# Graph Databases Study Points Assignment

Julius Krüger Madsen – cphjm352@cphbusiness.dk

## Provide in writing, each student individually, answers to the following questions:

### 1. What are the advantages and disadvantages of using graph databases and which are the best and worse scenarios for it?

Some of the better scenarios for using a graph database are applications where relationships between data points are critical, such as social networks, recommendation engines, fraud detection, and network management. SQL databases, on the other hand, are better suited for applications where data is highly structured, and relationships between data points are not as important. They are commonly used in finance, e-commerce, and other applications where transactional data is critical. If I was to mention a scenario where it would be better to use SQL databases, it would be a scenario where your application would require the use of ACID compliance.

### 2. How would you code in SQL the Cypher statements you developed for your graph-algorithms-based query, if the same data was stored in a relational database?

If we were to store the same data that we have in a graph database in a relational database, we would need to restructure our data model to fit the relational database paradigm. translating graph-based queries to SQL requires a different approach, as we need to consider the relational schema and how the data is structured in tables. However, the fundamental logic of the queries remains the same, and we can achieve similar results in SQL by using joins, filters, and other SQL operations.

### 3. How does the DBMS you work with organizes the data storage and the execution of the queries?

When a Cypher query is executed, Neo4j uses an optimizer to determine the most efficient way to traverse the graph based on the query's path expressions. Neo4j also uses caching to optimize query performance by storing frequently accessed data in memory. Neo4j's data storage and query execution strategies are optimized for graph-based data structures and allow for efficient traversal and querying of relationships between data points.

### 4. Which methods for scaling and clustering of databases you are familiar with so far?

**Casual Clusters:**

1. **Read/write Replicas:** This involves creating multiple copies of a database on different servers and keeping them in sync using replication mechanisms. This

approach provides increased availability and can improve query performance, as queries can be executed against any replica.

2. **Leader and followers:** The idea here is that you have some core servers and some replica servers. The leader is responsible for coordinating the cluster and accepting all writes. If a leader falls, the followers elect a new leader with majority votes.

## KNN-Similarity-Algorithm

I decided to work with the K-Nearest-Neighbor (KNN) algorithm on our database. If we take a quick look on the algorithm on the Neo4j website, it says that *"… creates new relationships between each node and its k nearest neighbors. The distance is calculated based on node properties."* In our database I decided that games ratings would be the best property of the games node to use for algorithm to calculate the distance between each game. The KNN algorithm can be used to solve many different problems in graph database, such as fraud detection, and network analysis. By finding the K nearest neighbors of a node, we can gain insights into the relationships and patterns in the data. This is all stuff that might not be apparent from a simple graph traversal.