Advanced Databases & noSQL (INFDEV026B)
Assignment 3

Instructions

- The assignment must be implemented using Neo4j.
- Practice the cypher queries and write them into a text file (not word!).

You can do either assignment A or B

Assignment A

Consider the Entity-Relationship diagram in Figure **??** representing the model for an airport database.

- Give a graph database implementation in Neo4j. Fill in the database with data satisfying the following constraints:
  - At least 6 airports, two of which must be name 'Schiphol', and 'Venezia Marco Polo', two must be located in 'London', and one in 'Rome'.
  - The values of `size` in `airports` must be 'Small', 'Medium', or 'Large'.
  - At least one airport must be 'Large'.
  - Each airport must have at least 5 terminals.
  - 'Venezia Marco Polo' must have a terminal 'B'.
  - At least 5 companies, two of which must be 'Lufthansa' and 'KLM'.
  - At least 5 flights, one of which must be scheduled before 15:00.
  - At least 3 gates per terminal.
  - The values for `state` in `gates` must be either 'Boarding' or 'Closed'.
  - There must be a 'Boarding' gate for terminal 'B' in 'Venezia Marco Polo'.
- Implement the following queries in cypher:
  1. Find the name and the capacity of all 'Large' airports.
  2. Find the total capacity of the airports in the same city. Output the name of the city and the total capacity.
  3. Find the name of the airport with the highest capacity. Output the name and the capacity.
  4. Find all the opened terminals in 'Schiphol'. Output the code.
  5. Find all the terminals of the airports in 'London'. Output the code.

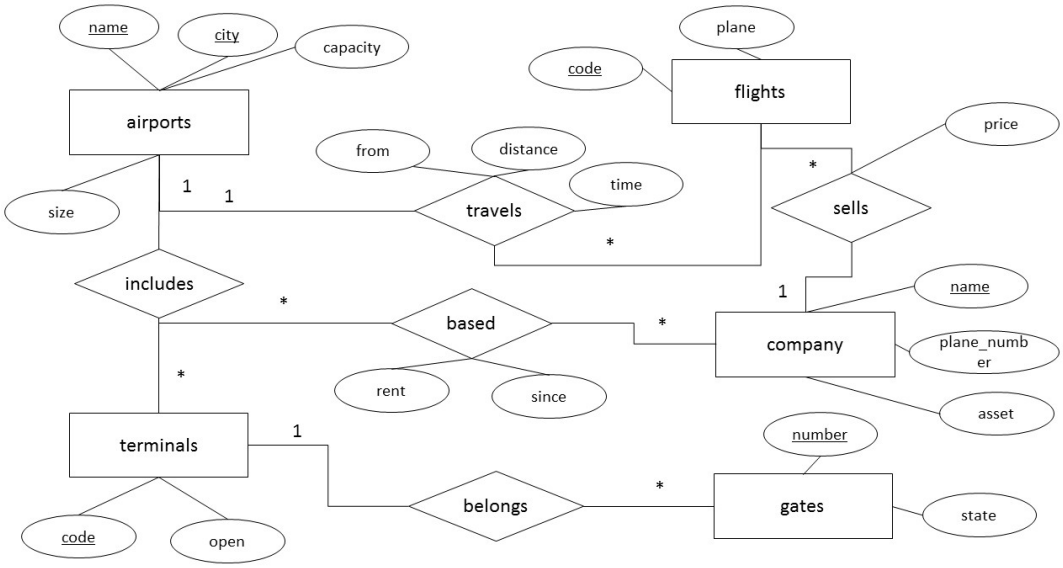

Figure 1: ER Diagram for the assignment

6. Find all the gates that are boarding in 'Venezia Marco Polo' for terminal 'B'. Print the number and the state.
7. Find all the flights landing in 'Rome' for 'Lufthansa' and 'KLM'. Print the code and the plane.
8. For each company find the amount of flights going to 'Rome' leaving before 15:00. Print the company name and the total of flights.

Assignment B

Consider the normalized table structure you have created for assignment 1 about the multinational company application. Your tasks are:

- Re-factor the application in order to use it with the graph database (Neo4j)
- Re-implement the CRUD operations for each table to get work with the graph database
- Map 5 entity sets and their relationships from your relational database to the equivalent structure in graph database
- Reimplement the queries from assignment 2 using cypher queries and show the results in your application

1. An employee is underworking if he is working less than 5 hours on his projects, he is working normally if he is working between 5 and 20 hours, and he is overworking if he is working more than 20 hours. Output the number of employees that are overworking per project.

2. Find the total working hours of all employees and average working hours per employee.
3. Find the total fee of an employee. This number is obtained by multiplying the working hours on all the projects by the hour fee of his position.