MySQL DB assignment: Company Positions Database

Purpose: The purpose of the assignment is the implementation of an application that stores data in a MySQL database.

STEP 1: create a new database and name it with a **name composed of your surname** and **student ID.** (example: Bertolotto123456)**.**

STEP 2: your database will represent the following **scenario**: “Candidates are called for interviews by companies who have open positions with specific requirements”.

Your database must include the following information and **may include any other information that you consider necessary for representing the concepts and implementing the queries listed below**:

- **Candidate details:** CandidateID, Name, address, telephone number, skills (administrative, managerial, etc.).
- **Position details:** PositionID, company, type of position, and skills required.
- **Company details:** CompanyID, Company name, reference person’s name, address and telephone number.
- **Interview details:** You must decide what information should be used to best represent this concept based on the constraints and information provided below.
- **Constraints:**
    - One company can request many interviews for a position.
    - One candidate can be called for many interviews in relation to a position.
    - One company can hire many candidates in relation to a position.
    - Each candidate may only be hired by one company and in relation to only one position.
    - Each candidate can have many skills.
    - Each position can require many skills.

NOTE: You must create table(s) and relationships that will allow you to represent the fact that **interviews** occur on particular dates, and whether a candidate is hired for a position after an interview or not (i.e., whether an interview has been successful).

STEP 3: For every table, create a stored procedure that includes a parametric query that allows you to insert a new row in such a table.

STEP 4: Implement the following queries (some of which are parametric) using stored procedures:

- Find the candidates with a given name.
- Find the candidates with a given CandidateID.
- Find the companies with a given name.
- Find the companies with a given CompanyID.
- Find the candidates who have a skill required by a given PositionID.
- Find the positions with a given PositionID.
- Find the positions requiring a given skill.
- Find the number of positions that require administrative skills.
- Find the interviews that occurred on a particular date.
- Find the name and CandidateID of candidates that were interviewed at least twice.
- Sort the positions according to the companies who are offering them.

STEP 5: export your database onto a self-contained .sql file **which should have the same name as your database** (example: Bertolotto123456.sql).

STEP 6: prepare the **related documentation** as detailed in the next page (**deliverable 2).**

Rules

Each section and subsection of the deliverables must be completed **individually**. Failure to complete any subsection of the deliverables will result in deduction of marks allocated for that subsection. **All questions should be directed to the demonstrators during lab hours.** Late submission will result in a deduction of marks unless agreed in writing with the lecturer at least two days prior to submission.

Due date: The final date for submission of the project is **Monday November 16th, 2015 at 12 p.m. (noon).**

SUBMISSION INSTRUCTIONS: The deliverables (.sql file + documentation pdf file as described below) must be submitted to CSI Moodle. In the course page, you will be able to submit your two files using the "**Submit Practical Assignment**" button (submissions via email will not be accepted).

Deliverables:

1. The completed database, implemented using MySQL, exported and saved in a self-contained .sql file (**which should have the same name as your database as detailed in STEP 5**), MUST contain the following:

   - Tables used to correctly represent all concepts as described above (with appropriate primary keys, constraints, etc.) including additional attributes necessary to link the tables according to the required relationships (and any other assumptions you made).

   - Appropriate data types for all attributes and primary key(s).

   - Tables should be populated with **at least** 10 rows per table.

   - Correct queries as per information sheet implemented by means of stored procedures.

   REMEMBER TO CALL YOUR DATABASE WITH A NAME CONTAINING YOUR SURNAME AND STUDENT NUMBER (eg. Bertolotto123456) AND THE FINAL SELF-CONTAINED .SQL FILE ACCORDINGLY.

2. **Supporting documentation** MUST be provided in one single PDF file including:
   - A short description of any assumptions made or additions to the information provided (e.g. reaction policies used and why they were used, etc.).

   - The Entity-Relationship (ER) diagram of your database (generated as described in lab Exercise 5).

   - **CALL YOUR .PDF FILE WITH THE SAME NAME AS YOUR DATABASE.**