Name：Kennedy Lumumba

Due Date：27 April 2025

**(MIT212)**

**Master of Information Technology**

**Advanced Database Design and Development**



**Question 1: Five Applications of Neo4jnin Recommendation System**

1. **E-commerce Product Recommendations**: Neo4j enables real-time recommendation of products by modeling users, products, and their interactions (like purchases or views) as a graph. Algorithms like collaborative filtering or content-based filtering can be implemented using relationships such as BOUGHT or VIEWED.
2. **Movie Recommendation Systems:** In movie platforms like Netflix, Neo4j can model user preferences, watch history, and movie metadata (genres, actors, etc.). The graph structure allows for personalized recommendations using similarity metrics and shortest path algorithms.
3. **Friend Suggestions in Social Media**: Platforms like Facebook or LinkedIn can utilize Neo4j to recommend new connections. Nodes represent users, and relationships like FRIENDS\_WITH or FOLLOWS are used. Algorithms like Common Neighbors or Jaccard Similarity help identify potential friends based on mutual connections.
4. **Job Recommendations in Professional Networks** Neo4j can represent users, companies, job listings, and skill sets. It allows for recommendations of job openings that align with a user’s experience, education, and skill profile.
5. **Online Learning Platforms**: Systems like Coursera or Udemy can use Neo4j to recommend courses. Users, courses, instructors, and interests form the graph. The system can suggest courses based on prior enrollments, ratings, or peer activity.

**Question 2: Component of Two Applications**

**Application 1: E-commerce Recommendation System**

* **Nodes**: User, Product, Category, Brand
* **Relationships**:

BOUGHT (date, quantity)

VIEWED (timestamp, duration)

SIMILAR\_TO (score, categoryMatch)

**Application 2: Social Media Friend Suggestion**

* **Nodes**: User, Interest, Location, Post, Event
* **Relationships**:

FRIENDS\_WITH (since, interactionCount)

FOLLOWS (since, strength)

INTERESTED\_IN (since, level)

LOCATED\_IN (since, visitsPerMonth)

**Question 3: Challenges in Building a Social Media Friend Suggestion System**

1. **Scalability**: As the number of users and relationships grows, query performance can degrade. Efficient indexing and query planning are crucial.
2. **Real-Time Suggestions**: Users expect fast suggestions. Neo4j's in-memory graph structure supports this but needs optimized query execution.
3. **Privacy and Security**: Recommendations must respect user privacy settings, avoiding exposure of private data.
4. **Bias and Fairness**: Algorithms might favor users with large networks. Care must be taken to ensure diverse and equitable suggestions.
5. **Data Quality**: Incomplete or outdated user data can lead to irrelevant suggestions.

**Question 4: Sample Data Model (For Social Media Friend Suggestion**

**Labels**

User: (id, name, age, gender)

Interest: (id, name, category, popularity)

Location: (id, name, region, population)

Post: (id, content, timestamp, likes)

Event: (id, name, date, type)

**Relationships**:

FRIENDS\_WITH (since, interactionCount)

FOLLOWS (since, strength)

INTERESTED\_IN (since, level)

LOCATED\_IN (since, visitsPerMonth)

### Question 5: Neo4j Technologies

**Graph Algorithms**:

gds.nodeSimilarity.stream(): Calculates similarity score between nodes (users) based on shared relationships (e.g., interests or friends)

gds.pageRank.stream(): Measures influence of users in the network.

**Built-in Functions**:

* size(): Measures list length or pattern match count.
* collect(): Collects items into lists for aggregation
* count(): Aggregates node or relationship counts
* exists(): Checks existence of properties or relationships