

✓ **Selvia Nasser Bekhit**

23011293

Step 1: Create a project in Neo4j Aura app and create database

The screenshot shows the Neo4j Aura web interface. The top navigation bar includes the Neo4j logo, 'New Organization / New project', a 'Feedback' button, and user profile icons. The left sidebar lists 'Data services' (Instances, Import, Data APIs) and 'Tools' (Query, Explore). The main content area displays a 'Free instance' with a status of 'RUNNING'. It shows the instance ID 'e493b4b1', type 'AuraDB Free', and metrics for nodes and relationships. A 'Connect' dropdown, a pause button, and an 'Upgrade' button are also visible.

Step 2: Get connection info (URI, USER, PASSWORD) from the connection status

✓
4s

```
[155] !pip install neo4j
```



Collecting neo4j

Downloading neo4j-5.28.1-py3-none-any.whl.metadata (5.9 kB)

Requirement already satisfied: pytz in /usr/local/lib/python3.11/dist-packages (from neo4j)

Downloading neo4j-5.28.1-py3-none-any.whl (312 kB)

312.3/312.3 kB 3.8 MB/s eta 0:00:00

Installing collected packages: neo4j

Successfully installed neo4j-5.28.1

✓
0s

```
[162] from neo4j import GraphDatabase
import networkx as nx
import pandas as pd
```

Step 3: Connect to the instance

```
✓ 1s [165] NEO4J_URI = "neo4j+s://e493b4b1.databases.neo4j.io"
      NEO4J_USER = "neo4j"
      NEO4J_PASS = "sxsqg8eIZ1AejBjSRkS4a5CaP1LG1R4N5K62jkhq_CY"

      driver = GraphDatabase.driver(NEO4J_URI, auth=(NEO4J_USER, NEO4J_PASS))

      def test_connection():
          with driver.session() as session:
              greeting = session.run("RETURN 'Connected to Neo4j!' AS message")
              for record in greeting:
                  print(record["message"])

      test_connection()
```

➡ Connected to Neo4j!

Step 4: Store the graph in Neo4j (already created the graph in the assignment in same notebook)

```
✓ 0s [173] def store_graph_in_neo4j(driver, G):
      with driver.session() as session:
          for node in G.nodes():
              session.run("MERGE (c:Course {name: $course_name})", course_name=node)

          for u, v in G.edges():
              session.run("""
                  MATCH (a:Course {name: $course_name1}), (b:Course {name: $course_name2})
                  MERGE (a)-[:CONFLICTS_WITH]->(b)
              """, course_name1=u, course_name2=v)
```

```
✓ 12m [174] store_graph_in_neo4j(driver, G)
```

```
[91] def build_graph(data):
      G = nx.Graph()

      all_courses = data["Course_Code"].unique()
      G.add_nodes_from(all_courses)

      student_groups = data.groupby("Student_ID")["Course_Name"].apply(list)

      for courses in tqdm(student_groups):
          if len(courses) > 1:
              for course_pair in combinations(sorted(set(courses)), 2):
                  if G.has_edge(*course_pair):
                      G[course_pair[0]][course_pair[1]]["weight"] += 1
                  else:
                      G.add_edge(course_pair[0], course_pair[1], weight=1)

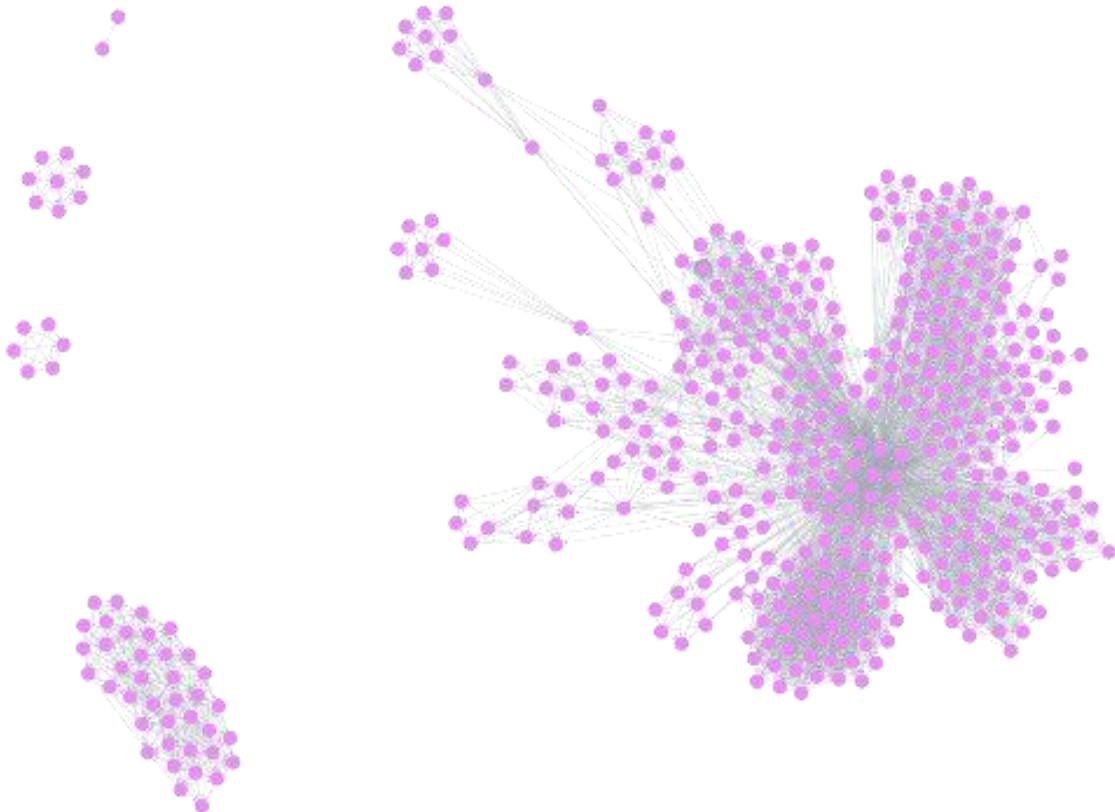
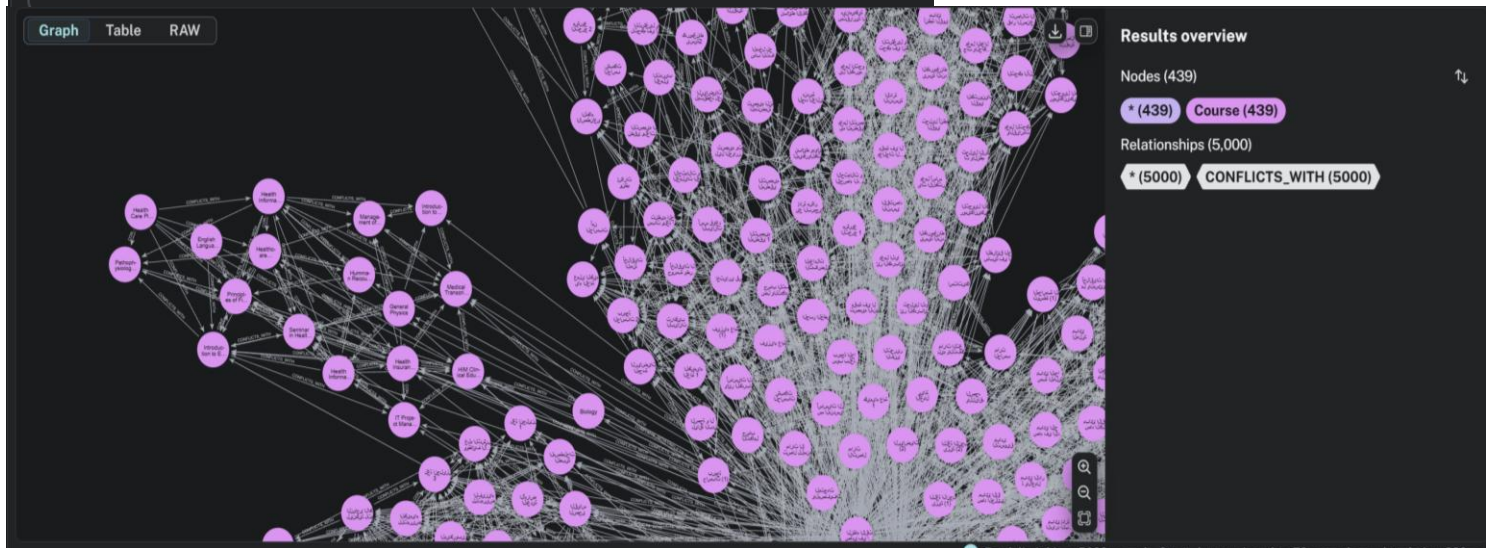
      return G

[92] G = build_graph(data)
      print(f"\nGraph created with {G.number_of_nodes()} nodes and {G.number_of_edges()} edges")

➡ 100% | 1821/1821 [00:00<00:00, 7163.99it/s]
      Graph created with 967 nodes and 5267 edges
```

Step 5: in this step we are going to visualize this graph in the server by opening our instance and type queries that visualize it

```
1 MATCH (c:Course)-[r:CONFLICTS_WITH]->(d:Course)
2 RETURN c, r, d
```



```
1 MATCH (c:Course)-[r:CONFLICTS_WITH]->(d:Course)
2 RETURN c, r, d
```

Graph Table RAW

	c	r	d
1	(:Course {name: "المهارات اللغوية"})	[:CONFLICTS_WITH]	(:Course {name: "التحرير العربي"})
2	(:Course {name: "التحرير العربي"})	[:CONFLICTS_WITH]	(:Course {name: "Biology"})
3	(:Course {name: "المهارات اللغوية"})	[:CONFLICTS_WITH]	(:Course {name: "النظام الإقتصادي في الإسلام"})
4	(:Course {name: "التحرير العربي"})	[:CONFLICTS_WITH]	(:Course {name: "النظام الإقتصادي في الإسلام"})
5	(:Course {name: "المهارات اللغوية"})	[:CONFLICTS_WITH]	(:Course {name: "General Physics"})
6	(:Course {name: "التحرير العربي"})	[:CONFLICTS_WITH]	(:Course {name: "General Physics"})
7	(:Course {name: "المهارات اللغوية"})	[:CONFLICTS_WITH]	(:Course {name: "Health Information Management II"})

Step 6: Closing the driver

✓
0s



```
driver.close()
```

