Sistemas de Informação e Bases de Dados
Assignment 2 - Implementing the Database

November 2019

1 The Database Schema

The following relational model is a database schema for the information system of a dental clinic, inspired by what you modeled in Part 1 of the project.

```
employee(VAT,name,birth_date,street,city,zip,IBAN,salary)
IC: All employees are either receptionists, nurses or doctors
IC: IBAN is a candidate key
IC: Salary is a positive number

phone_number_employee(VAT,phone)
VAT: FK(employee)

receptionist(VAT)
VAT: FK(employee)

doctor(VAT,specialization,biography,e-mail)
VAT: FK(employee)
IC: All doctors are either trainees or permanent
IC: E-mail is a candidate key

nurse(VAT)
VAT: FK(employee)

client(VAT,name,birth_date,street,city,zip,gender,age)
IC: Age is a positive number, derived from the birth date

phone_number_client(VAT,phone)
VAT: FK(client)

permanent_doctor(VAT,years)
VAT: FK(doctor)
```

trainee_doctor(VAT,supervisor)
VAT: FK(doctor)
supervisor: FK(permanent_doctor)

supervision_report(VAT,date_timestamp,description,evaluation)
VAT: FK(trainee_doctor)
IC: evaluation is a number in the range from 1 to 5

appointment(VAT_doctor,date_timestamp,description,VAT_client)
VAT_doctor: FK(doctor)
VAT_client: FK(client)

consultation(VAT_doctor,date_timestamp,SOAP_S,SOAP_O,SOAP_A,SOAP_P)
VAT_doctor,date_timestamp: FK(appointment)
IC: Consultations are always assigned to at least one assistant nurse

consultation_assistant(VAT_doctor,date_timestamp,VAT_nurse)
VAT_doctor,date_timestamp: FK(consultation)
VAT_nurse: FK(nurse)

diagnostic_code(ID,description)

diagnostic_code_relation(ID1,ID2,type)
ID1: FK(diagnostic_code)
ID2: FK(diagnostic_code)

consultation_diagnostic(VAT_doctor,date_timestamp,ID)
VAT_doctor,date_timestamp: FK(consultation)
ID: FK(diagnostic_code)

medication(name,lab)

prescription(name,lab,VAT_doctor,date_timestamp,ID,dosage,description)
VAT_doctor,date_timestamp,ID: FK(consultation_diagnostic)
name,lab: FK(medication)

procedure(name,type)

procedure_in_consultation(name,VAT_doctor,date_timestamp,description)
name: FK(procedure)
VAT_doctor,date_timestamp: FK(consultation)

```
procedure_radiology(name,file,VAT_doctor,date_timestamp)
name,VAT_doctor,date_timestamp: FK(procedure_in_consultation)

teeth(quadrant,number,name)

procedure_charting(name,VAT,date_timestamp,quadrant,number,desc,measure)
name,VAT,date_timestamp: FK(procedure_in_consultation)
quadrant,number: FK(teeth)
```

Part 2 of the project consists of implementing a relational database with this schema, and designing SQL queries for answering some relevant information needs based on the stored data.

2 Expected Results

In this assignment, you are expected to provide the following results:

- For the relational model above, write the SQL instructions to create the database in IST's database server (i.e., MySQL on db.ist.utl.pt). You should choose the most appropriate SQL data types for each column.
- Write a SQL script to populate the tables of the relational database with meaningful records of your choice, that you should design to ensure that we can validate the answers to the next questions.
- Write SQL queries for each of the following information needs:
    1. List the VAT, name, and phone number(s) for all clients that had consultations with the doctor named *Jane Sweettooth*. The list should be presented according to the alphabetical order for the names.
    2. List the name of all trainee doctors with reports associated to an evaluation score below the value of three, or with a description that contains the term *insufficient*. The name should be presented together with the VAT of the trainee, the name for the doctor that made the evaluation, the evaluation score, and the textual description for the evaluation report. Results should be sorted according to the evaluation score, in descending order.
    3. List the name, city, and VAT for all clients where the most recent consultation has the *objective* part of the SOAP note mentioning the terms *gingivitis* or *periodontitis*.
    4. List the name, VAT and address (i.e., street, city and zip) of all clients of the clinic that have had appointments but that never had a consultation (i.e., clients that never showed to an appointment).
    5. For each possible diagnosis, presenting the code together with the description, list the number of distinct medication names that have been prescribed to treat that condition. Sort the results according to the number of distinct medication names, in ascending order.

6. Present the average number of nurses/assistants, procedures, diagnostic codes, and prescriptions involved in consultations from the year 2019, respectively for clients belonging to two age groups: less or equal to 18 years old, and more than 18 years old.
7. For each diagnostic code, present the name of the most common medication used to treat that condition (i.e., the medication name that more often appears associated to prescriptions for that diagnosis).
8. List, alphabetically, the names and labs for the medications that, in the year 2019, have been used to treat "dental cavities", but have not been used to treat any "infectious disease". You can use the aforementioned names for searching diagnostic codes in the dataset, without considering relations (e.g., part-of relations) between diagnostic codes.
9. List the names and addresses of clients that have never missed an appointment in 2019 (i.e., the clients that, in the year 2019, have always appeared in all the consultations scheduled for them).

- Suggest database indexes that could be used to improve the performance of the first two queries, from the list of information needs. Justify your choice, and provide SQL instructions for implementing the indexes.
- Write SQL instructions for each of the following changes in the database:

1. Change the address of the doctor named *Jane Sweettooth*, to a different city and street of your choice.
2. Change the salary of all doctors that had more than 100 appointments in 2019. The new salaries should correspond to an increase in 5% from the old values.
3. Delete the doctor named *Jane Sweettooth* from the database, removing also all the appointments and all the consultations (including the associated procedures, diagnosis and prescriptions) in which she was involved. Notice that if there are procedures/diagnosis that were only performed/assigned by this doctor, you should remove them also from the database.
4. Find the diagnosis code corresponding to *gingivitis*. Create also a new diagnosis code corresponding to *periodontitis*. Change the diagnosis from *gingivitis* to *periodontitis* for all clients where, for the same consultation/diagnosis, a dental charting procedure shows a value above 4 in terms of the average gap between the teeth and the gums.

- Create views over the tables in the database model, corresponding to the following relational schema.

```
dim_date(date_timestamp,day,month,year)
IC: date_timestamp corresponds to a date existing in consultations

dim_client(VAT,gender,age)
VAT: FK(client)

dim_location_client(zip,city)
IC: zip corresponds to a zip code existing in clients

facts_consults(VAT,date,zip,num_procedures,num_medications,num_diagnostic_codes)
VAT: FK(dim_client)
date: FK(dim_date)
zip: FK(dim_location_client)
```

  Present the SQL code for creating each of the views corresponding to the tables in the previous model, so that the views feature the corresponding records in the database (i.e., information on all the clients that had consultations, together with the associated number of procedures, number of diagnostic codes, and number of prescribed medications).

3 Submission Notes

A report for the 2nd assignment should be submitted to Fénix as a single PDF file, readable with a standard program such as Adobe Reader. The report should have one separate section for each of the aforementioned expected results.

The document cover page should mention the names, student numbers, and group number of its authors. Provide notes explaining the rationale behind non-trivial implementation decisions.