**Project Part 2 (50%)**

* You will need to download PostgreSQL and Neo4J Desktop to do this project. You will need to use Cypher to create commands in Neo4J (along with using the Neo4J graph algorithm libraries)

Imagine you work for a company called ***GlanScanna***. They have a database (create\_movies.sql, movies\_data.sql available in the Project part 2 folder) which support their online movie recommender app where a person can type in a movie and get a recommendation for a similar movie. The recommendation is based on a hypercube which scores movies across 18 genres as discussed in the PostgreSQL lecture.

Now however they want to expand their business by adding social media aspects such as **users**, who can be **friends\_with** other users, and who can **like** certain movies (rating how much they like the movie out of 5).

The company have asked you to explore the issues in extending the existing database with new tables or else moving the existing relational database to a graph database implemented in Neo4J which includes the new social media aspects.

Answer the following questions in your report (show the commands/queries used and show screen dumps for each question below in your report)

1. Create the database in PostgreSQL (2 marks)
2. Using the cube, create a query showing the most similar movie to ‘Alien’ in the database (2 marks)
3. Describe the strategy you will take to convert from a relational model to a graph model. (3 marks)
4. Export the data in the tables to csv files and show any modifications you make in the data (in excel) (2 marks)
5. Import the data for actors, movies and genres into Neo4J (2 marks)
6. Create the relationships between actors and movies (use the file movie\_actors.csv and create the relationships) (2 marks)
7. Create the relationships between movies and genres. Describe how the code works. (10 marks)
8. Add 6 people (Joe,Jack,John,Jill, Jane,Janice), Joe is friends with Jill, Jill is friends with Jane, Joe is friends with Jack, Jack is friends with Janice. Jack likes ‘’movie1” rating it 3, Jack likes ‘”’ rating it 5, Jack likes ‘’” rating it 4, Jill likes ‘’” rating it 4, Jill likes ‘”’ rating it 5, Jill likes ‘”’ rating it 5. (2 marks) (Add your own favourite movies here). Add tables and the same data to the relational database model as well. (2 marks)
9. Build the following queries in Neo4J. For each query show the query and a screen capture of the result.
   1. Show all the movies your own favourite actor acts in (2 marks)
   2. Show all the genres of all the movies your favourite actor acts in. (2 marks)
   3. Show all the movies that Joe’s friends like (2 marks). Compare this to how you might implement this in the relational model. (3 marks). Can query optimization help improve the performance in the relational model? (3 marks)
   4. Show who has most friends (using degree centrality algorithm in gds package) (9 marks). Describe how the code works. Compare this with an implementation in the relational model. (3 marks)
   5. Show the degree of separation between any two actors, showing the movies and actors that are used to reach them (hint: use the shortestPath algorithm in the gds package) (9 marks). Describe how the code works. Discuss if this can be achieved in the relational model. (3 marks)
   6. Show an example of using Bloom to visualise the data about users, their friends and the movies they like. (2 marks)
10. Compare and contrast the Relational Database versus the Graph Database implementation making at least five points supported by your experience from the above exercises. Make a final recommendation on choosing either Relational or Graph to the company based on your experience. (35 marks)

Include any references you use in your report.

Upload your report with your Tnumber to Ouriginal by Friday December 9.