bonus for study case 1

In [135…
```
import pandas as pd
import networkx as nx
from py2neo import Graph as NeoGraph, Node, Relationship
```

In [137…
```
# ---------- إعداد الاتصال بـ Neo4j ----------
graphdb = NeoGraph("bolt://localhost:7687", auth=("neo4j", "10001000"))
```

In [139…
```
# ---------- حذف البيانات القديمة ----------
graphdb.run("MATCH (n) DETACH DELETE n")
```

Out[139… (No data)

In [141…
```
# ---------- تحميل ملف الإكسل ----------
file_path = "C:\\Users\\Hello\\output_file.xlsx"
sheets = pd.read_excel(file_path, sheet_name=None)  # None = كل الشيتات
```

In [142…
```
# ---------- بناء وتخزين الجراف في Neo4j ----------
for sheet_name, df in sheets.items():
    print(f"Processing sheet: {sheet_name}")

    students_courses = {}
    for _, row in df.iterrows():
        student_id = row.iloc[1]
        course = row.iloc[2]
        students_courses.setdefault(student_id, set()).add(course)

    # بناء الجراف
    G = nx.Graph()
    for courses in students_courses.values():
        for course1 in courses:
            for course2 in courses:
                if course1 != course2:
                    node1 = f"{sheet_name}:{course1}"
                    node2 = f"{sheet_name}:{course2}"
                    G.add_edge(node1, node2)

    # رفع النودز إلى Neo4j
    for node in G.nodes():
        course_name = node.split(":")[1]
        graphdb.merge(
            Node("Course", name=node, course=course_name, sheet=sheet_name),
            "Course", "name"
        )

    # رفع العلاقات إلى Neo4j
    for edge in G.edges():
        node1, node2 = edge
        course1 = graphdb.nodes.match("Course", name=node1).first()
        course2 = graphdb.nodes.match("Course", name=node2).first()
        if course1 and course2:
```

```
            rel = Relationship(course1, "CONFLICTS_WITH", course2)
            graphdb.create(rel)
```

```
Processing sheet: علوم طبية _جدول_الامتحانات
Processing sheet: صيدلة وطب اسنان_جدول_الامتحانات
Processing sheet: العلوم الادارية_جدول_الامتحانات
Processing sheet: الهندسة_جدول_الامتحانات
Processing sheet: حاسب الي_جدول_الامتحانات
```

In [144…

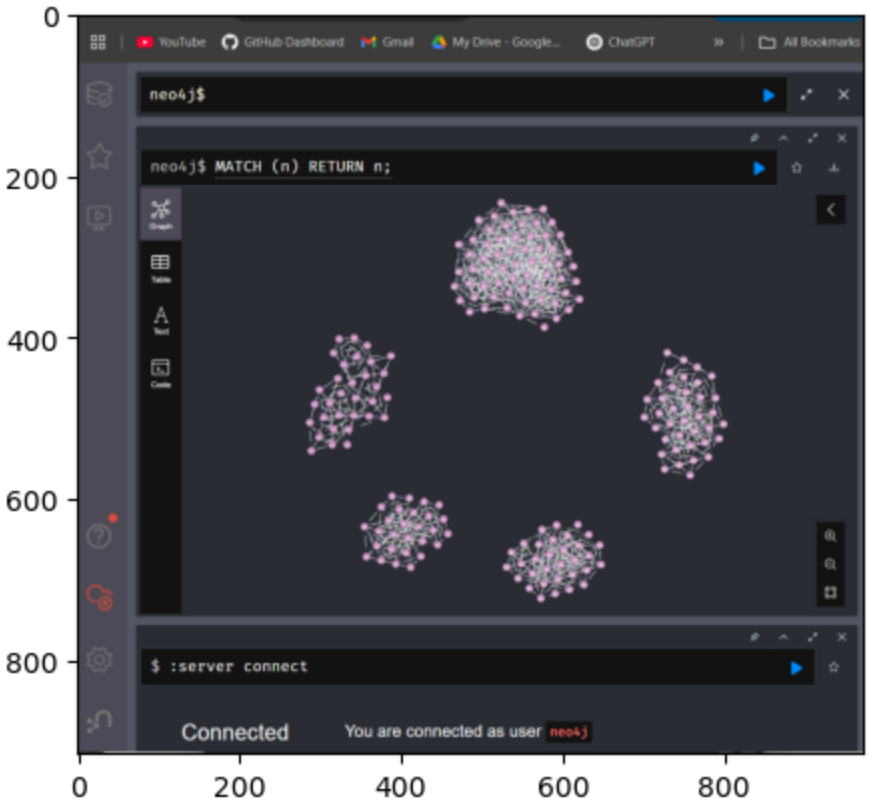
```
print("Intermediate graph successfully stored in Neo4j (before coloring).")
```

```
Intermediate graph successfully stored in Neo4j (before coloring).
```

In [145…

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

# تحميل الصورة وعرضها
img = mpimg.imread("C:\\Users\\Hello\\OneDrive\\Pictures\\Screenshots\\studyB1.png")
imgplot = plt.imshow(img)
plt.show()
```



How to Access Graphs in Neo4j:

1-Open your browser and go to: http://localhost:7474/browser/

2-Enter the connection details:

```
Server: bolt://localhost:7687
Username: neo4j
Password: 10001000
```

3-Click Connect.

4-Once connected, run the following query to view all graphs:

```
MATCH (n) RETURN n;
```

In [ ]:

In [ ]:

This code was written to make dealing with Excel tables easier and cleaner, and to prevent duplication of cycles or relationships that may appear multiple times across different tables. But from my point of view, I do not think it is correct because it requires separating each sheet (college) on its own and having its own courses.

In [129…
```
# ---------- حذف البيانات القديمة ----------
graphdb.run("MATCH (n) DETACH DELETE n")
```

Out[129… (No data)

In [131…
```
def upload_graph(df):
    students_courses = {}

    for _, row in df.iterrows():
        student_id = row.iloc[1]
        course = row.iloc[2]
        students_courses.setdefault(student_id, set()).add(course)

    G = nx.Graph()

    for courses in students_courses.values():
        for course1 in courses:
            for course2 in courses:
                if course1 != course2:
                    G.add_edge(course1, course2)

    # تخزين العقد الفعلية
    node_objects = {}

    for node in G.nodes():
        course_node = Node("Course", name=node)
        graphdb.merge(course_node, "Course", "name")
        node_objects[node] = course_node

    for edge in G.edges():
        node1, node2 = edge
        rel = Relationship(node_objects[node1], "CONFLICTS_WITH", node_objects[node
        graphdb.create(rel)
# ---------- التعامل مع كل شيت ----------
if isinstance(sheets, dict):
    for name, df in sheets.items():
        print(f"Processing sheet: {name}")
```

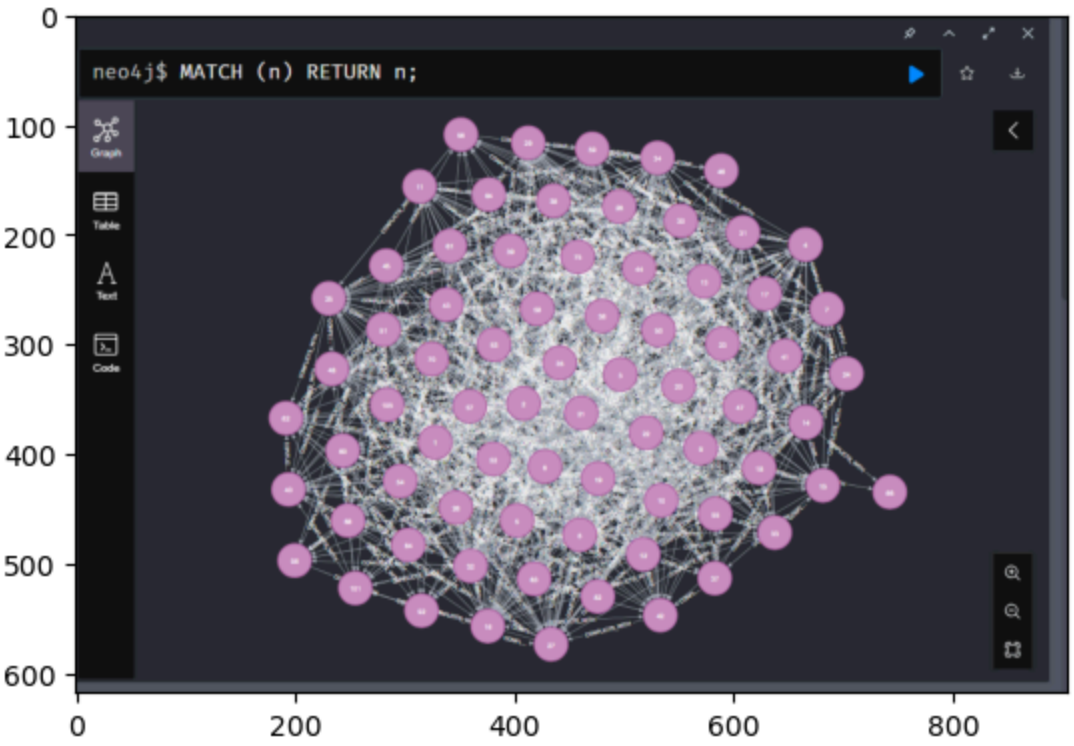
```
        upload_graph(df)
else:
    print("Processing single sheet")
    upload_graph(sheets)

print("✅ Graph uploaded successfully to Neo4j.")
```

```
Processing sheet: علوم طبية _جدول_الامتحانات
Processing sheet: صيدلة وطب اسنان_جدول_الامتحانات
Processing sheet: العلوم الادارية_جدول_الامتحانات
Processing sheet: الهندسة_جدول_الامتحانات
Processing sheet: حاسب الي_جدول_الامتحانات
✅ Graph uploaded successfully to Neo4j.
```

In [133…

```
# تحميل الصورة وعرضها
img = mpimg.imread("C:\\Users\\Hello\\OneDrive\\Pictures\\Screenshots\\studyB12.png
imgplot = plt.imshow(img)
plt.show()
```



to show the graph follow Previous steps

In [ ]: