# GG-e2-project

December 6, 2023

## 1 GG E2 Project


- istXXXXXXX1 FirstName1 LastName1 (33%)
- istXXXXXXX2 FirstName2 LastName2 (33%)
- istXXXXXXX3 FirstName3 LastName3 (33%)

Prof. FirstName LastName

Lab Shift number: PBXX


### 1.1 PART I – Database Schema

**1. The Relational Model** 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.

**Relational Model** client(VAT, name, birth_date, street, city, zip, gender)

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

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)

nurse(VAT) > VAT: FK(employee)

doctor(VAT, specialization, biography, email) > VAT: FK(employee)
> IC: All doctors are either trainees or permanent
> IC: Email is a candidate key

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, VAT_client, description) > 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(VAT_doctor, date_timestamp, ID, name, lab, 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)

teeth(quadrant, number, name)

procedure_charting(name, VAT_doctor, date_timestamp, quadrant, number, desc, measure) > name, VAT_doctor, date_timestamp: FK(procedure_in_consultation)
> quadrant, number: FK(teeth)

procedure_imaging(name, VAT_doctor, date_timestamp, file) > name, VAT_doctor, date_timestamp: FK(procedure_in_consultation)

**2. The Database Schema** For the relational model above, write the SQL instructions to create the database in the PostgreSQL database server. You should choose the most appropriate SQL data types for each column.

You can create the database `db` in Postgres using the instructions in Lab 01.

[ ]:
```
%load_ext sql
%sql postgresql+psycopg://db:db@postgres/db
```

[ ]:
```
%%sql

-- DROP TABLE
```

```
-- CREATE TABLE
```

**3. Populate the Database** 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.

[ ]:
```
%%sql

-- INSERT INTO
```

## 1.2 PART II – SQL

**1. SQL Queries** 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.

[ ]:
```
%%sql
```

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.

[ ]:
```
%%sql
```

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.

[ ]:
```
%%sql
```

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).

[ ]:
```
%%sql
```

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.

[ ]:
```
%%sql
```

6. 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).

[ ]:
```
%%sql
```

7. 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.

[ ]:
```
%%sql
```

8. 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).

[ ]:
```
%%sql
```

**2. SQL Updates and Deletes** 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.

[ ]:
```
%%sql
```

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.

[ ]:
```
%%sql
```

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.

[ ]:
```
%%sql
```

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.

[ ]:
```
%%sql
```

### 1.3 PART III - Functions, Stored Procedures, and Triggers

**1. Functions and Stored Procedures** Provide the SQL instructions corresponding to each of the aforementioned tasks:

1. Write a function to compute the total number of no-shows (i.e., appointments where the client missed the consult) for clients of a given gender, within a given age group, and within

a given year (i.e., the gender, year, and upper/lower limits for the age should all be provided as parameters).

[ ]:
```
%%sql
```

2. Write a stored procedure to change the salary of all doctors that have been practicing for more than x years, where x is an input parameter. The new salary should correspond to a raise of 10 percent over the original salary, in the case of doctors with more than 100 consults in the current year, and to a raise of 5 percent otherwise.

[ ]:
```
%%sql
```

**2. Triggers** Provide the SQL instructions corresponding to each of the aforementioned tasks:

1. Write triggers to ensure that (a) an individual that is a receptionist or a nurse at the clinic cannot simultaneously be a doctor, and (b) doctors cannot simultaneously be trainees and permanent staff.

[ ]:
```
%%sql
```

2. Write triggers to ensure that different individuals (doctors or clients) cannot have the same phone number.

[ ]:
```
%%sql
```