02 E2 Project

Information Systems and Databases

Instituto Superior Técnico

December 16, 2023

Group 2:

- 66325 Tomás Marques Videira Fonseca (100%) tomas.mvf@gmail.com
- 96135 Afonso Brito Caiado Correia Alemão (100%) afonso.alemao@tecnico.ulisboa.pt
- 96317 Rui Pedro Canário Daniel (100%) ruipcdaniel@tecnico.ulisboa.pt

Teachers:

- Flávio Martins
- Alessandro Gianola
- Francisco Regateiro

Lab Shift number: PB03

Project E2

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.

In [1]:
```
%load_ext sql
%sql postgresql+psycopg://db:db@postgres/db
```

In [2]:
```
%%sql

/* Drop all tables */
DROP TABLE IF EXISTS procedure_imaging;
DROP TABLE IF EXISTS procedure_charting;
DROP TABLE IF EXISTS teeth;
DROP TABLE IF EXISTS procedure_in_consultation;
DROP TABLE IF EXISTS procedure;
DROP TABLE IF EXISTS prescription;
DROP TABLE IF EXISTS medication;
DROP TABLE IF EXISTS consultation_diagnostic;
DROP TABLE IF EXISTS diagnostic_code_relation;
DROP TABLE IF EXISTS diagnostic_code;
DROP TABLE IF EXISTS consultation_assistant;
DROP TABLE IF EXISTS consultation;
DROP TABLE IF EXISTS appointment;
DROP TABLE IF EXISTS supervison_report;
DROP TABLE IF EXISTS trainee_doctor;
DROP TABLE IF EXISTS permanent_doctor;
DROP TABLE IF EXISTS doctor;
DROP TABLE IF EXISTS nurse;
DROP TABLE IF EXISTS receptionist;
DROP TABLE IF EXISTS phone_number_employee;
DROP TABLE IF EXISTS employee;
DROP TABLE IF EXISTS phone_number_client;
DROP TABLE IF EXISTS client;

CREATE TABLE client(
    VAT VARCHAR(20),
    name VARCHAR(80) NOT NULL,
    birth_date DATE NOT NULL,
    street VARCHAR(255) NOT NULL,
    city VARCHAR(30) NOT NULL,
    zip VARCHAR(12) NOT NULL,
    gender CHAR(1) NOT NULL,
    PRIMARY KEY(VAT),
    CHECK(LENGTH(zip) >= 2)
);

CREATE TABLE phone_number_client(
    VAT VARCHAR(20),
    phone VARCHAR(15),
    PRIMARY KEY(VAT, phone),
    FOREIGN KEY(VAT) REFERENCES client(VAT),
    CHECK(LENGTH(phone) >= 3)
);

CREATE TABLE employee(
    VAT VARCHAR(20),
    name VARCHAR(80) NOT NULL,
    birth_date DATE NOT NULL,
    street VARCHAR(255) NOT NULL,
    city VARCHAR(30) NOT NULL,
    zip VARCHAR(12) NOT NULL,
    IBAN VARCHAR(30) NOT NULL,
    salary NUMERIC(16,4) NOT NULL,
    PRIMARY KEY(VAT),
    UNIQUE(IBAN),
    CHECK(salary > 0),
    CHECK(LENGTH(zip) >= 2)
    /* -- No Employee can exist at the same time in both the table 'nurse'
    and in the table 'doctor' */
    /* -- No Employee can exist at the same time in both the table 'receptionist'
    and in the table 'doctor' */
    /* -- No Employee can exist at the same time in both the table 'nurse' and
    in the table 'receptionist' */
    /* -- Every Employee must exist either in the table 'nurse' or in the table 'doctor'
    or in the table 'receptionist' */
);

CREATE TABLE phone_number_employee(
    VAT VARCHAR(20),
    phone VARCHAR(15),
    PRIMARY KEY(VAT, phone),
    FOREIGN KEY(VAT) REFERENCES employee(VAT),
    CHECK(LENGTH(phone) >= 3)
);

CREATE TABLE receptionist(
    VAT VARCHAR(20),
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES employee(VAT)
);

CREATE TABLE nurse(
    VAT VARCHAR(20),
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES employee(VAT)
);

CREATE TABLE doctor(
    VAT VARCHAR(20),
    specialization VARCHAR(200) NOT NULL,
    biography TEXT NOT NULL,
    email VARCHAR(254) NOT NULL,
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES employee(VAT),
    UNIQUE(email),
    CHECK(LENGTH(email) >= 6)
    /* -- No Doctor can exist at the same time in both the table 'permanent_doctor'
    and in the table 'trainee_doctor' */
    /* -- Every Doctor must exist either in the table 'permanent_doctor' or
    in the table 'trainee_doctor' */
);

CREATE TABLE permanent_doctor(
    VAT VARCHAR(20),
    years INTEGER NOT NULL,
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES doctor(VAT),
    CHECK(years >= 0)
);

CREATE TABLE trainee_doctor(
    VAT VARCHAR(20),
    supervisor VARCHAR(20) NOT NULL,
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES doctor(VAT),
    FOREIGN KEY(supervisor) REFERENCES permanent_doctor(VAT)

);

CREATE TABLE supervison_report(
    VAT VARCHAR(20),
    date_timestamp TIMESTAMP,
    description TEXT NOT NULL,
    evaluation NUMERIC(3,2) NOT NULL,
    PRIMARY KEY(VAT, date_timestamp),
    FOREIGN KEY(VAT) REFERENCES trainee_doctor(VAT),
    CHECK(evaluation >= 1 and evaluation <= 5)
);

CREATE TABLE appointment(
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    VAT_client VARCHAR(20) NOT NULL,
```

```sql
    description TEXT NOT NULL,
    PRIMARY KEY(VAT_doctor, date_timestamp),
    FOREIGN KEY(VAT_doctor) REFERENCES doctor(VAT),
    FOREIGN KEY(VAT_client) REFERENCES client(VAT)
);

CREATE TABLE consultation(
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    SOAP_S TEXT NOT NULL,
    SOAP_O TEXT NOT NULL,
    SOAP_A TEXT NOT NULL,
    SOAP_P TEXT NOT NULL,
    PRIMARY KEY(VAT_doctor, date_timestamp),
    FOREIGN KEY(VAT_doctor, date_timestamp)
        REFERENCES appointment(VAT_doctor, date_timestamp)
    /* -- Consultations are always assigned to at least one assistant nurse */
);

CREATE TABLE consultation_assistant(
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    VAT_nurse VARCHAR(20) NOT NULL,
    PRIMARY KEY(VAT_doctor, date_timestamp),
    FOREIGN KEY(VAT_doctor, date_timestamp)
        REFERENCES consultation(VAT_doctor, date_timestamp),
    FOREIGN KEY(VAT_nurse) REFERENCES nurse(VAT)
);

CREATE TABLE diagnostic_code(
    ID VARCHAR(7),
    description TEXT NOT NULL,
    PRIMARY KEY(ID)
);

CREATE TABLE diagnostic_code_relation(
    ID1 VARCHAR(7),
    ID2 VARCHAR(7),
    type VARCHAR(200) NOT NULL,
    PRIMARY KEY(ID1, ID2),
    FOREIGN KEY(ID1) REFERENCES diagnostic_code(ID),
    FOREIGN KEY(ID2) REFERENCES diagnostic_code(ID)
);

CREATE TABLE consultation_diagnostic(
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    ID VARCHAR(7),
    PRIMARY KEY(VAT_doctor, date_timestamp, ID),
    FOREIGN KEY(VAT_doctor, date_timestamp)
        REFERENCES consultation(VAT_doctor, date_timestamp),
    FOREIGN KEY(ID) REFERENCES diagnostic_code(ID)
);

CREATE TABLE medication(
    name VARCHAR(255),
    lab VARCHAR(200),
    PRIMARY KEY(name, lab)
);

CREATE TABLE prescription(
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    ID VARCHAR(7),
    name VARCHAR(255),
    lab VARCHAR(200),
    dosage VARCHAR(80) NOT NULL,
    description TEXT NOT NULL,
    PRIMARY KEY(VAT_doctor, date_timestamp, ID, name, lab),
    FOREIGN KEY(VAT_doctor, date_timestamp, ID)
        REFERENCES consultation_diagnostic(VAT_doctor, date_timestamp, ID),
    FOREIGN KEY(name, lab) REFERENCES medication(name, lab)

);
```

```
CREATE TABLE procedure(
    name VARCHAR(200),
    type VARCHAR(150) NOT NULL,
    PRIMARY KEY(name)
);

CREATE TABLE procedure_in_consultation(
    name VARCHAR(200),
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    description TEXT NOT NULL,
    PRIMARY KEY(name, VAT_doctor, date_timestamp),
    FOREIGN KEY(name) REFERENCES procedure(name),
    FOREIGN KEY(VAT_doctor, date_timestamp)
        REFERENCES consultation(VAT_doctor, date_timestamp)

);

CREATE TABLE teeth(
    quadrant CHAR(1),
    number CHAR(1),
    name VARCHAR(200) NOT NULL,
    PRIMARY KEY(quadrant, number)
);

CREATE TABLE procedure_charting(
    name VARCHAR(200),
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    quadrant CHAR(1),
    number CHAR(1),
    description TEXT NOT NULL,
    measure NUMERIC(6,2) NOT NULL,
    PRIMARY KEY(name, VAT_doctor, date_timestamp, quadrant, number),
    FOREIGN KEY(name, VAT_doctor, date_timestamp)
        REFERENCES procedure_in_consultation(name, VAT_doctor, date_timestamp),
    FOREIGN KEY(quadrant, number) REFERENCES teeth(quadrant, number)

);

CREATE TABLE procedure_imaging(
    name VARCHAR(200),
    VAT_doctor VARCHAR(20),
    date_timestamp TIMESTAMP,
    file TEXT,
    PRIMARY KEY(name, VAT_doctor, date_timestamp, file),
    FOREIGN KEY(name, VAT_doctor, date_timestamp)
        REFERENCES procedure_in_consultation(name, VAT_doctor, date_timestamp)

);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[2]:

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.

In [3]:
```
%%sql

INSERT INTO client (VAT, name, birth_date, street, city, zip, gender)
VALUES
    ('123456789', 'John Doe', '1990-05-15', '123 Pink St', 'New York', '1006119', 'M'),
    ('987654321', 'Margarida Corceiro', '2002-10-26', '456 Viana da Mota St', 'Lisbon', '9001001', 'F'),
    ('537450341', 'Marylin Monroe', '1985-10-20', ' St', 'Los Angeles', '9005561', 'F'),
    ('567890123', 'Tobey Maguire', '1995-03-07', '789 Oak St', 'Chicago', '6060551', 'M'),
    ('345678901', 'Tomás Fonseca', '1980-12-12', '101 Pine St', 'Lisbon', '7700552', 'M'),
    ('904444567', 'Rui Daniel', '2001-07-12', '210 Estalagem St', 'Rosário', '3315581', 'M'),
    ('905234567', 'Afonso Alemão', '2001-06-12', '210 Viana da Mota St', 'Lisbon', '3310551', 'M'),
    ('234567850', 'Júlio Paisana', '1969-04-20', '777 Elm St', 'Bora Bora', '9413301', 'M'),
    ('457054321', 'Billie Eilish', '1999-11-06', '456 Viana da Mota St', 'Lisbon', '9001005', 'F'),
    ('357652351', 'Mariana Almeida', '2001-06-13', '333 Big St', 'Mexico City', '9035521', 'F'),
    ('255152351', 'Sofia Almeida', '2002-10-26', '456 Viana da Mota St', 'Lisbon', '8201201', 'F'),
    ('181657351', 'Jessica Silva', '1985-10-20', '11 Carlos Mardel St', 'Coimbra', '2304521', 'F'),
    ('182797366', 'Mantorras', '1982-03-18', '11 Lisboa St', 'Lisbon', '2324522', 'M'),
    ('3083334733', 'Luis Vieira', '1960-03-04', '121 Estalagem St', 'Prisa', '10104', 'M');

INSERT INTO employee (VAT, name, birth_date, street, city, zip, IBAN, salary)
VALUES
    ('9121512141', 'Jane Sweettooth', '1992-08-20', '789 Oak St', 'Chicago', '60601', '123656889', 4000.00),
    ('7222324262', 'David Smith', '1985-03-15', '456 Elm St', 'Los Angeles', '90001', '987654321', 2000.00),
    ('0135012541', 'Emily Davis', '1980-11-10', '123 Main St', 'New York', '10001', '563890123', 4300.00),
    ('7322220202', 'Dolores Aveiro', '1977-06-12', '127 Main St', 'New York', '10003', '537093123', 5700.00),
    ('2032062803', 'Stone Cold', '1970-07-10', '128 Main St', 'Lisbon', '10004', '267830223', 2800.00),
    ('7423225472', 'Ed Sheeran', '1977-02-12', '127 Main St', 'New York', '10003', '537891123', 5700.00),
    ('2042462003', 'Tate McRae', '2001-03-17', '128 Estalagem St', 'Rosário', '10004', '367835223', 2800.00),
    ('3043042703', 'Cristiano Ronaldo', '2005-03-04', '128 Estalagem St', 'Rosário', '10004', '367330223', 280000.00),
    ('3083334733', 'Luis Vieira', '1960-03-04', '121 Estalagem St', 'Prisa', '10104', '317311123', 900.00);

INSERT INTO receptionist (VAT)
VALUES ('3043042703');

INSERT INTO nurse (VAT)
VALUES
    ('2042462003'),
    ('7423225472');

INSERT INTO doctor (VAT, specialization, biography, email)
VALUES
    ('9121512141', 'Orthodontics', 'Dr. Sweettooth specializes in Orthodontics.', 'sweet@gmail.com'),
    ('7222324262', 'Endodontics', 'Dr. Smith is passionate about children health
        and has been practicing pediatrics for over a decade.', 'smith@gmail.com'),
    ('0135012541', 'Periodontics', 'Dr. Davis specializes in Periodontics.', 'davis@gmail.com'),
    ('7322220202', 'Prosthodontics', 'Dr. Aveiro loves multi-tasking', 'aveiro@gmail.com'),
    ('2032062803', 'Pediatric Dentistry', 'Dr. Cold is dedicated to understanding Pediatric Dentistry.',
        'cold@gmail.com'),
    ('3083334733', 'Pediatric Dentistry', 'Dr. Vieira is dedicated to understanding Pediatric Dentistry.',
        'presi@gmail.com');


INSERT INTO permanent_doctor (VAT, years)
VALUES
    ('7322220202', 10),
    ('2032062803', 5),
    ('0135012541', 7),
    ('3083334733', 9);


INSERT INTO trainee_doctor (VAT, supervisor)
VALUES
    ('9121512141', '7322220202'),
    ('7222324262', '2032062803');

INSERT INTO supervison_report (VAT, date_timestamp, description, evaluation)
VALUES
    ('9121512141', '2023-01-15 09:00:00', 'Insufficient performance.', 3),
    ('9121512141', '2023-02-10 10:30:00', 'Bad performance', 1),
    ('7222324262', '2023-01-20 11:15:00', 'Sufficient diagnostic skills observed.', 3),
    ('7222324262', '2023-03-05 14:00:00', 'Excellent diagnostic skills observed.', 5);

INSERT INTO phone_number_client (VAT, phone)
VALUES
    ('123456789', '912345678'),
    ('987654321', '939876543'),
    ('537450341', '953745034'),
    ('567890123', '956789012'),
    ('345678901', '934567890'),
    ('345678901', '914464852'),
    ('345678901', '924367491'),
    ('904444567', '990444456'),
    ('905234567', '990523456'),
    ('234567850', '923456785'),
    ('457054321', '945705432'),
    ('357652351', '935765235'),
    ('255152351', '925515235'),
    ('255152351', '963455631'),
    ('181657351', '918165735');
```

```sql
INSERT INTO phone_number_employee (VAT, phone)
VALUES
    ('9121512141', '919121512'),
    ('9121512141', '919121513'),
    ('7222324262', '972223242'),
    ('0135012541', '901350125'),
    ('7322220202', '973222020'),
    ('2032062803', '920320628'),
    ('7423225472', '917423225'),
    ('7423225472', '917423323'),
    ('2042462003', '920424620'),
    ('3043042703', '913043042'),
    ('3083334733', '929292929');

INSERT INTO appointment (VAT_doctor, date_timestamp, VAT_client, description)
VALUES
    ('9121512141', '2023-12-15 10:00:00', '123456789', 'Regular checkup'),
    ('7222324262', '2023-12-16 11:00:00', '987654321', 'Regular checkup'),
    ('0135012541', '2023-12-17 09:30:00', '567890123', 'Regular checkup'),
    ('7322220202', '2023-12-18 13:45:00', '345678901', 'Regular checkup'),
    ('2032062803', '2023-12-19 14:30:00', '345678901', 'Surgery'),
    ('9121512141', '2023-11-15 10:00:00', '345678901', 'Regular checkup'),
    ('2032062803', '2019-11-07 10:30:00', '357652351', 'Regular checkup'),
    ('9121512141', '2023-11-19 11:15:00', '904444567', 'Regular checkup'),
    ('9121512141', '2023-11-21 09:45:00', '255152351', 'Surgery'),
    ('9121512141', '2023-11-23 08:30:00', '345678901', 'Regular checkup'),
    ('7222324262', '2023-11-26 11:00:00', '457054321', 'Regular checkup'),
    ('7222324262', '2019-11-26 11:00:00', '234567850', 'Regular checkup'),
    ('0135012541', '2019-11-17 09:30:00', '345678901', 'Surgery'),
    ('7322220202', '2023-11-28 13:45:00', '181657351', 'Regular checkup'),
    ('2032062803', '2023-11-29 14:30:00', '357652351', 'Regular checkup'),
    ('2032062803', '2019-11-29 14:30:00', '123456789', 'Regular checkup'),
    ('2032062803', '2019-01-01 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-02 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-03 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-04 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-05 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-06 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-07 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-08 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-09 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-10 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-11 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-12 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-13 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-14 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-15 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-16 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-17 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-18 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-19 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-20 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-21 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-22 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-23 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-24 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-25 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-26 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-27 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-28 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-29 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-30 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-01-31 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-01 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-02 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-03 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-04 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-05 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-06 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-07 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-08 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-09 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-10 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-11 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-12 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-13 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-14 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-15 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-16 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-17 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-18 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-19 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-20 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-21 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-22 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-23 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-24 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-25 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-26 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-27 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-02-28 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-01 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-02 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-03 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-04 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-05 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-06 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-07 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-08 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-09 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-10 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-11 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-12 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-13 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-14 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-15 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-16 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-17 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-18 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-19 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-20 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-21 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-22 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-23 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-24 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-25 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-26 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-27 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-28 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-29 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-30 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-03-31 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-01 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-02 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-03 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-04 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-05 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-06 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-07 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-08 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-09 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-10 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-11 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-12 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-13 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-14 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-15 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-16 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-17 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-18 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-19 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-20 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-21 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-22 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-23 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-24 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-25 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-26 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-27 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-28 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-29 14:30:00', '181657351', 'Daily checkup'),
    ('2032062803', '2019-04-30 14:30:00', '181657351', 'Daily checkup'),
    ('3083334733', '2023-01-01 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-02 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-03 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-04 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-05 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-06 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-07 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-08 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-09 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-10 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-11 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-12 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-13 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-14 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-15 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-16 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-17 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-18 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-19 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-20 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-21 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-22 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-23 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-24 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-25 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-26 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-27 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-28 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-29 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-30 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-01-31 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-01 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-02 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-03 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-04 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-05 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-06 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-07 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-08 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-09 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-10 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-11 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-12 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-13 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-14 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-15 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-16 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-17 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-18 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-19 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-20 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-21 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-22 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-23 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-24 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-25 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-26 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-27 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-02-28 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-01 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-02 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-03 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-04 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-05 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-06 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-07 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-08 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-09 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-10 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-11 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-12 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-13 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-14 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-15 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-16 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-17 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-18 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-19 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-20 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-21 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-22 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-23 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-24 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-25 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-26 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-27 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-28 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-29 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-30 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-03-31 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-01 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-02 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-03 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-04 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-05 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-06 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-07 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-08 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-09 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-10 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-11 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-12 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-13 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-14 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-15 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-16 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-17 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-18 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-19 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-20 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-21 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-22 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-23 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-24 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-25 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-26 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-27 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-28 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-29 14:30:00', '182797366', 'Daily checkup'),
    ('3083334733', '2023-04-30 14:30:00', '182797366', 'Daily checkup');

INSERT INTO consultation (VAT_doctor, date_timestamp, SOAP_S, SOAP_O, SOAP_A, SOAP_P)
VALUES
    ('9121512141', '2023-11-15 10:00:00', 'gingivitis', 'Objective2', 'Assessment23', 'Plan5'),
    ('2032062803', '2019-11-07 10:30:00', 'Subjective4', 'periodontitis', 'Assessment25', 'Plan6'),
    ('9121512141', '2023-11-19 11:15:00', 'Subjective2', 'Objective22', 'Assessment6', 'Plan1'),
    ('9121512141', '2023-11-21 09:45:00', 'Subjective1', 'Objective11', 'Assessment1', 'Plan2'),
    ('9121512141', '2023-11-23 08:30:00', 'periodontitis', 'Objective4', 'Assessment15', 'Plan3'),
    ('7222324262', '2019-11-26 11:00:00', 'Subjective2', 'Objective5', 'Assessment2', 'Plan4'),
    ('7222324262', '2023-11-26 11:00:00', 'Subjective2', 'Objective5', 'Assessment2', 'Plan4'),
    ('0135012541', '2023-12-17 09:30:00', 'Subjective1', 'Objective10', 'gingivitis', 'Plan5'),
    ('7322220202', '2023-11-28 13:45:00', 'Subjective4', 'Objective8', 'Assessment3', 'Plan2'),
    ('2032062803', '2023-11-29 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('0135012541', '2019-11-17 09:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-01 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-02 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-03 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-04 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-05 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-06 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-07 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-08 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-09 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-10 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-11 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-12 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-13 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-14 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-15 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-16 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-17 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-18 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-19 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-20 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-21 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-22 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-23 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-24 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-25 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-26 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-27 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-28 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-29 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-30 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-01-31 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-01 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-02 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-03 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-04 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-05 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-06 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-07 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-08 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-09 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-10 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-11 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-12 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-13 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-14 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-15 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-16 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-17 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-18 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-19 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-20 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-21 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-22 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-23 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-24 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-25 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-26 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-27 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-02-28 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-01 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-02 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-03 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-04 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-05 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-06 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-07 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-08 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-09 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-10 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-11 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-12 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-13 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-14 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-15 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-16 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-17 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-18 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-19 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-20 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-21 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-22 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-23 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-24 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-25 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-26 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-27 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-28 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-29 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
```

```sql
    ('3083334733', '2023-03-30 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-03-31 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-01 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-02 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-03 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-04 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-05 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-06 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-07 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-08 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-09 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-10 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-11 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-12 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-13 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-14 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-15 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-16 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-17 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-18 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-19 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-20 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-21 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-22 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-23 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-24 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-25 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-26 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-27 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-28 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-29 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis'),
    ('3083334733', '2023-04-30 14:30:00', 'Subjective5', 'Objective9', 'Assessment3', 'gingivitis');

INSERT INTO diagnostic_code (ID, description)
VALUES
    ('110011D', 'Dental Cavities'),
    ('110021D', 'Gingivitis'),
    ('110031D', 'Caries'),
    ('110041D', 'Active dental caries'),
    ('110051D', 'Poor Oral Hygiene'),
    ('110061D', 'Overbite'),
    ('110071D', 'Infectious Disease');

INSERT INTO diagnostic_code_relation (ID1, ID2, type)
VALUES
    ('110011D', '110021D', 'Related-To'),
    ('110021D', '110011D', 'Related-To'),
    ('110041D', '110051D', 'Related-To'),
    ('110051D', '110041D', 'Related-To');

INSERT INTO consultation_diagnostic (VAT_doctor, date_timestamp, ID)
VALUES
    ('9121512141', '2023-11-15 10:00:00', '110011D'),
    ('9121512141', '2023-11-15 10:00:00', '110071D'),
    ('2032062803', '2019-11-07 10:30:00', '110021D'),
    ('9121512141', '2023-11-23 08:30:00', '110021D'),
    ('0135012541', '2019-11-17 09:30:00', '110011D'),
    ('0135012541', '2023-12-17 09:30:00', '110011D'),
    ('0135012541', '2023-12-17 09:30:00', '110071D'),
    ('0135012541', '2023-12-17 09:30:00', '110031D'),
    ('2032062803', '2023-11-29 14:30:00', '110011D'),
    ('7222324262', '2019-11-26 11:00:00', '110051D'),
    ('7222324262', '2023-11-26 11:00:00', '110041D'),
    ('7222324262', '2023-11-26 11:00:00', '110071D'),
    ('7222324262', '2023-11-26 11:00:00', '110021D');

INSERT INTO medication (name, lab)
VALUES
    ('Amoxicillin', 'GenericLab'),
    ('Ibuprofen', 'PainReliefCo'),
    ('Chlorhexidine', 'MouthCareInc'),
    ('Paracetamol', 'MouthCareInc'),
    ('Listerine', 'MouthCareInc'),
    ('Elugel', 'MouthCareInc');
```

```sql
INSERT INTO teeth (quadrant, number, name)
VALUES
    ('1', '2', 'Upper Right Lateral Incisor'),
    ('2', '1', 'Upper Left Central Incisor'),
    ('2', '2', 'Upper Left Lateral Incisor'),
    ('3', '1', 'Lower Left Central Incisor'),
    ('3', '2', 'Lower Left Lateral Incisor'),
    ('4', '1', 'Lower Right Central Incisor'),
    ('4', '2', 'Lower Right Lateral Incisor');

INSERT INTO procedure (name, type)
VALUES
    ('Tooth Extraction', 'Surgical'),
    ('Dental Implant', 'Surgical'),
    ('Teeth Cleaning', 'Non-Surgical'),
    ('Root Canal', 'Non-Surgical'),
    ('Cavity Filling', 'Non-Surgical'),
    ('Filling', 'Non-Surgical'),
    ('Cleaning', 'Non-Surgical'),
    ('Braces Adjustment', 'Non-Surgical'),
    ('CT Scan', 'Non-Surgical'),
    ('MRI', 'Non-Surgical'),
    ('Panoramic X-ray', 'Non-Surgical');

INSERT INTO procedure_in_consultation (name, VAT_doctor, date_timestamp, description)
VALUES
    ('Tooth Extraction', '9121512141', '2023-11-15 10:00:00', 'Extraction of wisdom tooth due to impaction'),
    ('Dental Implant', '7222324262', '2019-11-26 11:00:00', 'Implant placement in the upper right quadrant'),
    ('Filling', '0135012541', '2023-12-17 09:30:00', 'Routine dental cleaning to remove plaque and calculus'),
    ('Cleaning', '2032062803', '2019-11-07 10:30:00', 'Root canal treatment on lower left molar'),
    ('Braces Adjustment', '7222324262', '2023-11-26 11:00:00', 'Filling of caries on upper left premolar'),
    ('CT Scan', '2032062803', '2023-11-29 14:30:00', 'CT SCAN'),
    ('MRI', '2032062803', '2023-11-29 14:30:00', 'MRI'),
    ('Panoramic X-ray', '2032062803', '2023-11-29 14:30:00', 'Pan');

INSERT INTO procedure_charting (name, VAT_doctor, date_timestamp, quadrant, number, description, measure)
VALUES
    ('Filling', '0135012541', '2023-12-17 09:30:00', '2', '1', 'Filling for cavity', 5.0),
    ('Filling', '0135012541', '2023-12-17 09:30:00', '3', '1', 'Filling for cavity', 6.0),
    ('Cleaning', '2032062803', '2019-11-07 10:30:00', '1', '2', 'Teeth cleaning', 6.0),
    ('Braces Adjustment', '7222324262', '2023-11-26 11:00:00', '3', '1', 'Routine braces adjustment', 3.0),
    ('Braces Adjustment', '7222324262', '2023-11-26 11:00:00', '2', '1', 'Routine braces adjustment', 4.5);

INSERT INTO procedure_imaging (name, VAT_doctor, date_timestamp, file)
VALUES
    ('CT Scan', '2032062803', '2023-11-29 14:30:00', 'ctscan1.png'),
    ('MRI', '2032062803', '2023-11-29 14:30:00', 'mri1.png'),
    ('Panoramic X-ray', '2032062803', '2023-11-29 14:30:00', 'panoramic_xray.png');

INSERT INTO prescription (VAT_doctor, date_timestamp, ID, name, lab, dosage, description)
VALUES
    ('9121512141', '2023-11-15 10:00:00', '110011D', 'Amoxicillin', 'GenericLab', '500mg every 8 hours',
        'Antibiotic for gingivitis'),
    ('9121512141', '2023-11-15 10:00:00', '110071D', 'Ibuprofen', 'PainReliefCo', '200mg as needed',
        'Pain relief for dental pain'),
    ('9121512141', '2023-11-15 10:00:00', '110011D', 'Chlorhexidine', 'MouthCareInc', 'Use twice daily',
        'Mouthwash for periodontitis'),
    ('2032062803', '2019-11-07 10:30:00', '110021D', 'Chlorhexidine', 'MouthCareInc', 'Use twice daily',
        'Mouthwash for periodontitis'),
    ('0135012541', '2019-11-17 09:30:00', '110011D', 'Amoxicillin', 'GenericLab', '500mg every 8 hours',
        'Antibiotic for gingivitis'),
    ('7222324262', '2019-11-26 11:00:00', '110051D', 'Ibuprofen', 'PainReliefCo', '200mg as needed',
        'Pain relief for dental pain'),
    ('7222324262', '2023-11-26 11:00:00', '110041D', 'Ibuprofen', 'PainReliefCo', '200mg as needed',
        'Pain relief for dental pain'),
    ('7222324262', '2023-11-26 11:00:00', '110071D', 'Chlorhexidine', 'MouthCareInc', 'Use twice daily',
        'Mouthwash for periodontitis'),
    ('0135012541', '2019-11-17 09:30:00', '110011D', 'Paracetamol', 'MouthCareInc', '200mg as needed',
        'Pain relief for dental pain'),
    ('7222324262', '2023-11-26 11:00:00', '110021D', 'Listerine', 'MouthCareInc', '200mg as needed',
        'Pain relief for dental pain'),
    ('0135012541', '2019-11-17 09:30:00', '110011D', 'Elugel', 'MouthCareInc', '200mg as needed',
        'Pain relief for dental pain');
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
```

```
14 rows affected.
9 rows affected.
1 rows affected.
2 rows affected.
6 rows affected.
4 rows affected.
2 rows affected.
4 rows affected.
15 rows affected.
11 rows affected.
256 rows affected.
131 rows affected.
7 rows affected.
4 rows affected.
13 rows affected.
6 rows affected.
7 rows affected.
11 rows affected.
8 rows affected.
5 rows affected.
3 rows affected.
11 rows affected.
```

Out[3]:

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.

In [4]:
```
%%sql

SELECT DISTINCT
    c.VAT, c.name, p.phone
FROM
    client c
    JOIN phone_number_client AS p ON c.VAT = p.VAT
    JOIN appointment AS a ON c.VAT = a.VAT_client
    JOIN consultation AS cc ON cc.VAT_doctor = a.VAT_doctor AND cc.date_timestamp = a.date_timestamp
    JOIN doctor AS d ON cc.VAT_doctor = d.VAT
    JOIN employee AS e ON e.VAT = d.VAT
WHERE
    e.name = 'Jane Sweettooth'
ORDER BY
    c.name ASC;
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
6 rows affected.
```

Out[4]:

vat	name	phone
904444567	Rui Daniel	990444456
255152351	Sofia Almeida	925515235
255152351	Sofia Almeida	963455631
345678901	Tomás Fonseca	914464852
345678901	Tomás Fonseca	924367491
345678901	Tomás Fonseca	934567890

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.

In [5]:
```
%%sql

SELECT
    e2.name AS trainee_name, t.VAT AS trainee_VAT, e.name AS supervisor, s.evaluation, s.description
FROM
    trainee_doctor AS t
    JOIN doctor AS d2 ON d2.VAT = t.VAT
    JOIN employee AS e2 ON e2.VAT = d2.VAT
    JOIN permanent_doctor AS p ON p.VAT = t.supervisor
    JOIN doctor AS d ON d.VAT = p.VAT
    JOIN employee AS e ON e.VAT = d.VAT
    JOIN supervison_report AS s ON s.VAT = t.VAT
WHERE
    s.evaluation < 3 OR s.description ILIKE '%insufficient%'
ORDER BY
    s.evaluation DESC;
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
2 rows affected.
```

Out[5]:

trainee_name	trainee_vat	supervisor	evaluation	description
Jane Sweettooth	9121512141	Dolores Aveiro	3.00	Insufficient performance.
Jane Sweettooth	9121512141	Dolores Aveiro	1.00	Bad performance

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.

In [6]:
```
%%sql

SELECT DISTINCT
    c.name, c.city, c.VAT
FROM
    client c
    JOIN appointment AS a ON c.VAT = a.VAT_client
    JOIN consultation AS cc ON cc.VAT_doctor = a.VAT_doctor AND cc.date_timestamp = a.date_timestamp
WHERE cc.date_timestamp >= ALL(
    SELECT cc1.date_timestamp
    FROM
        client c1
        JOIN appointment AS a1 ON c1.VAT = a1.VAT_client
        JOIN consultation AS cc1 ON cc1.VAT_doctor = a1.VAT_doctor AND cc1.date_timestamp = a1.date_timestamp
    WHERE c1.VAT = c.VAT)
AND (cc.SOAP_S ILIKE '%gingivitis%'
    OR cc.SOAP_O ILIKE '%gingivitis%'
    OR cc.SOAP_A ILIKE '%gingivitis%'
    OR cc.SOAP_P ILIKE '%gingivitis%'
    OR cc.SOAP_S ILIKE '%periodontitis%'
    OR cc.SOAP_O ILIKE '%periodontitis%'
    OR cc.SOAP_A ILIKE '%periodontitis%'
    OR cc.SOAP_P ILIKE '%periodontitis%')
```

```
ORDER BY
    c.name ASC;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

4 rows affected.

Out[6]:

name	city	vat
Mantorras	Lisbon	182797366
Mariana Almeida	Mexico City	357652351
Tobey Maguire	Chicago	567890123
Tomás Fonseca	Lisbon	345678901

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).

In [7]:
```
%%sql
SELECT DISTINCT c.name, c.VAT, c.street, c.city, c.zip
FROM client AS c
WHERE NOT EXISTS(
    SELECT *
    FROM
        appointment AS a
        JOIN consultation AS cc ON cc.VAT_doctor = a.VAT_doctor AND cc.date_timestamp = a.date_timestamp
    WHERE c.VAT = a.VAT_client
) AND EXISTS(
    SELECT *
    FROM
        appointment AS a
    WHERE c.VAT = a.VAT_client);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

2 rows affected.

Out[7]:

name	vat	street	city	zip
John Doe	123456789	123 Pink St	New York	1006119
Margarida Corceiro	987654321	456 Viana da Mota St	Lisbon	9001001

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.

In [8]:
```
%%sql

SELECT DISTINCT d.ID AS diagnostic_ID, d.description AS diagnostic_description, COUNT(d.ID)
    AS number_distinct_medication
FROM diagnostic_code AS d
    JOIN consultation_diagnostic AS cd ON cd.ID = d.ID
    JOIN prescription AS p ON cd.VAT_doctor = p.VAT_doctor
        AND cd.date_timestamp = p.date_timestamp
        AND cd.ID = p.ID
    LEFT JOIN medication AS m ON m.name = p.name AND m.lab = p.lab
GROUP BY d.ID
ORDER BY COUNT(d.ID) ASC;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

5 rows affected.

Out[8]:

diagnostic_id	diagnostic_description	number_distinct_medication
110041D	Active dental caries	1
110051D	Poor Oral Hygiene	1
110021D	Gingivitis	2
110071D	Infectious Disease	2
110011D	Dental Cavities	5

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).

In [9]:
```
%%sql
SELECT
    d.ID AS diagnostic_code,
    m.name AS medication_name
FROM
    diagnostic_code d
JOIN
    prescription p ON d.ID = p.ID
JOIN
    medication m ON p.name = m.name AND p.lab = m.lab
GROUP BY
    d.ID, m.name
HAVING
    COUNT(*) >= ALL (
        SELECT COUNT(*)
        FROM prescription p2
        WHERE p2.ID = d.ID
        GROUP BY p2.name, p2.lab
    )
ORDER BY d.ID;
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
7 rows affected.
```

Out[9]:

diagnostic_code	medication_name
110011D	Amoxicillin
110021D	Chlorhexidine
110021D	Listerine
110041D	Ibuprofen
110051D	Ibuprofen
110071D	Chlorhexidine
110071D	Ibuprofen

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.

In [10]:
```
%%sql

SELECT m.name, m.lab
FROM
    medication as m
    JOIN prescription AS p ON m.name = p.name AND m.lab = p.lab
    JOIN consultation_diagnostic AS cd ON cd.VAT_doctor = p.VAT_doctor
        AND cd.ID = p.ID AND cd.date_timestamp = p.date_timestamp
WHERE TO_CHAR(cd.date_timestamp, 'YYYY') = '2019' AND EXISTS(
    SELECT *
    FROM
        diagnostic_code AS d
    WHERE d.ID = cd.ID
        AND d.description ILIKE '%dental cavities%'
        AND NOT EXISTS(
            SELECT *
            FROM
                diagnostic_code AS d2
            WHERE d2.description ILIKE '%infectious disease%' AND d2.ID = d.ID
        )
);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

3 rows affected.

Out[10]:

name	lab
Amoxicillin	GenericLab
Paracetamol	MouthCareInc
Elugel	MouthCareInc

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).

In [11]:
```
%%sql

SELECT c.name, c.street, c.city, c.zip
FROM client AS c
WHERE NOT EXISTS(
    SELECT a.VAT_doctor, a.date_timestamp
    FROM appointment AS a
    WHERE TO_CHAR(a.date_timestamp, 'YYYY') = '2019' AND a.VAT_client = c.VAT
    EXCEPT
    SELECT cc.VAT_doctor, cc.date_timestamp
    FROM consultation AS cc
    JOIN appointment AS a2 ON cc.VAT_doctor = a2.VAT_doctor AND cc.date_timestamp = a2.date_timestamp
    WHERE TO_CHAR(a2.date_timestamp, 'YYYY') = '2019' AND a2.VAT_client = c.VAT
) AND EXISTS(
    SELECT *
    FROM appointment AS a
    WHERE TO_CHAR(a.date_timestamp, 'YYYY') = '2019' AND a.VAT_client = c.VAT
)
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

3 rows affected.

Out[11]:

name	street	city	zip
Júlio Paisana	777 Elm St	Bora Bora	9413301
Tomás Fonseca	101 Pine St	Lisbon	7700552
Mariana Almeida	333 Big St	Mexico City	9035521

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.

In [12]:
```
%%sql

UPDATE employee
SET city = 'Rosário', street = '12 Estalagem St'
WHERE name LIKE 'Jane Sweettooth';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

1 rows affected.

Out[12]:

In [13]:
```
%%sql

-- SELECT *
-- FROM employee
-- WHERE name LIKE 'Jane Sweettooth';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[13]:

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.

In [14]:
```
%%sql

UPDATE employee AS e
SET salary = e.salary * 1.05
WHERE  100 < (
    SELECT COUNT(*)
    FROM appointment AS a
    JOIN doctor AS d ON d.VAT = a.VAT_doctor
    JOIN employee AS e2 ON e2.VAT = d.VAT
    WHERE e2.VAT = e.VAT and TO_CHAR(a.date_timestamp, 'YYYY') = '2019'
);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

1 rows affected.

Out[14]:

In [15]:
```
%%sql
-- -- Stone Cold had more than 100 appointments in 2019
-- SELECT vat, name, salary
-- FROM employee
-- ORDER BY name ASC;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[15]:

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.

In [16]:
```
%%sql

DELETE FROM supervison_report
WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM phone_number_employee
WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM prescription
WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM consultation_diagnostic
WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM procedure_charting
WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM procedure_in_consultation
WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM consultation
WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM appointment
WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM trainee_doctor
WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');

DELETE FROM doctor
WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

```sql
DELETE FROM employee
WHERE name = 'Jane Sweettooth';
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
2 rows affected.
2 rows affected.
3 rows affected.
3 rows affected.
1 rows affected.
4 rows affected.
5 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
```

Out[16]:

In [17]:
```sql
%%sql

-- SELECT * FROM supervison_report
-- WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
```

Out[17]:

In [18]:
```sql
%%sql

-- SELECT * FROM phone_number_employee
-- WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
```

Out[18]:

In [19]:
```sql
%%sql

-- SELECT * FROM prescription
-- WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
```

Out[19]:

In [20]:
```sql
%%sql

-- SELECT * FROM consultation_diagnostic
-- WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
```

Out[20]:

In [21]:
```sql
%%sql

-- SELECT * FROM procedure_charting
-- WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
```

Out[21]:

In [22]:
```sql
%%sql

-- SELECT * FROM procedure_in_consultation
-- WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

```
Running query in 'postgresql+psycopg://db:***@postgres/db'
```

Out[22]:

In [23]:
```sql
%%sql
```

```
-- SELECT * FROM consultation
-- WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[23]:

In [24]:
```
%%sql

-- SELECT * FROM appointment
-- WHERE VAT_doctor IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[24]:

In [25]:
```
%%sql

-- SELECT * FROM trainee_doctor
-- WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[25]:

In [26]:
```
%%sql

-- SELECT * FROM doctor
-- WHERE VAT IN (SELECT VAT FROM employee WHERE name = 'Jane Sweettooth');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[26]:

In [27]:
```
%%sql

-- SELECT * FROM employee
-- WHERE name = 'Jane Sweettooth';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[27]:

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.

In [28]:
```
%%sql

SELECT d.ID, d.description
FROM diagnostic_code AS d
WHERE d.description ILIKE '%gingivitis%';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

1 rows affected.

Out[28]:

id	description
110021D	Gingivitis

In [29]:
```
%%sql

INSERT INTO diagnostic_code (ID, description)
VALUES
    ('110081D', 'Periodontitis');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

1 rows affected.

Out[29]:

In [30]:
```
%%sql
-- -- Before update
-- SELECT c.VAT_doctor, c.date_timestamp, dc.description, AVG(pc.measure) average_GAP
-- FROM consultation_diagnostic AS c
--     JOIN diagnostic_code AS dc ON dc.ID = c.ID
--     JOIN consultation AS c1 ON c.date_timestamp = c1.date_timestamp
--     JOIN procedure_in_consultation AS pic ON c1.VAT_doctor = pic.VAT_doctor
--         AND c1.date_timestamp = pic.date_timestamp
--     JOIN procedure_charting AS pc ON pic.name = pc.name
--         AND pic.VAT_doctor = pc.VAT_doctor
--         AND pic.date_timestamp = pc.date_timestamp
-- GROUP BY c.VAT_doctor, c.date_timestamp, dc.description
-- ORDER BY c.VAT_doctor, c.date_timestamp, dc.description;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[30]:

In [31]:
```
%%sql

ALTER TABLE prescription DROP CONSTRAINT prescription_vat_doctor_date_timestamp_id_fkey;

UPDATE prescription AS p
SET ID = (SELECT ID FROM diagnostic_code WHERE description ILIKE '%periodontitis%')
WHERE EXISTS(
    SELECT *
    FROM diagnostic_code AS d
    WHERE description ILIKE '%gingivitis%' AND d.ID = p.ID
) AND 4 < (
    SELECT AVG(pc.measure) average_GAP
    FROM consultation_diagnostic AS c
        JOIN diagnostic_code AS dc ON dc.ID = c.ID
        JOIN consultation AS c1 ON c.date_timestamp = c1.date_timestamp
        JOIN procedure_in_consultation AS pic ON c1.VAT_doctor = pic.VAT_doctor
            AND c1.date_timestamp = pic.date_timestamp
        JOIN procedure_charting AS pc ON pic.name = pc.name
            AND pic.VAT_doctor = pc.VAT_doctor
            AND pic.date_timestamp = pc.date_timestamp
     WHERE c.VAT_doctor = p.VAT_doctor AND c.date_timestamp = p.date_timestamp AND dc.ID = p.ID
);

UPDATE consultation_diagnostic AS cd
SET ID = (SELECT ID FROM diagnostic_code WHERE description ILIKE '%periodontitis%')
WHERE EXISTS(
    SELECT *
    FROM diagnostic_code AS d
    WHERE description ILIKE '%gingivitis%' AND d.ID = cd.ID
) AND 4 < (
    SELECT AVG(pc.measure) average_GAP
    FROM consultation_diagnostic AS c
        JOIN diagnostic_code AS dc ON dc.ID = c.ID
        JOIN consultation AS c1 ON c.date_timestamp = c1.date_timestamp
        JOIN procedure_in_consultation AS pic ON c1.VAT_doctor = pic.VAT_doctor
            AND c1.date_timestamp = pic.date_timestamp
        JOIN procedure_charting AS pc ON pic.name = pc.name
            AND pic.VAT_doctor = pc.VAT_doctor
            AND pic.date_timestamp = pc.date_timestamp
     WHERE c.VAT_doctor = cd.VAT_doctor AND c.date_timestamp = cd.date_timestamp AND dc.ID = cd.ID
);

ALTER TABLE prescription
ADD CONSTRAINT prescription_vat_doctor_date_timestamp_id_fkey
    FOREIGN KEY(VAT_doctor, date_timestamp, ID) REFERENCES consultation_diagnostic(VAT_doctor, date_timestamp, ID);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

1 rows affected.

1 rows affected.

Out[31]:

In [32]:
```
%%sql
-- -- After update
-- SELECT c.VAT_doctor, c.date_timestamp, dc.description, AVG(pc.measure) average_GAP
-- FROM consultation_diagnostic AS c
--     JOIN diagnostic_code AS dc ON dc.ID = c.ID
--     JOIN consultation AS c1 ON c.date_timestamp = c1.date_timestamp
--     JOIN procedure_in_consultation AS pic ON c1.VAT_doctor = pic.VAT_doctor
--         AND c1.date_timestamp = pic.date_timestamp
--     JOIN procedure_charting AS pc ON pic.name = pc.name
--          AND pic.VAT_doctor = pc.VAT_doctor
--          AND pic.date_timestamp = pc.date_timestamp
-- GROUP BY c.VAT_doctor, c.date_timestamp, dc.description
-- ORDER BY c.VAT_doctor, c.date_timestamp, dc.description;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[32]:

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).

In [33]:
```
%%sql

CREATE OR REPLACE FUNCTION count_no_shows(
    gd CHAR(1) , year VARCHAR(4), lower_age INTEGER, upper_age INTEGER)
RETURNS INTEGER AS
$$
DECLARE total_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO total_count
    FROM appointment AS a
    JOIN client AS c ON c.VAT = a.VAT_client
    WHERE NOT EXISTS(
        SELECT cc.VAT_doctor, cc.date_timestamp
        FROM consultation AS cc
        JOIN appointment AS a2 ON cc.VAT_doctor = a2.VAT_doctor AND cc.date_timestamp = a2.date_timestamp
        WHERE TO_CHAR(a2.date_timestamp, 'YYYY') = year
            AND a2.VAT_client = a.VAT_client
            AND a2.date_timestamp = a.date_timestamp
    ) AND c.gender = gd AND TO_CHAR(a.date_timestamp, 'YYYY') = year
    AND (EXTRACT(YEAR FROM AGE(current_date, c.birth_date)) BETWEEN lower_age AND upper_age);
 RETURN total_count;
END
$$ LANGUAGE plpgsql;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[33]:

In [34]:
```
%%sql

-- SELECT * FROM count_no_shows('F', '2023', 2, 80);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[34]:

In [35]:
```
%%sql

-- SELECT c.name, c.gender, c.birth_date
-- FROM appointment AS a
-- JOIN client as c ON c.VAT = a.VAT_client
-- WHERE TO_CHAR(a.date_timestamp, 'YYYY') = '2023' AND c.gender = 'F';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[35]:

In [36]:
```
%%sql

-- SELECT c1.name, c1.gender, c1.birth_date
-- FROM consultation AS c
-- JOIN appointment AS a ON a.VAT_doctor = c.VAT_doctor AND a.date_timestamp = c.date_timestamp
-- JOIN client as c1 ON c1.VAT = a.VAT_client
-- WHERE TO_CHAR(a.date_timestamp, 'YYYY') = '2023' AND c1.gender = 'F';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[36]:

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.

In [37]:
```
%%sql

CREATE OR REPLACE FUNCTION update_salary(
    x INTEGER)
RETURNS VOID AS
$$
DECLARE MULT FLOAT;
DECLARE NUM_CONS INTEGER;
DECLARE AUX_VAT VARCHAR(20);
DECLARE cursor_employee CURSOR FOR
    SELECT VAT FROM employee;
BEGIN
    OPEN cursor_employee;
    LOOP
    FETCH cursor_employee INTO AUX_VAT;
    EXIT WHEN NOT FOUND;

    SELECT COUNT(*) INTO NUM_CONS
    FROM consultation AS c
        JOIN appointment AS a ON c.VAT_doctor = a.VAT_doctor AND c.date_timestamp = a.date_timestamp
        JOIN doctor AS d ON d.VAT = a.VAT_doctor
        JOIN employee AS e ON e.VAT = d.VAT
    WHERE EXTRACT(YEAR FROM c.date_timestamp) = EXTRACT(YEAR FROM current_date) AND e.VAT = AUX_VAT;

    IF NUM_CONS <= 100 THEN
        MULT = 1.05;
    ELSE
        MULT = 1.10;
    END IF;

    UPDATE employee AS e
    SET salary = e.salary * MULT
    WHERE x < (
        SELECT pd.years
        FROM permanent_doctor AS pd
        JOIN doctor as d ON d.VAT = pd.VAT
        WHERE e.VAT = d.VAT AND e.VAT = AUX_VAT
    ) AND e.VAT = AUX_VAT;

    END LOOP;
    CLOSE cursor_employee;

 RETURN;
END
$$ LANGUAGE plpgsql;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[37]:

In [38]:
```
%%sql
-- -- Before updates
-- SELECT e.name, e.salary, pd.years
-- FROM permanent_doctor AS pd
-- JOIN doctor AS d ON pd.VAT = d.VAT
-- JOIN employee AS e ON e.VAT = d.VAT;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[38]:

In [39]:
```
%%sql
```

```
-- SELECT * FROM update_salary(6);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[39]:

In [40]:
```
%%sql
-- -- After updates
-- -- Luis Vieira had more than 100 consultations in 2023
-- SELECT e.name, e.salary, pd.years
-- FROM permanent_doctor AS pd
-- JOIN doctor AS d ON pd.VAT = d.VAT
-- JOIN employee AS e ON e.VAT = d.VAT;
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[40]:

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.

In [41]:
```
%%sql

CREATE OR REPLACE FUNCTION check_nurse()
RETURNS TRIGGER AS
$$
DECLARE nurse_count INTEGER;
DECLARE doctor_count INTEGER;
DECLARE receptionist_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO nurse_count
    FROM nurse AS n
    WHERE n.VAT = NEW.VAT;

    SELECT COUNT(*) INTO doctor_count
    FROM doctor AS d
    WHERE d.VAT = NEW.VAT;

    SELECT COUNT(*) INTO receptionist_count
    FROM receptionist AS r
    WHERE r.VAT = NEW.VAT;

    IF (doctor_count <> 0 OR receptionist_count <> 0) THEN
        RAISE EXCEPTION 'Employee already exists';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION check_doctor()
RETURNS TRIGGER AS
$$
DECLARE nurse_count INTEGER;
DECLARE doctor_count INTEGER;
DECLARE receptionist_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO nurse_count
    FROM nurse AS n
    WHERE n.VAT = NEW.VAT;

    SELECT COUNT(*) INTO doctor_count
    FROM doctor AS d
    WHERE d.VAT = NEW.VAT;

    SELECT COUNT(*) INTO receptionist_count
    FROM receptionist AS r
    WHERE r.VAT = NEW.VAT;

    IF (nurse_count <> 0 OR receptionist_count <> 0) THEN
```

```
        RAISE EXCEPTION 'Employee already exists';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION check_receptionist()
RETURNS TRIGGER AS
$$
DECLARE nurse_count INTEGER;
DECLARE doctor_count INTEGER;
DECLARE receptionist_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO nurse_count
    FROM nurse AS n
    WHERE n.VAT = NEW.VAT;

    SELECT COUNT(*) INTO doctor_count
    FROM doctor AS d
    WHERE d.VAT = NEW.VAT;

    SELECT COUNT(*) INTO receptionist_count
    FROM receptionist AS r
    WHERE r.VAT = NEW.VAT;

    IF (nurse_count <> 0 OR doctor_count <> 0) THEN
        RAISE EXCEPTION 'Employee already exists';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION check_permanent_doctor()
RETURNS TRIGGER AS
$$
DECLARE nurse_count INTEGER;
DECLARE doctor_count INTEGER;
DECLARE receptionist_count INTEGER;
DECLARE permanent_count INTEGER;
DECLARE trainee_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO nurse_count
    FROM nurse AS n
    WHERE n.VAT = NEW.VAT;

    SELECT COUNT(*) INTO doctor_count
    FROM doctor AS d
    WHERE d.VAT = NEW.VAT;

    SELECT COUNT(*) INTO receptionist_count
    FROM receptionist AS r
    WHERE r.VAT = NEW.VAT;

    IF (nurse_count <> 0 OR receptionist_count <> 0) THEN
        RAISE EXCEPTION 'Employee already exists';
    END IF;

    SELECT COUNT(*) INTO permanent_count
    FROM permanent_doctor AS pd
    WHERE pd.VAT = NEW.VAT;

    SELECT COUNT(*) INTO trainee_count
    FROM trainee_doctor AS td
    WHERE td.VAT = NEW.VAT;

    IF (trainee_count <> 0) THEN
        RAISE EXCEPTION 'Employee already exists';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION check_trainee_doctor()
RETURNS TRIGGER AS
$$
DECLARE nurse_count INTEGER;
DECLARE doctor_count INTEGER;
DECLARE receptionist_count INTEGER;
DECLARE permanent_count INTEGER;
DECLARE trainee_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO nurse_count
    FROM nurse AS n
    WHERE n.VAT = NEW.VAT;

    SELECT COUNT(*) INTO doctor_count
    FROM doctor AS d
    WHERE d.VAT = NEW.VAT;

    SELECT COUNT(*) INTO receptionist_count
    FROM receptionist AS r
    WHERE r.VAT = NEW.VAT;

    IF (nurse_count <> 0 OR receptionist_count <> 0) THEN
        RAISE EXCEPTION 'Employee already exists';
    END IF;

    SELECT COUNT(*) INTO permanent_count
    FROM permanent_doctor AS pd
    WHERE pd.VAT = NEW.VAT;

    SELECT COUNT(*) INTO trainee_count
    FROM trainee_doctor AS td
    WHERE td.VAT = NEW.VAT;

    IF (permanent_count <> 0) THEN
        RAISE EXCEPTION 'Employee already exists';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_nurse_specialization_trigger
BEFORE INSERT OR UPDATE ON nurse
FOR EACH ROW EXECUTE PROCEDURE check_nurse();

CREATE TRIGGER check_doctor_specialization_trigger
BEFORE INSERT OR UPDATE ON doctor
FOR EACH ROW EXECUTE PROCEDURE check_doctor();

CREATE TRIGGER check_receptionist_specialization_trigger
BEFORE INSERT OR UPDATE ON receptionist
FOR EACH ROW EXECUTE PROCEDURE check_receptionist();

CREATE TRIGGER check_trainee_specialization_trigger
BEFORE INSERT OR UPDATE ON trainee_doctor
FOR EACH ROW EXECUTE PROCEDURE check_trainee_doctor();

CREATE TRIGGER check_permanent_specialization_trigger
BEFORE INSERT OR UPDATE ON permanent_doctor
FOR EACH ROW EXECUTE PROCEDURE check_permanent_doctor();
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[41]:

In [42]:
```
%%sql
-- -- Cristiano Ronaldo is already a receptionist
-- INSERT INTO doctor (VAT, specialization, biography, email)
-- VALUES
--     ('3043042703', 'Orthodontics', 'Dr. Cristiano specializes in Orthodontics.', 'CR7@gmail.com');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[42]:

In [43]:
```
%%sql
-- --David Smith is already a doctor
-- INSERT INTO doctor (VAT, specialization, biography, email)
-- VALUES
--     ('7222324262', 'Endodontics', 'Dr. Smith is passionate about children health and
--         has been practicing pediatrics for over a decade.', 'smith@gmail.com');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[43]:

In [44]:
```
%%sql
-- INSERT INTO employee (VAT, name, birth_date, street, city, zip, IBAN, salary)
-- VALUES
--     ('9122222221', 'Marcelo Sweettooth', '1992-08-20', '789 Oak St', 'Chicago', '60601', '177657889', 10000.00);

-- INSERT INTO doctor (VAT, specialization, biography, email)
-- VALUES
--     ('9122222221', 'Orthodontics', 'Dr. Marcelo specializes in Orthodontics.', 'Marcelo@gmail.com');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[44]:

In [45]:
```
%%sql
-- SELECT *
-- FROM doctor
-- WHERE doctor.VAT = '9122222221';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[45]:

In [46]:
```
%%sql
-- --Cristiano Ronaldo is already a receptionist
-- INSERT INTO nurse (VAT)
-- VALUES ('3043042703');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[46]:

In [47]:
```
%%sql
-- --Tate McRae is already a nurse
-- INSERT INTO nurse (VAT)
-- VALUES ('2042462003');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[47]:

In [48]:
```
%%sql
-- INSERT INTO employee (VAT, name, birth_date, street, city, zip, IBAN, salary)
-- VALUES
--     ('9323232321', 'Anthony Coast', '1992-08-20', '789 Oak St', 'Chicago', '60601', '133337883', 10000.00);

-- INSERT INTO nurse (VAT)
-- VALUES
--     ('9323232321');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[48]:

In [49]:
```
%%sql
-- SELECT *
-- FROM nurse
-- WHERE nurse.VAT = '9323232321';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[49]:

In [50]:
```
%%sql
-- --Cristiano Ronaldo is already a receptionist
-- INSERT INTO receptionist (VAT)
-- VALUES
--     ('3043042703');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[50]:

In [51]:
```
%%sql
-- --David Smith is already a doctor
-- INSERT INTO receptionist (VAT)
-- VALUES
--     ('7222324262');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[51]:

In [52]:
```
%%sql
-- INSERT INTO employee (VAT, name, birth_date, street, city, zip, IBAN, salary)
-- VALUES
--     ('5323232321', 'Miguel Silva', '1992-08-20', '789 Oak St', 'Chicago', '60601', '155835853', 1000.00);

-- INSERT INTO receptionist (VAT)
-- VALUES
--     ('5323232321');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[52]:

In [53]:
```
%%sql
-- SELECT *
-- FROM receptionist
-- WHERE receptionist.VAT = '5323232321';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[53]:

In [54]:
```
%%sql

-- UPDATE doctor SET specialization = 'Orthodontics' WHERE VAT = '2032062803';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[54]:

In [55]:
```
%%sql

-- SELECT * FROM doctor WHERE VAT = '2032062803';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[55]:

In [56]:
```
%%sql
-- -- Dolores Aveiro is already a permanent_doctor
-- INSERT INTO trainee_doctor (VAT, supervisor)
-- VALUES
--     ('7322220202', '2032062803');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[56]:

In [57]:
```
%%sql
-- -- David Smith is already a trainee_doctor
-- INSERT INTO permanent_doctor (VAT, years)
-- VALUES
--     ('7222324262', 2);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[57]:

In [58]:
```
%%sql
-- -- Dolores Aveiro is already a permanent_doctor
-- INSERT INTO permanent_doctor (VAT, years)
-- VALUES
--     ('7322220202', 10);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[58]:

In [59]:
```
%%sql
-- -- David Smith is already a trainee_doctor
-- INSERT INTO trainee_doctor (VAT, supervisor)
-- VALUES
--     ('7222324262', '7322220202');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[59]:

In [60]:
```
%%sql

-- INSERT INTO employee (VAT, name, birth_date, street, city, zip, IBAN, salary)
-- VALUES
--     ('9142242821', 'João Maria', '1992-08-20', '789 Oak St', 'Chicago', '60601', '147454889', 10000.00);

-- INSERT INTO doctor (VAT, specialization, biography, email)
-- VALUES
--     ('9142242821', 'Orthodontics', 'Dr. João specializes in Orthodontics.', 'Joao@gmail.com');

-- INSERT INTO permanent_doctor (VAT, years)
-- VALUES
--     ('9142242821', 10);
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[60]:

In [61]:
```
%%sql
-- SELECT *
-- FROM permanent_doctor AS pd
-- WHERE pd.VAT = '9142242821';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[61]:

In [62]:
```
%%sql

-- INSERT INTO employee (VAT, name, birth_date, street, city, zip, IBAN, salary)
-- VALUES
--     ('919849821', 'João Mário', '1992-08-20', '789 Oak St', 'Chicago', '60601', '147959999', 10000.00);

-- INSERT INTO doctor (VAT, specialization, biography, email)
-- VALUES
--     ('919849821', 'Orthodontics', 'Dr. João specializes in Orthodontics.', 'Joao1@gmail.com');

-- INSERT INTO trainee_doctor (VAT, supervisor)
-- VALUES
--     ('919849821', '7322220202');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[62]:

In [63]:
```
%%sql
-- SELECT *
-- FROM trainee_doctor AS pd
-- WHERE pd.VAT = '919849821';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[63]:

In [64]:
```
%%sql

-- UPDATE permanent_doctor SET years = 3
-- WHERE VAT = '7322220202';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[64]:

In [65]:
```
%%sql
-- SELECT *
-- FROM permanent_doctor AS pd
-- WHERE pd.VAT = '7322220202';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[65]:

In [66]:
```
%%sql
-- UPDATE trainee_doctor SET supervisor = '0135012541'
-- WHERE VAT = '7222324262';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[66]:

In [67]:
```
%%sql
-- SELECT *
-- FROM trainee_doctor AS td
-- WHERE td.VAT = '7222324262';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[67]:

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

In [68]:
```
%%sql

CREATE OR REPLACE FUNCTION check_phones()
RETURNS TRIGGER AS
$$
DECLARE phone_client_count INTEGER;
DECLARE phone_employee_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO phone_client_count
    FROM phone_number_client AS pnc
    WHERE pnc.phone = NEW.phone AND pnc.VAT <> NEW.VAT;

    SELECT COUNT(*) INTO phone_employee_count
    FROM phone_number_employee AS pne
    WHERE pne.phone = NEW.phone AND pne.VAT <> NEW.VAT;

    IF phone_employee_count <> 0 OR phone_client_count <> 0 THEN
        RAISE EXCEPTION 'Phone Number already exists';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;


CREATE TRIGGER check_phone_number_client_trigger
BEFORE INSERT OR UPDATE ON phone_number_client
FOR EACH ROW EXECUTE PROCEDURE check_phones();

CREATE TRIGGER check_phone_number_employee_trigger
BEFORE INSERT OR UPDATE ON phone_number_employee
FOR EACH ROW EXECUTE PROCEDURE check_phones();
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

Out[68]:

In [69]:
```
%%sql
-- -- phone number is already in phone_number_employee
-- INSERT INTO phone_number_employee (VAT, phone)
-- VALUES
--      ('7222324262', '972223242');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[69]:
```

```
In [70]: %%sql
-- -- phone number is already in phone_number_client and VAT not present in employees
-- INSERT INTO phone_number_employee (VAT, phone)
-- VALUES
--     ('7423225472', '972223242');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[70]:
```

```
In [71]: %%sql
-- -- phone number already in phone_number_employee and VAT not present in clients
-- INSERT INTO phone_number_client (VAT, phone)
-- VALUES
--     ('123456789', '972223242');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[71]:
```

```
In [72]: %%sql
-- -- phone number already in phone_number_client
-- INSERT INTO phone_number_client (VAT, phone)
-- VALUES
--     ('123456789', '912345678');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[72]:
```

```
In [73]: %%sql

-- INSERT INTO phone_number_client (VAT, phone)
-- VALUES
--     ('123456789', '912355558');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[73]:
```

```
In [74]: %%sql

-- INSERT INTO phone_number_employee (VAT, phone)
-- VALUES
--     ('7222324262', '973233242');
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[74]:
```

```
In [75]: %%sql

-- UPDATE phone_number_client SET phone = '910216227'
-- WHERE VAT = '123456789' AND phone = '912345678';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[75]:
```

```
In [76]: %%sql

-- SELECT *
-- FROM phone_number_client
-- WHERE VAT = '123456789' AND phone = '910216227';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[76]:
```

```
In [77]: %%sql

-- UPDATE phone_number_employee SET phone = '917217227'
-- WHERE VAT = '7222324262' AND phone = '972223242';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[77]:
```

```
In [78]: %%sql

-- SELECT *
-- FROM phone_number_employee
-- WHERE VAT = '7222324262' AND phone = '917217227';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[78]:
```

```
In [79]: %%sql
-- --3083334733 is both client and doctor. This phone number is already in phone_number_employee
-- INSERT INTO phone_number_client (VAT, phone)
-- VALUES
--     ('3083334733', '929292929');

-- SELECT *
-- FROM phone_number_client
-- WHERE VAT = '3083334733' AND phone = '929292929';
```

Running query in 'postgresql+psycopg://db:***@postgres/db'

```
Out[79]:
```