CS 343 Graph Data Science
Midterm Exam 1

29 Feb 2024

Academic Integrity Pledge

I, ________________________________, solemnly pledge that I will uphold the principles of academic integrity throughout the completion of this assignment. I understand that academic integrity is essential to the integrity of the learning process and the value of the education I receive. I promise that I will not engage in any form of academic dishonesty, including but not limited to plagiarism, cheating, or unauthorized collaboration.

I acknowledge that any violation of academic integrity undermines the trust and fairness within the educational community and may result in severe consequences, including academic penalties and disciplinary action. I understand that my commitment to academic integrity is not only a personal responsibility but also a reflection of my respect for the intellectual property of others and the integrity of the academic institution.

By signing this pledge, I affirm my dedication to honesty, integrity, and ethical conduct in all academic endeavors.

Signature: ________________________________

Student Name and ID: ________________________________

Date: ________________________________

DO NOT TURN OVER UNTIL INSTRUCTED.

1. (30 points) Question: Imagine you are tasked with designing a comprehensive graph data model to digitally represent the spatial layout of a university campus. This model will serve as the foundation for navigation applications aimed at efficiently guiding students, faculty, and visitors to various campus destinations.

   Your model should account for the diverse array of spaces found on campus, including cafeterias, courtyards, classrooms, labs, and learning spaces. Each of these spaces will be represented as nodes in the graph, with appropriate metadata to distinguish between them.

   In addition to spaces, the model must also capture the pathways connecting these spaces. These pathways will serve as the edges in the graph, facilitating movement from one location to another. Consider the different modes of traversal available on campus, such as walking, taking stairs, or using elevators. Each pathway should be annotated with metadata indicating its accessibility features, including whether it is wheelchair-friendly.

   Furthermore, the model should support efficient route planning between any two locations on campus. Navigation applications will rely on this capability to provide users with optimal paths based on their starting and ending points, considering factors such as distance, accessibility, and any specific user preferences.

   Your task is to design a graph data model that effectively captures the spatial layout of the university campus, enabling seamless navigation for all campus users while accommodating the diverse needs and preferences of the university community.

   Solution: A very basic answer could be as follows:

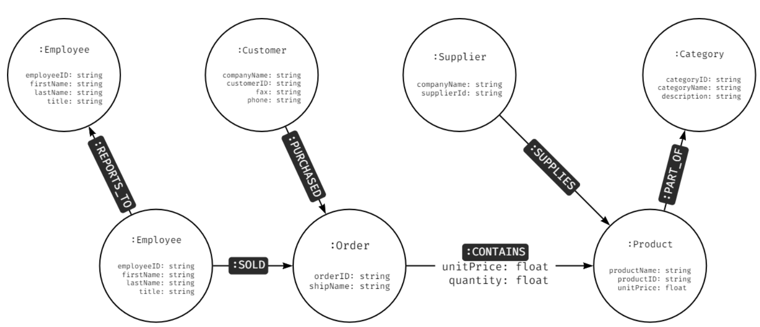
   - Nodes: Space Name, Capacity, openningTime, closingTime
   - Edges/Relation: PATH distance, time

   However, a better answer should provide specialize labels for Spaces and PATH.

   - Nodes: Space Name, Capacity, openingTime, closingTime, Cafeteria type , Courtyard, Classroom facility, Lab type, LearningSpace
   - Relation: WALK distance, time, STAIRS steps, time, ELEVATOR time, RAMP time

2. (25 points) Northwind Database stores the sales data of a company. It has Employees, Customers, Orders, OrderDetails, Products, Categories, and Suppliers. The ERD of the Northwind database is given at the end of the paper for reference purpose.

   We have converted the some part of the ERD into a graph property model. The graph property model is shown below:



   Provide Cypher queries based on this provided graph property model to answer the following questions:

(a) List the names of all the employees who have sold products to the customers and the names of products they have sold.
RETURN: Employees, List of Products

> **Solution:**
> $MATCH(e : Employee) - [: SOLD]- > (o : Order) - [: CONTAINS]- > (p : Product)$
> $RETURNe, collect(p.productName)$

(b) List all those employees who are not managers.
RETURN: Employees

> **Solution:** MATCH (e:Employee) WHERE NOT (e)-[:REPORTS_TO] -> () RETURN e

(c) List the names of all employees along with the number of orders they have handled.
RETURN: Employees, Number of Orders

> **Solution:** MATCH (e:Employee)-[:HANDLED]->(o:Order) RETURN e, count(o) testing

(d) List all managers and number of employees reporting to them.
RETURN: Manager Name, Number of Employees

> **Solution:** MATCH (m:Employee)-[:REPORTS_TO]− >(e:Employee) RETURN m.firstName, count(e)

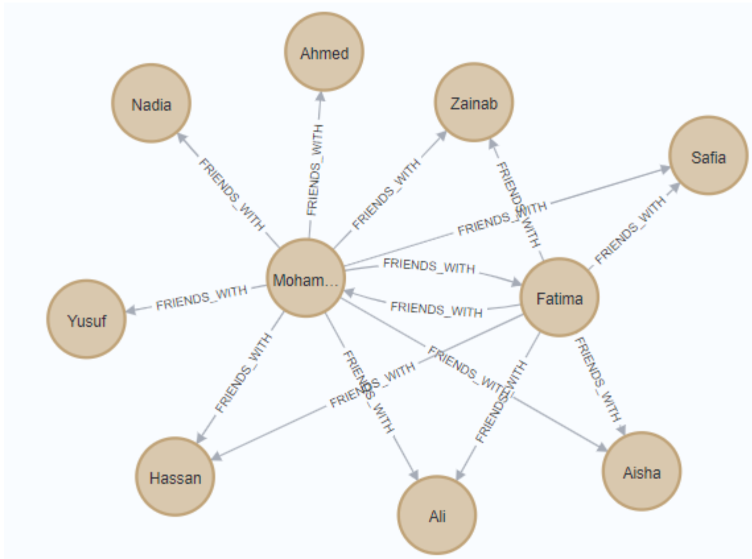
(e) Determine the intermediary managers between the "Vice President, Sales" and an employee named "Robert King".
RETURN: Number of intermediary managers

> **Solution:** MATCH (v:Employeetitle: 'Vice President, Sales')-[:REPORTS_TO*]− >(m:Employee)-[:REPORTS_TO*] − > (r:EmployeefirstName: 'Robert King') RETURN count(m)

3. (25 points) You are given a social network dataset where individuals define who their friends are through friendship relationships. The dataset consists of 10 people and friendships between them have been established to create a graph network.

Provide Cypher queries for the following questions. Consider the graph structure where each node represents a person, and friendship relationships are represented by edges between nodes.

You are also provided a script to generate the graph on Neo4j. The following is the graph after generating this network for reference purpose.



(a) Determine the node with the maximum in-degree.

Solution: match (n:Person)¡-[r]-()
return n, count(distinct r) as inDegree
ORDER BY inDegree desc LIMIT 1

(b) Calculate the average number of friends per person in the network.

Solution:

```
match (n:Person)-[r:FRIENDS_WITH]->(m)
with  n, count(m) as friends
return avg(friends)
```

Partially correct answer: excluding the direction of the relationship

(c) Determine the longest friendship chain in the network.

Solution:

```
match p=()-[:FRIENDS_WITH*]->() return p, length(p)
ORDER BY length(p) desc LIMIT 1
```

(d) Calculate clustering coefficinet for the person “Fatima”.

Solution:

```
match (a:Person{name:"Fatima"})-[:FRIENDS_WITH]-(b:Person)
WITH a, count(DISTINCT b) as friends
MATCH (a)-[:FRIENDS_WITH]-(b:Person)-[common:FRIENDS_WITH]-(c:Person)-[:FRIENDS_WITH]-(a)
with  a, friends, count(distinct common) as common_friends
return (2.0 * common_friends) / (friends * (friends-1)) as clustering_coefficient
```

(e) List person and their friends with list of their mutal friends.
RETURN: Person, List of Friends, List of Mutual Friends

Solution:
```
MATCH (a:Person)-[:FRIENDS_WITH]->(b:Person)
WITH a, b, collect(b.name) as friends
MATCH (a)-[:FRIENDS_WITH]-(c:Person)-[:FRIENDS_WITH]-(b)
WITH a, b, friends, collect(c.name) as mutual_friends
RETURN a.name, friends, mutual_friends
```

(f) Find the friends of "Fatima" and their friends of friends using 2-hop traversal.

Solution:
```
Match (a:Person{name:"Fatima"})-[:FRIENDS_WITH*2]-(b:Person) return *
```

4. (20 points) Multiple Choice Questions

(a) Select a valid WHERE clause to follow this MATCH clause:
MATCH $(a) - [b : CONNECTED_TO] - > (c)$

A. WHERE b:CONNETED_TO.key = 'value'
B. A WHERE clause cannot follow a MATCH clause
C. WHERE b.key = 'value'
D. WHERE :CONNECTED_TO.key = 'value'

Solution: WHERE b.key = 'value'

(b) Labels are best described as:

A. Table names in the graph database.
B. Tags that are used to group nodes into sets.
C. Special types of node or relationship properties.
D. Unique tags on each node for fast lookup.

Solution: Tags that are used to group nodes into sets.

(c) Suppose you have a database that contains a million Person nodes and a million Places nodes. All person nodes have a property called name. The Places nodes do not have a property, name. What is the fastest way to count the number of Person nodes in the database?

A. MATCH (p:Person) RETURN sum(p)
B. MATCH (p) WHERE "Person"IN labels(p) RETURN count(p)
C. MATCH (p) WHERE EXISTS(p.name) RETURN count(p)
D. MATCH (p:Person) RETURN count(p)

Solution: MATCH (p:Person) RETURN count(p)

(d) Suppose that you have created Car nodes in the graph with both labels Vehicle and Car (There can be nodes with Vehicle and other labels). Which Cypher statement(s) can be used to remove the Vehicle label from all Car nodes? Chose all that apply.

A. MATCH (n:Car) REMOVE n:Vehicle

B. MATCH (n:Vehicle) REMOVE n:Vehicle
C. MATCH (n:Vehicle:Car) SET n:Car
D. MATCH (n:Vehicle) label(n) = Car

Solution: MATCH (n:Car) REMOVE n:Vehicle

(e) The PROFILE keyword can be used for what purpose?
A. Identify the schema for the current database.
B. return the query plan and execution information
C. provide current statistics for the database.
D. create parameterized queries.

Solution: return the query plan and execution information

(f) Select the Cypher statements that will delete a node with an id 3563 and all of its possibly connected relationships? Choose all that apply.
A. MATCH (a:Thingid:3563) DELETE a
B. MATCH (a:Thingid:3563) DETACH DELETE a
C. MATCH (a:Thingid:3563)-[r]-(b) DELETE a,r
D. MATCH (a:Thingid:3563)
OPTIONAL MATCH (a)-[r]-() DELETE a,r

Solution: MATCH (a:Thingid:3563) DETACH DELETE a
MATCH (a:Thingid:3563)
OPTIONAL MATCH (a)-[r]-() DELETE a,r

(g) All nodes with the same label must have the same property keys.
A. True
B. False

Solution: False

(h) When you create a relationship in Cypher using the CREATE clause, what happens if the relationship already exists?
A. The relationship is updated with the new properties.
B. The relationship is deleted and recreated.
C. The relationship is not created.
D. The relationship is created with the new properties.

Solution: The relationship is created with the new properties.

(i) What Cypher keyword do we use to add a property to an existing node?
A. ADD
B. SET
C. CREATE
D. UPDATE

Solution: SET

(j) We have a graph that has millions of Person nodes that are related using the :FOLLOWS relationship. Given the following query, // MATCH (p:Person)-[:FOLLOWS*2]-¿(p2:Person)
WHERE p.name = 'Alice'
RETURN p, p2

What node or nodes are returned by this query?

A. Two Person nodes that are followed by Alice.
B. All Person nodes that are exactly two hops away from Alice using the :FOLLOWS relationship.
C. All Person nodes that are one or two hops away from Alice using the :FOLLOWS relationship.
D. All Person nodes that are exactly two hops away from Alice using the :FOLLOWS relationship.

Solution: All Person nodes that are exactly two hops away from Alice using the :FOLLOWS relationship.

(k) What Cypher statement will return actors and the directors of the movies they have acted in?

A. MATCH (a:Actor)-[:ACTED_IN]-¿(m:Movie)¡-[:DIRECTED]-(d:Director) RETURN a, d
B. MATCH (:Actor)-[a:ACTED_IN]-¿(:Movie)¡-[d:DIRECTED]-(:Director) RETURN a, d
C. MATCH (a:Actor)-[a:ACTED_IN]-¿(m:Movie)¡-[d:DIRECTED]-(d:Director) RETURN a, d
D. MATCH (a:Actor)-[:ACTED_IN]-¿(m:Movie)¡-[:DIRECTED]-(d:Director) RETURN a, m, d

Solution: MATCH (a:Actor)-[:ACTED_IN]-¿(m:Movie)¡-[:DIRECTED]-(d:Director) RETURN a, d

(l) What statement best descirbes index-free adjacency?

A. Each group of nodes is stored as a table where the relationship are mapped to nodes in the same group.
B. The ability to use indexes to find nodes.
C. The ability to traverse the graph using indexes.
D. Nodes and their relationships are stores as pointers so that access is very fast without the need for an index to traverse nodes.

Solution: Nodes and their relationships are stored as pointers so that access is very fast without the need for an index to traverse nodes.