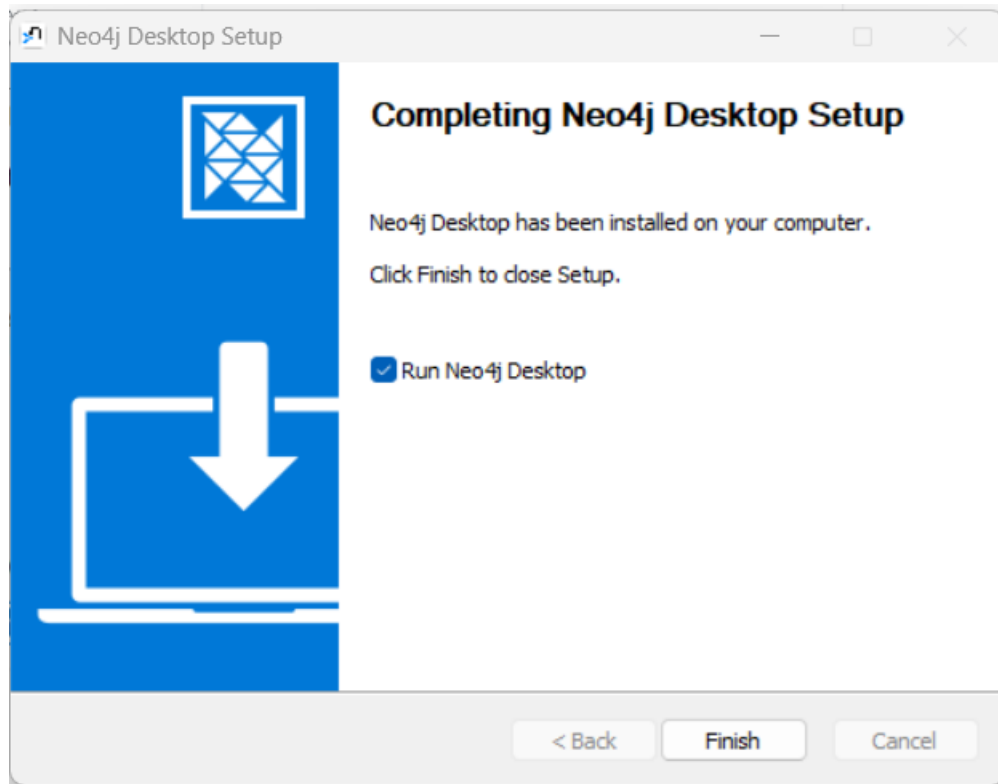


Consider the “**Research Papers Database**” scenario as follows :

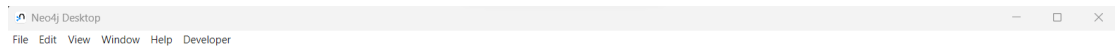
The research papers have authors (often more than one). Most papers have a classification (what the paper is about). The classifications form a hierarchy in several levels (for example, the classification “Databases” has the subclassifications “Relational” and “Object-Oriented”). A paper usually has a list of references, which are other papers. These are called citations.



NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11



Please choose path where you want to store application data

C:\Users\SAGAR RAHANGDALE\Neo4j\Desktop

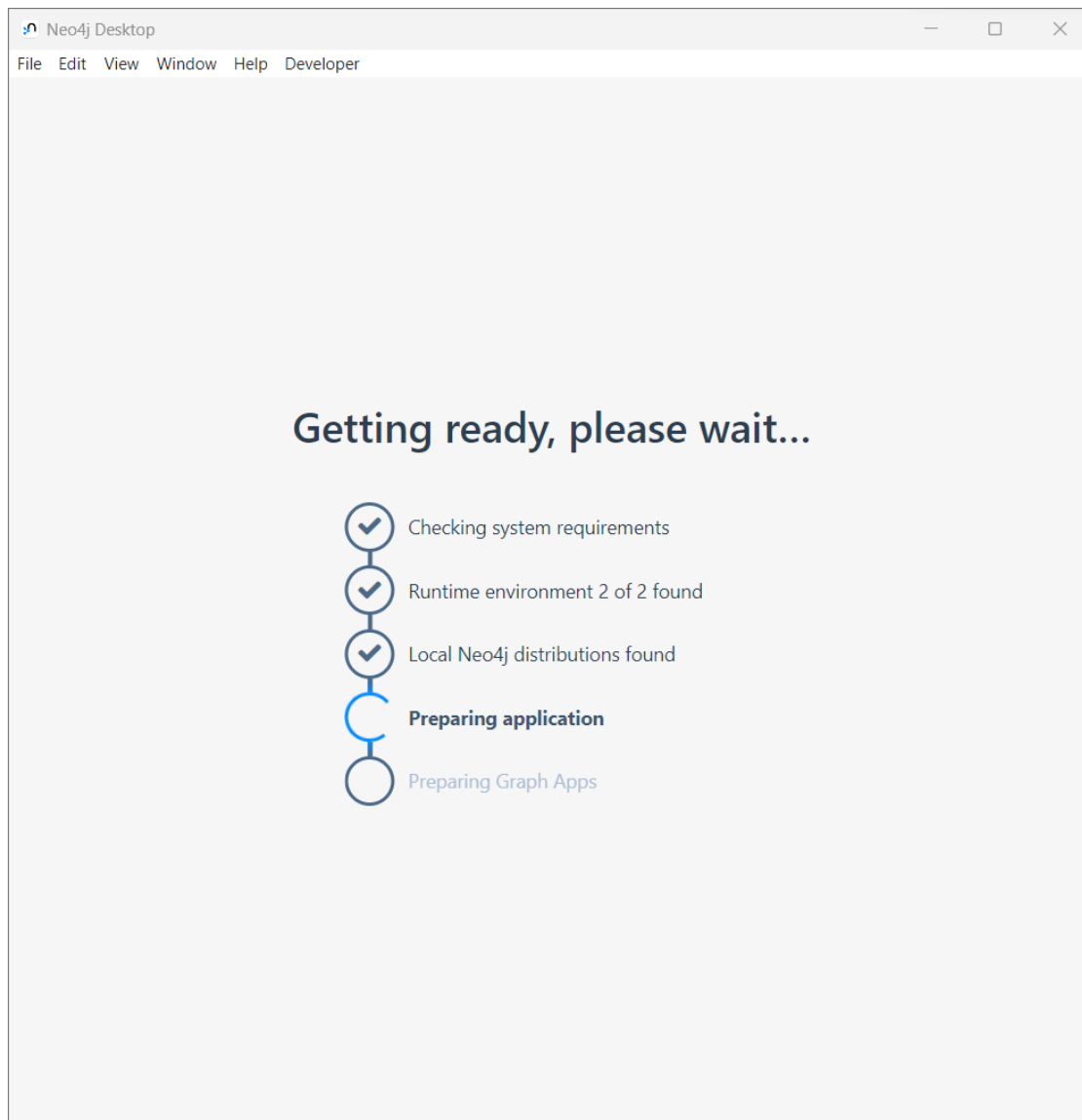
Choose

Confirm

NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11

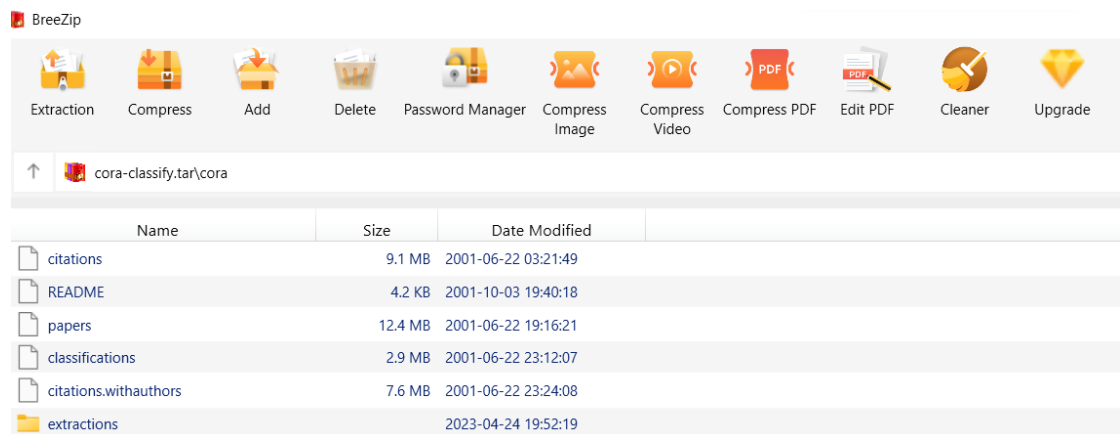


1. Design/model the graph database using Neo4j for above scenario.
2. Download the raw data from **Cora Research Paper Classification Project** : <http://people.cs.umass.edu/~mccallum/data.html> The database contains approximately 25,000 authors, 37,000 papers and 220,000 relationships.

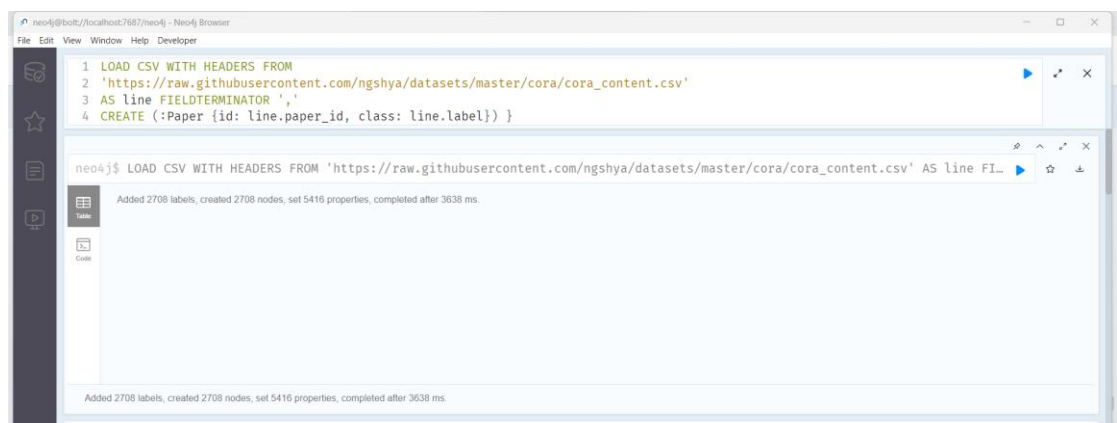
NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11



3. Load this data using Neo4j Data Browser

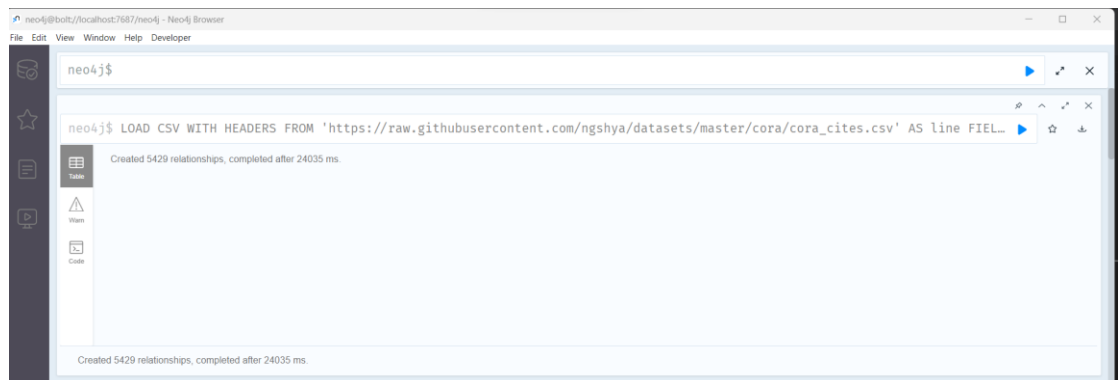


LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/ngshya/datasets/master/cora/cora_content.csv'
,
AS line FIELDTERMINATOR ','
CREATE (:Paper {id: line.paper_id, class: line.label})

NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11



```
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/ngshya/datasets/master/cora/cora_cites.csv'
AS line FIELDTERMINATOR ','
MATCH (citing_paper:Paper {id: line.citing_paper_id}),(cited_paper:Paper {id:
line.cited_paper_id})
CREATE (citing_paper)-[:CITES]->(cited_paper)
```



```
MATCH (p:Paper) RETURN DISTINCT p.class ORDER BY p.class
```

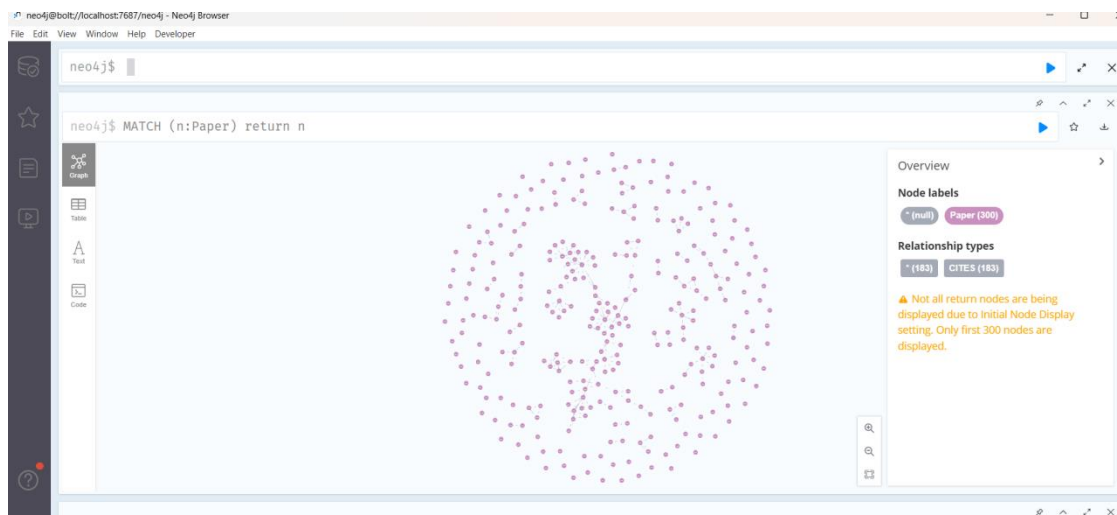
NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11



MATCH p=()-[r:CITES]->() RETURN p

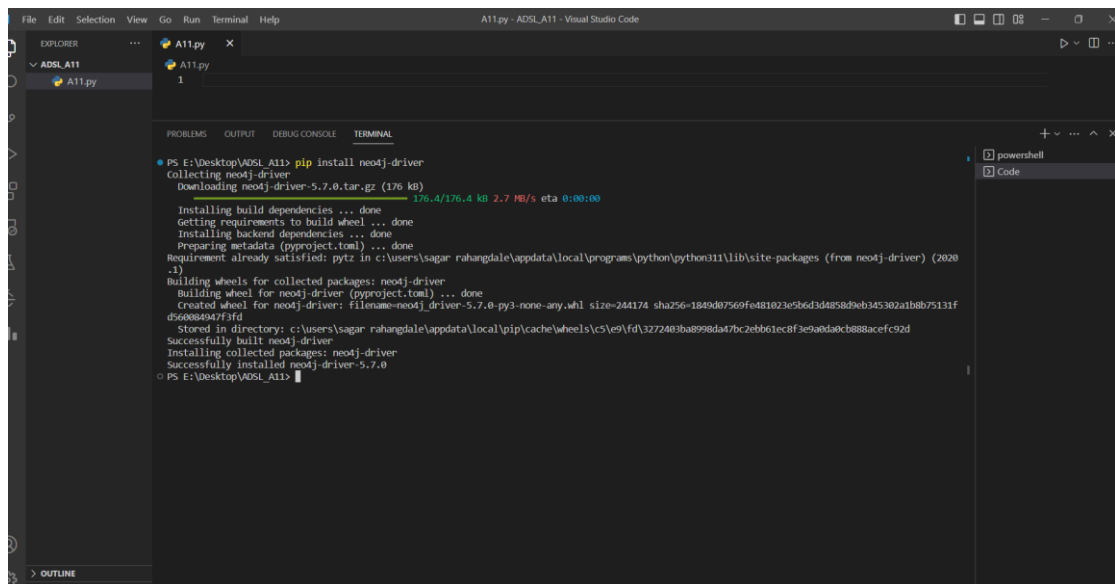


4. Design the python based desktop application for any kind of search on above database. The application should be able to answer queries like

NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11



```
File Edit Selection View Go Run Terminal Help
A11.py - ADSL_A11 - Visual Studio Code

EXPLORER
  ADSL_A11
    A11.py

TERMINAL
  PS E:\Desktop\ADSL_A11> pip install neo4j-driver
  Collecting neo4j-driver
  Downloading neo4j-driver-5.7.0.tar.gz (176 kB)
    176.4/176.4 kB 2.7 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
  Requirement already satisfied: pytz in c:\users\sagar rahangdale\appdata\local\programs\python\python311\lib\site-packages (from neo4j-driver) (2020.7)
  Building wheels for collected packages: neo4j-driver
  Building wheel for neo4j-driver (pyproject.toml) ... done
  Created wheel for neo4j-driver: filename=neo4j_driver-5.7.0-py3-none-any.whl size=244174 sha256=1849d07569fd481023e5b6d3d4858d9eb345302a1b8b75131fd506080a73fd
  Stored in directory: c:\users\sagar rahangdale\appdata\local\pip\cache\wheels\c5\ea\fd\3272403ba8998da7bc2ebb61ec8f3e9ada0cb888acefc92d
  Successfully built neo4j-driver
  Installing collected packages: neo4j-driver
  Successfully installed neo4j-driver-5.7.0
  PS E:\Desktop\ADSL_A11>
```

a) Does paper A cite paper B? If not directly, does paper A cite a paper which in its turn cites paper B? And so on, in several levels.

```
5. import sys
6. import os
7. import tkinter as tk
8. from tkinter import *
9. import tkinter.messagebox
10.
11. # For Neo4j Connection
12. from neo4j import GraphDatabase
13.
14. class Neo4jConnection:
15.
16.     def __init__(self, uri, user, pwd):
17.         self.__uri = uri
18.         self.__user = user
19.         self.__pwd = pwd
20.         self.__driver = None
21.         try:
22.             self.__driver = GraphDatabase.driver(
23.                 self.__uri, auth=(self.__user, self.__pwd))
24.         except Exception as e:
25.             print("Failed to create the driver:", e)
26.
27.     def close(self):
28.         if self.__driver is not None:
29.             self.__driver.close()
30.
31.     def query(self, query, db=None):
```

```
32.         assert self.__driver is not None, "Driver not
initialized!"
33.         session = None
34.         response = None
35.         try:
36.             session = self.__driver.session(
37.                 database=db) if db is not None else
self.__driver.session()
38.             response = list(session.run(query))
39.         except Exception as e:
40.             print("Query failed:", e)
41.         finally:
42.             if session is not None:
43.                 session.close()
44.         return response
45.
46. conn = Neo4jConnection(uri="bolt://localhost:7687", user="neo4j",
pwd="neo4j")
47. # ^ Neo4j Connected
48.
49. window = tk.Tk()
50. window.title("Desktop App by Sagar")
51. window.geometry("700x500")
52. window.configure(bg="grey")
53. blog = tk.StringVar()
54. blog_title = tk.StringVar()
55. direct_id1 = tk.StringVar()
56. direct_id2 = tk.StringVar()
57. recur_id1 = tk.StringVar()
58. recur_id2 = tk.StringVar()
59.
60. # submitting query
61.
62. def submit():
63.     query_string = blog_title.get()
64.     result = conn.query(query_string, db='neo4j')
65.     print(result)
66.     blog.set("")
67.
68. def direct_check():
69.     id1 = direct_id1.get()
70.     id2 = direct_id2.get()
71.     query_string = '''MATCH p=(Paper{id:'''+id1 + \
72.         '''}-[r:CITES]->(Paper{id:'''+id2+'''})) RETURN p'''
73.     result = conn.query(query_string, db='neo4j')
```



```
74.     if (result):
75.         Label(window, text="YES", fg="blue", font=(
76.             "Arial", 15), width=37).grid(row=160)
77.     else:
78.         Label(window, text="NO", fg="RED", font=(
79.             "Arial", 15), width=37).grid(row=160)
80.     blog.set("")
81.
82. def indirect_check():
83.     id1 = recur_id1.get()
84.     id2 = recur_id2.get()
85.     query_string = '''MATCH p=(Paper{id:'''+id1 + \
86.         '''})-[r:CITES]->() MATCH q=(Paper{id:'''+id2+'''})
RETURN q'''
87.     result = conn.query(query_string, db='neo4j')
88.     if (result):
89.         Label(window, text="YES", fg="blue", font=(
90.             "Arial", 15), width=37).grid(row=220)
91.     else:
92.         Label(window, text="NO", fg="RED", font=(
93.             "Arial", 15), width=37).grid(row=220)
94.     blog.set("")
95.
96. # tkinter window
97. Label(window, text="Neo4j Application", fg="black",
98.     font=("Arial", 25, 'bold'), width=37).grid(row=0, column=0)
99. name_label = tk.Label(window, text='Query', font=(
100.     'calibre', 10, 'bold')).grid(row=70)
101.     name_entry = tk.Entry(window, textvariable=blog_title,
102.     font=(
103.         'calibre', 10, 'normal'), width=70).grid(row=80)
104.     sub_btn = tk.Button(window, text='Run Query',
105.     command=submit).grid(row=110)
106.
107.     name_label = tk.Label(window, text='Does Paper with id1
108.     cites id2 directly?', font=(
109.         'calibre', 10, 'bold')).grid(row=120)
110.     name_entry = tk.Entry(window, textvariable=direct_id1,
111.     font=('calibre', 10,
112.     'normal')).grid(row=130)
113.     name_entry = tk.Entry(window, textvariable=direct_id2,
114.     font=('calibre', 10,
115.     'normal')).grid(row=140)
116.     sub_btn = tk.Button(window, text='Check',
117.     command=direct_check).grid(row=150)
```

NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11

```
112.
113.     name_label = tk.Label(window, text='Does Paper with id1
114.     cites id2 indirectly?', font=(
115.         'calibre', 10, 'bold')).grid(row=180)
116.     name_entry = tk.Entry(window, textvariable=recur_id1,
117.                             font=('calibre', 10,
118.                                 'normal')).grid(row=190)
119.     name_entry = tk.Entry(window, textvariable=recur_id2,
120.                             font=('calibre', 10,
121.                                 'normal')).grid(row=200)
122.     sub_btn = tk.Button(window, text='Check',
123.                           command=indirect_check).grid(row=210)
124.
125.     window.mainloop()
```

Desktop App by Sagar

Neo4j Application

Query

MATCH (p:Paper) RETURN DISTINCT p.class ORDER BY p.class

Run Query

Does Paper with id1 cites id2 directly?

40

29000

Check

NO

Does Paper with id1 cites id2 indirectly?

40

29000

Check

NO

- a) Show the full classification of a paper (for example, Databases / Relational)

Note : Follow the submission guidelines.

NAME: Tushar Rajendra Patil

PRN:2020BTECS00075

ADS Assignment :11

Deadline : 16/04/2023

Dr. B. F. Momin
Course Coordinator