UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

School of Information Technology
Department of Computer Science

COS326 Database Systems:
Practical 10 2022

Release Date: 22 October 2022
Submission Date: 31 October 2022
Lecturer: Mr S.M Makura

Total: 50 Marks

Objectives

1. Get exposure to the Neo4j graph DBMS.
2. Learn how to create and use a graph for a Neo4j database.
3. Appreciate the differences between SQL and NoSQL databases.

You are expected to have completed the tutorial on Neo4j before you start on this practical exercise. When you are done:

1. You must submit the following files:
   a. ***Task1Queries.txt*** with all the CREATE and MATCH queries for Task 1
   b. ***Task2Queries.txt*** with all the MATCH queries for Task 2
   c. ***Task3Queries.txt*** with all the queries for Task 3
   d. All the necessary files for your ***Java application*** for Task 4
   e. Compress the above documents into an archive and upload it to ClickUP **before** the due date/time. The file name for the archive must have your student number as part of the file name, e.g. **uxxxxxxxx-prac10.zip** or **uxxxxxxxx-prac10.tar.gz** where uxxxxxxxx is your student number.
2. Book for a demo session via Discord to demo the practical.

NO LATE submissions will be accepted after the submission date and time has lapsed. Do not wait till the last minute to submit and start giving excuses that you faced technical challenges when you tried to submit.

Task 1: CREATE AND QUERY A GRAPH DATABASE [16 marks]

1. Using Neo4J, create a database called ***Prac10Neo4j.graphdb.***

2. Study the data description of a Twitter social network in the table below:

Entity	**property: value**	**relationships**	**with entity whose label**	**relationship property:val**
Person	**name: @Thandi** from: Durban	FOLLOWS	**Person,** Johan	since: 2014
			Person, Neo	since: 2011
		TWEETED	**Tweet,** #Trees	date: 20-08-
Person	**name: @Johan**	FOLLOWS	**Person,** Thandi	since: 2012
		RETWEETED	**Tweet,** #Trees	date: 20-08-
Person	**name: @Neo** from: Tshwane age:	FOLLOWS	**Person,** Thandi	since: 2016
			Person, Melanie	since: 2016
		TWEETED	**Tweet,** #Rhinos	date: 05-09-
Person	**name: @Melanie** from: Joburg age: 21	FOLLOWS	**Person,** Johan	since: 2012
			Person, Thandi	since: 2012
		FAVORITED	**Tweet,** #Rhinos	date: 05-09-
Tweet	**hashtag: #Trees** message:	As above		
Tweet	**hashtag: #Rhinos** message:	As above		

3. Using Neo4j, create and test the queries (write and read clauses) given below and then store them in the file ***Task1Queries.txt.***

Queries:

a. Write Cypher statements to create the graph as follows**: (9 marks)**

   i. Create all the Person nodes and all the FOLLOWS relationships (4)

   ii. Show the current contents (nodes and relationships) of the database. (1)

   iii. Create all the Tweet nodes, TWEETED, RETWEETED and FAVORITED relationships (3)

   iv. Show the current contents (nodes and relationships) of the database. (1)

b. Write Cypher queries to do the following: **(7 marks)**

i. List all the node labels with no duplicates in the list

ii. List names of the people in the graph in alphabetical order.

iii. List all the #tags for the tweets in the graph in alphabetical order

iv. List all the relationship types with no duplicates

v. List names of the people in the graph who have tweeted & their tweets

vi. List all the #tags and messages for the tweets which have been retweeted

vii. List all the #tags and messages for the tweets which have been favorited.

Task 2: AGGREGATION AND PATH QUERIES [15 marks]

Create and test the queries for the tasks specified below and then store them in the file ***Task2Queries.txt***

1. Path and other queries (8 marks)

a. Find the tweets that are 1 or 2 links away from the Person Neo. (2)

b. Show the nodes in the path from Melanie to Neo (2)

c. For each person who has tweeted, provide a report to indicate (true/false) whether their tweet has been retweeted. (2)

d. Find the shortest (FOLLOWS) path from Melanie to Neo. (2)

2. Aggregation queries: (7 marks)

a. Count the number of nodes in the network (1)

b. For each person, count the number of persons they follow. (2)

c. For each person, count the number of persons who have been following them since 2014. (2)

d. Show the name and number of followers for the person who has the largest number of followers. (2)

Task 3: IMPORTING DATA INTO A Neo4j DATABASE [8 marks]

Create a new database and import data into this database:

1. Download the files ***persons.csv*** and ***follows.csv*** from ClickUP.

2. Create a second database called ***Prac10Neo4jB.graphdb and*** *then click the 'Start' button.*

3. Under the database folder ***Pra107Neo4jB.graphdb create*** a sub-folder called ***import***. Copy the files ***persons.csv*** and ***follows.csv*** to the ***import*** folder.

4. Write a Cypher statement to import the ***persons.csv*** file data into the database. (3)

5. Write a Cypher statement to show the current database contents. (1)

6. Write a Cypher statement which uses the ***follows.csv*** file data to create the FOLLOWS relationships for the person nodes that have been created in the database. (3)

7. Write a Cypher statement to show the current database contents. (1)

Task 4: JAVA APPLICATION TO ACCESS Neo4j DATABASE [11 marks]

Create a Java application using NetBeans or any Java IDE you are comfortable with that accesses the ***Prac10Neo4jB.graphdb*** database. The application should:

1. Report that it has successfully connected to the database. (1)
2. Make use of transactions. (2)
3. Provide a menu for the actions that a user can perform (2)
4. Enable the user to perform the following actions on the database:

    a. count the number of person nodes in the network (1)

    b. for each person, count the number of persons they follow. Show only the top 100 results (2)

    c. show the names of persons who are in a relationship of the form:

    (person1)-[:FOLLOWS]->(person2)-[:FOLLOWS]->(person3) (3)