

Date:04.04.2022

Third Year B. Tech., Sem VI 2021-22

4CS372 : Advanced Database System Lab

Assignment Submission

PRN No: 2019BTECS00064

Full name: Kunal Santosh Kadam

Batch: T2

Assignment: 9

Title of assignment: Neo4j Graph Database

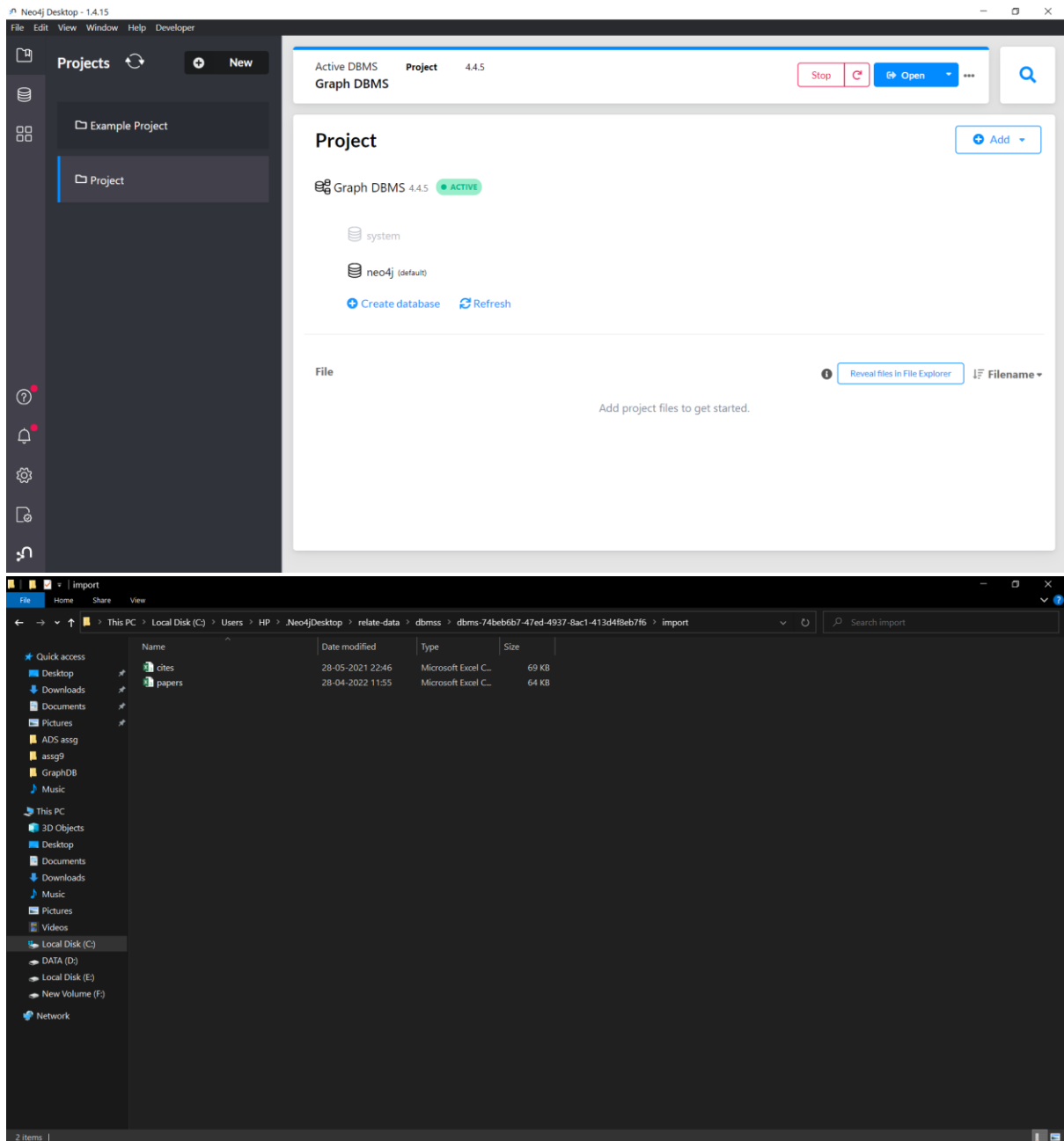
Objective:

Consider the "Research Papers Database", load and query this data using Neo4j Data Browser

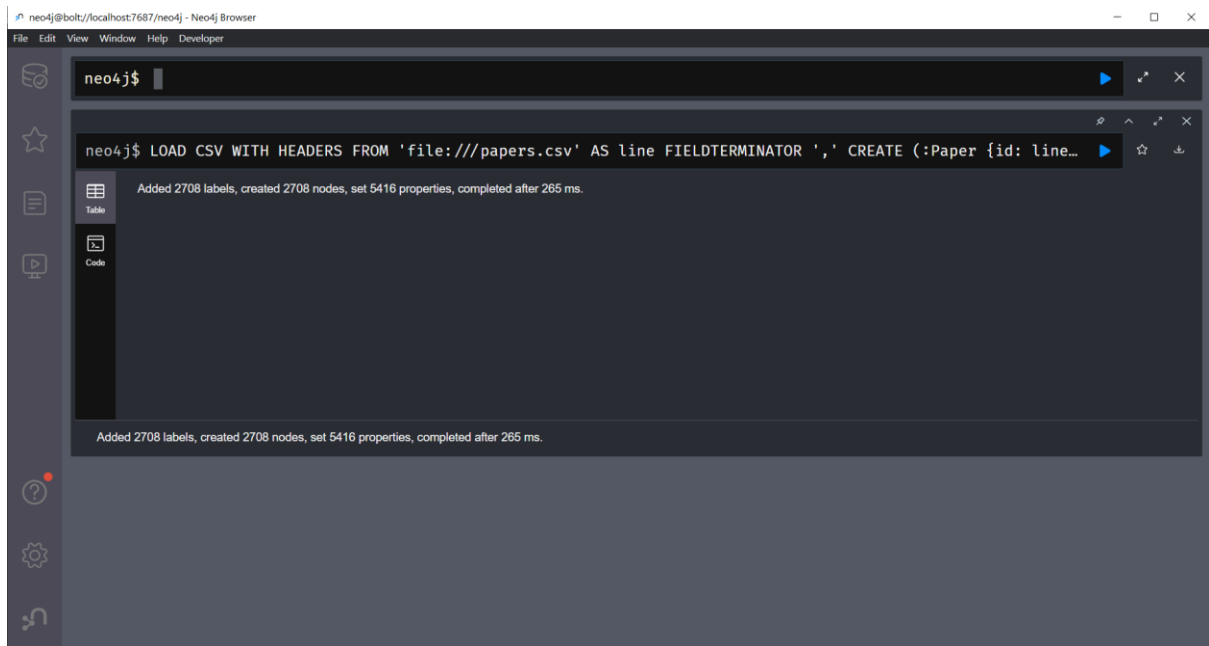
Introduction & Theory:

Neo4j is a graph database management system. Neo4j is a native graph database, built from the ground up to leverage not only data but also data relationships. Neo4j connects data as it's stored, enabling queries never before imagined, at speeds never thought possible.

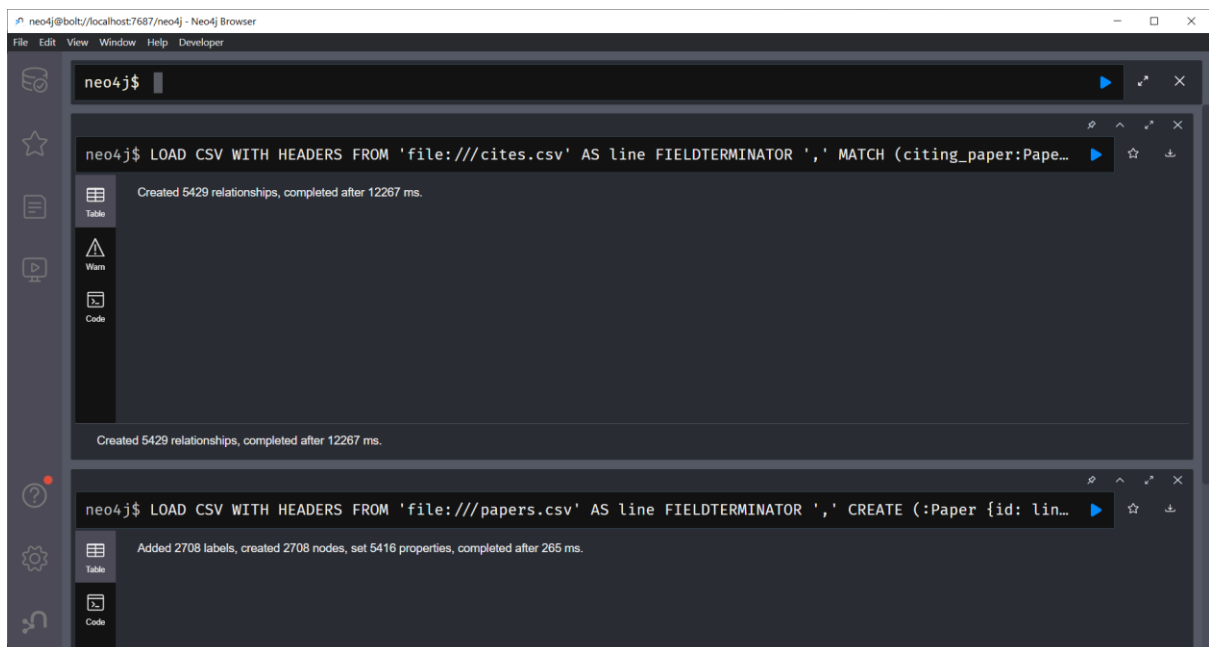
Procedure / Experiment :



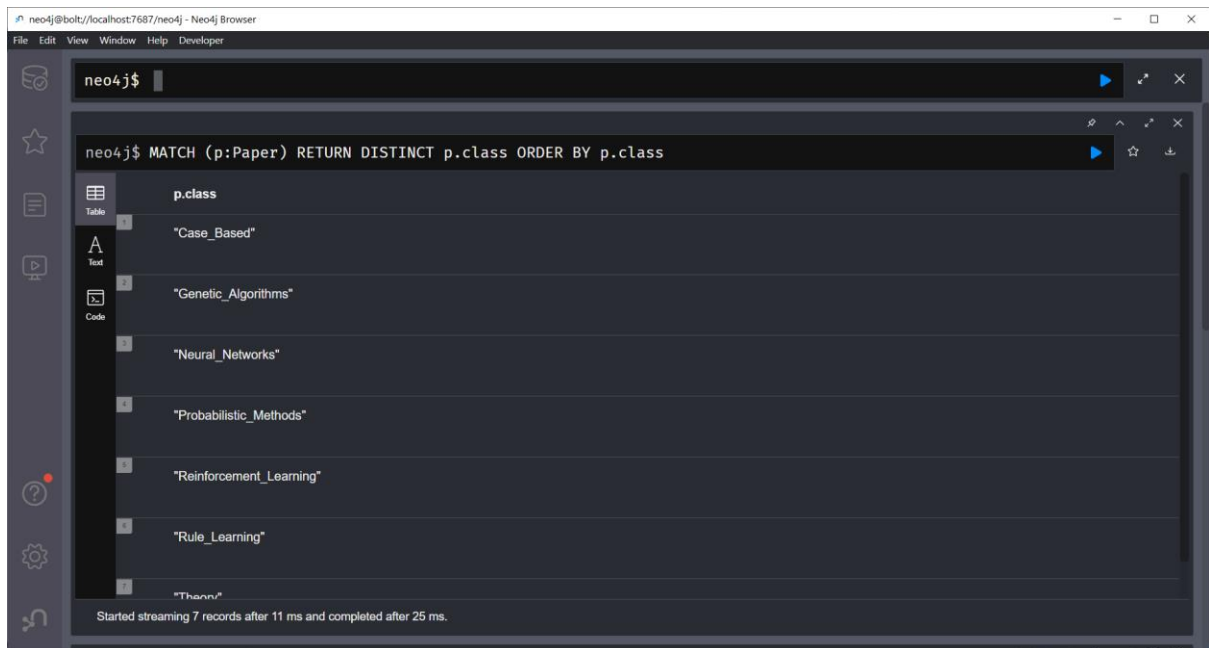
LOAD CSV WITH HEADERS FROM 'file:///papers.csv' AS line FIELDTERMINATOR ',' CREATE (:Paper {id: line.paper_id, class: line.label})



LOAD CSV WITH HEADERS FROM 'file:///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)



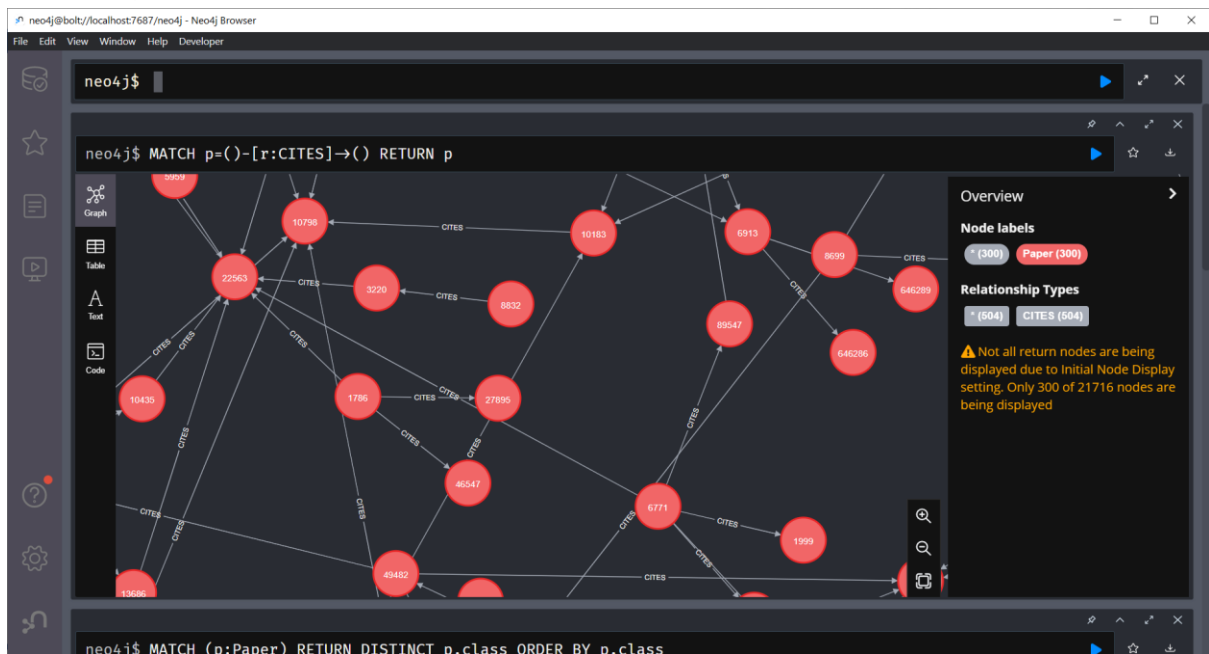
MATCH (p:Paper) RETURN DISTINCT p.class ORDER BY p.class



The screenshot shows the Neo4j Browser interface. The query editor at the top contains the Cypher query: `neo4j$ MATCH (p:Paper) RETURN DISTINCT p.class ORDER BY p.class`. Below the query, the results are displayed in a table view. The table has a single column labeled `p.class` and contains seven distinct values: `"Case_Based"`, `"Genetic_Algorithms"`, `"Neural_Networks"`, `"Probabilistic_Methods"`, `"Reinforcement_Learning"`, `"Rule_Learning"`, and `"Thannu"`. A status message at the bottom indicates: "Started streaming 7 records after 11 ms and completed after 25 ms."

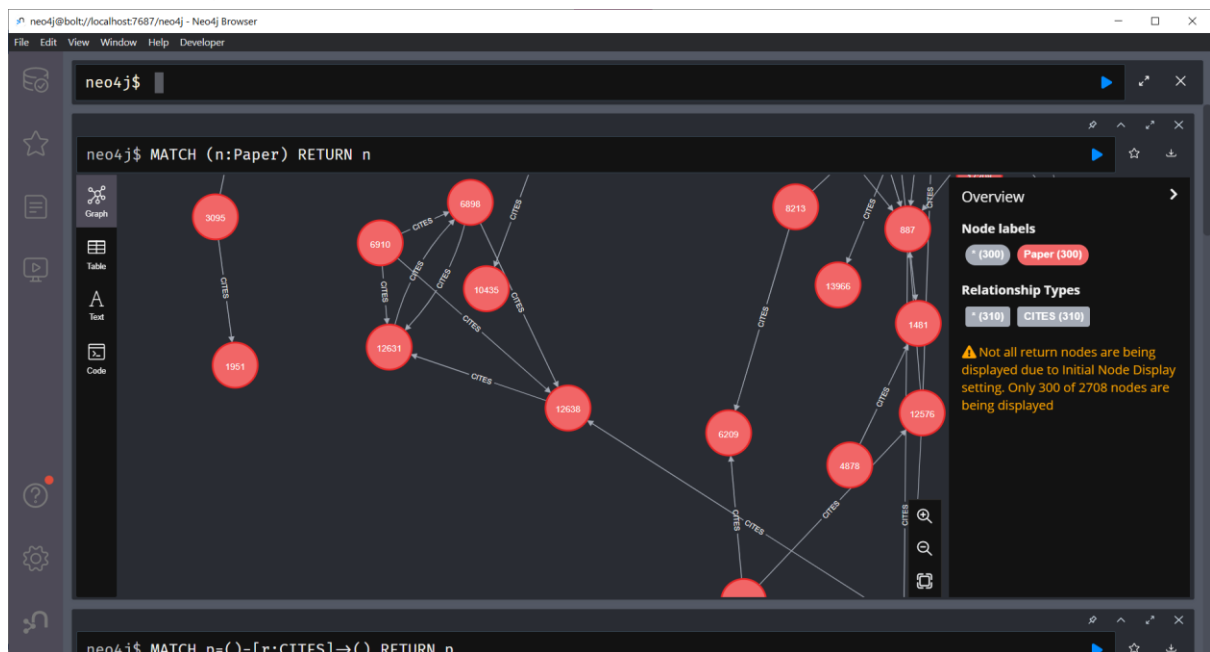
p.class
"Case_Based"
"Genetic_Algorithms"
"Neural_Networks"
"Probabilistic_Methods"
"Reinforcement_Learning"
"Rule_Learning"
"Thannu"

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



The screenshot shows the Neo4j Browser interface with the same query: `neo4j$ MATCH p=()-[r:CITES]->() RETURN p`. The results are displayed as a graph visualization, showing a network of nodes (red circles) connected by edges (gray lines) labeled `CITES`. The nodes are labeled with IDs such as 22563, 10736, 10183, 6913, 5699, 646269, 646266, 83547, 27895, 86547, 6771, 1999, 49482, 13686, 10430, 1786, and 3220. On the right side, an "Overview" panel provides summary statistics: "Node labels" shows 300 nodes for `Paper`, and "Relationship Types" shows 504 relationships for `CITES`. A warning message states: "⚠ Not all return nodes are being displayed due to Initial Node Display setting. Only 300 of 21716 nodes are being displayed".

MATCH (n:Paper) RETURN n

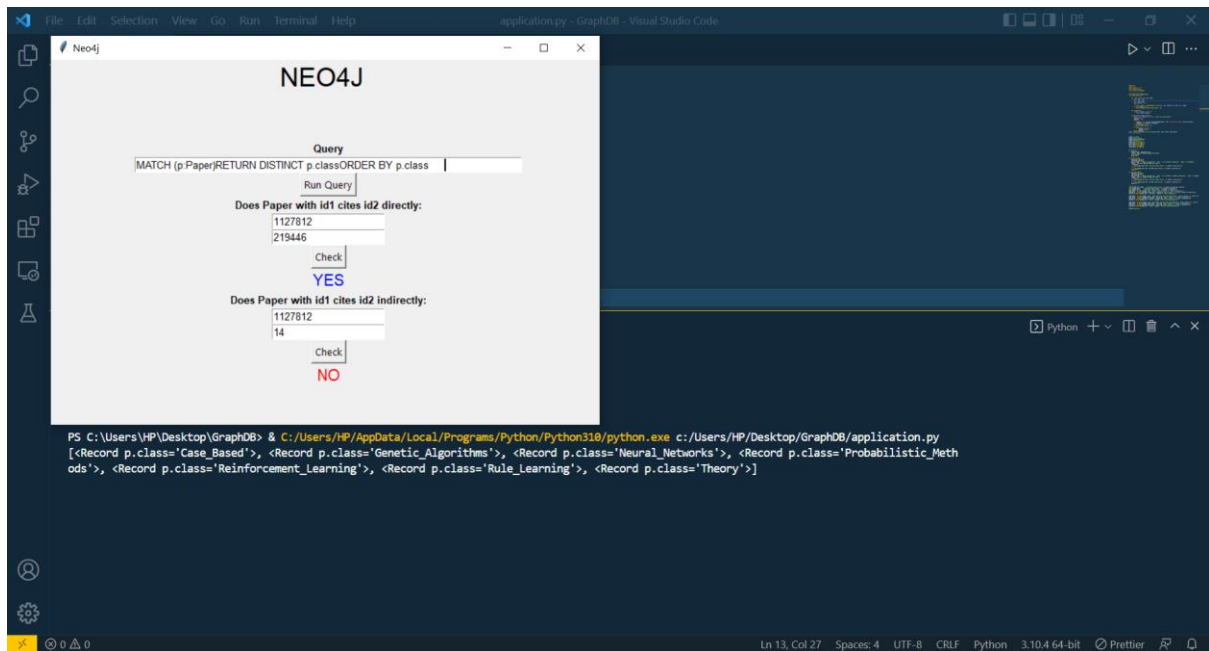


Application

application.py

```
File Edit Selection View Go Run Terminal Help application.py - GraphDB - Visual Studio Code
application.py X
application.py > ...
1 import sys
2 import os
3 import tkinter as tk
4 from tkinter import *
5 import tkinter.messagebox
6
7 # For Neo4j Connection
8 from neo4j import GraphDatabase
9 class Neo4jConnection:
10
11     def __init__(self, uri, user, pwd):
12         self.__uri = uri
13         self.__user = user
14         self.__pwd = pwd
15         self.__driver = None
16         try:
17             self.__driver = GraphDatabase.driver(self.__uri, auth=(self.__user, self.__pwd))
18         except Exception as e:
19             print("Failed to create the driver:", e)
20
21     def close(self):
22         if self.__driver is not None:
23             self.__driver.close()
24
25     def query(self, query, db=None):
26         assert self.__driver is not None, "Driver not initialized!"
27         session = None
28         response = None
29         try:
30             session = self.__driver.session(database=db) if db is not None else self.__driver.session()
31             response = list(session.run(query))
32         except Exception as e:
```

GUI



Conclusion:

Able to load data from “Research Papers Database” and to perform queries on this data. Able to create python application connected with neo4j database.

References:

<https://neo4j.com/>