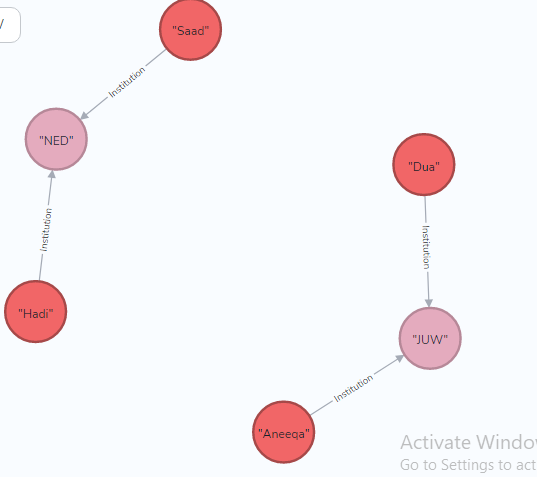


Create a relationship between institution and person nodes 

Now creating relationships between person, comapny and university nodes:







To find all relationships among nodes run the following query

