Phase III

Database implementation

Data Modelling and Databases

Specifications

You will implement the ER diagrams from the previous phase using one of the following relational databases management systems: MySQL, PostgreSQL, SQLite or SQL Server Express. The programming part of this project must be implemented in Python, Java or C. You will need to implement the structure of the database (tables with relations), provide sample data for your tables and provide an implementation of SELECT queries (see below for details).

Goals

1. **Create database structure:** convert your ER diagram into a real database using Data Definition Language (DDL). Keep in mind that your database structure must also preserve integrity and consistency of your data and must be in the **third normal form.**
2. **Populate your database with sample data:** provide sample data for each table (HINT: writing a script to generate data might save you considerable time).
3. **Implement SELECT queries:** write a script (preferred Python 3.X) that will interact with your database and perform select queries shown in the next section.

Select Queries

You will find five statements - each one describes a scenario of interaction with your database using SQL's SELECT statements. Please note that **your ER diagram might require some modifications** in order to allow these tasks.

HINT: We do not require you to implement all tasks using pure SQL solutions (e.g. you may combine SQL and Python processing). However, **full points will be awarded to pure SQL solutions**.

1. A patient claims that she forgot her bag in the room where she had a medical appointment on the last time she came to the hospital. The problem is that she had several appointments on that same day. She believes that the doctor's name (first or last name, but not both) starts with "M" or "L" - she doesn't have a good memory either. Find all the possible doctors that match the description.

2. The hospital management team wants to get statistics on the appointments per doctors. For each doctor, the report should present the total and average number of appointments in each time slot of the week during the last year. For example, a report generated on 01/12/2019 should consider data since 01/12/2018.

3. The hospital wants to retrieve information on the patients who had an appointment during the previous month. However, an information which is relevant for some managers is to find which patients visited the hospital every week, at least twice a week. Such patients probably should receive home visits from doctors.

4. Managers want to project the expected monthly income if the hospital start to charge a small value from each patient. The value per appointment would depend on the age and the number of appointments per month. The rules are summarised as follows:

	# appointments in a month < 3	# appointments in a month >= 3
Age < 50	200 Rub	250 Rub
Age >= 50	400 Rub	500 Rub

Based on the rules above, what would be the income of the hospital in the previous month?

5. The managers want to reward experienced and long serving doctors. For that, they want to find out the doctors who have attended at least five patients per year for the last 10 years. Also, such doctors should have had attended a total of at least 100 patients in this period.

Deliverables and grading

1. A SQL dump file for creation and population of the database (called hospital) to be executed on PostgreSQL without errors **(20 points)**
    a. It should include the creation of tables and views
    b. It should include the INSERT statements
2. SQL queries for the specified statements **(50 points)**
3. Database implemented in two different RDBMS (e.g. PostgreSQL and MySQL) (10 points)
4. Python, Java or C implementation **(20 points)**
    a. Script for an automatic and pseudo-random population of the database. The output is a textual sequence of INSERT instructions to be added to the SQL dump file.
    b. User interface allowing the user to run the specified queries.
5. Updated schema and appropriate design decisions (e.g. data types, foreign keys, etc) **(10 points)**
6. The instructor will grade his general impression of the quality, completeness and professionalism of the work, from **0 to 10 points**, **on top of the final score (110/100 points are available).**

Submission

Submission is in Moodle, only **one person per team** should submit.

Make sure that your submission includes the names of all the team members.

Deadline

28/11/2019.