Instituto Superior Técnico

Master's Degree in Biomedical Engineering

Information Systems and Databases

ASSIGNMENT 2
DATABASE MODELING

Group 50:
83805 - Afonso Raposo
83849 - Miguel Froes
83854 - Ricardo Brito

Professors:
Bruno Martins
Francisco Regateiro
João Marques

November, 2019

Conteúdo

1 Create the Database

The first step in the SQL script that creates the database is to drop all the previous tables that interfere with the ones being created. This step must be done carefully, since a lot of the tables possess `FOREIGN KEY`s relating to other tables, therefore, the tables must be dropped in a reverse order from their creation, *i.e.*, the last table to be created is the first to be dropped.

```
DROP TABLE IF EXISTS procedure_charting;
DROP TABLE IF EXISTS teeth
DROP TABLE IF EXISTS procedure_in_radiology;
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 supervision_report;
DROP TABLE IF EXISTS trainee_doctor;
DROP TABLE IF EXISTS permanent_doctor;
DROP TABLE IF EXISTS phone_number_client;
DROP TABLE IF EXISTS client;
DROP TABLE IF EXISTS nurse;
DROP TABLE IF EXISTS doctor;
DROP TABLE IF EXISTS receptionist;
DROP TABLE IF EXISTS phone_number_employee;
DROP TABLE IF EXISTS employee;
```

1.1 employee

```
CREATE TABLE employee
(
    VAT CHAR(10),
    name VARCHAR(64) NOT NULL,
    birth_date DATE NOT NULL,
    street VARCHAR(32) NOT NULL,
    city VARCHAR(32) NOT NULL,
    zip CHAR(8) NOT NULL,
    IBAN CHAR(25) NOT NULL,
    salary NUMERIC(8, 2) NOT NULL,
    PRIMARY KEY(VAT),
```

```
    UNIQUE(IBAN),
    CHECK(salary>=0)
);
```

Every VAT has a fixed size of 10 digits, therefore, they are stored as a string with fixed size, `CHAR`, and correspond to the `PRIMARY KEY`, which means they are `UNIQUE` and `NOT NULL`. The IBAN has a fixed size of 25, therefore, `CHAR(25)`, and it is also `NOT NULL` and `UNIQUE`, since the IBAN is specific to each person. The salary is a `NUMERIC` value with 8 digits total and 2 digits to the right of the decimal point, this means that an employee as a maximum salary of 999.999,99€, and since it must be positive, the `CHECK` statement is included.

1.2 phone_number_employee

```
CREATE TABLE phone_number_employee
(
    VAT CHAR(10) NOT NULL,
    phone CHAR(9),
    PRIMARY KEY(phone),
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE ON
       UPDATE CASCADE
);
```

Each phone number is `UNIQUE`, therefore, it is used as the `PRIMARY KEY` of this table. Each row will relate to one employee, presented as a `FOREIGN KEY` which `REFERENCES` the employee table. `ON DELETE` of a row on the employee table, every phone number corresponding to that employee should be deleted. `ON UPDATE` of a employee's VAT, that value is also updated in the phone_number_employee table.

1.3 receptionist

```
CREATE TABLE receptionist
(
    VAT CHAR(10) NOT NULL,
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE ON
       UPDATE CASCADE,
    UNIQUE(VAT)
);
```

This table contains all the employees that specialize as receptionist, therefore, the VAT should be `UNIQUE`, since every there is not two receptionists with the same VAT. Then the only column of this table is the VAT of these employees. Like previously, the `ON DELETE` and `ON UPDATE` should `CASCADE`.

1.4 doctor

```
CREATE TABLE doctor
(
    VAT CHAR(10) NOT NULL,
    specialization VARCHAR(32) NOT NULL,
    biography VARCHAR(512) NOT NULL,
    email VARCHAR(64) NOT NULL,
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE ON
       UPDATE CASCADE,
    UNIQUE(VAT),
    UNIQUE(email)
);
```

This is also a specialization of the employees, so the column VAT is a `FOREIGN KEY` that `REFERENCES` the VAT column on the employees table. The VAT is `UNIQUE` since this is a specialization.

Other columns are stored in this table and the email is considered to be `UNIQUE` to each row.

1.5 nurse

```
CREATE TABLE nurse
(
    VAT CHAR(10) NOT NULL,
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE ON
       UPDATE CASCADE,
    UNIQUE(VAT)
);
```

This table is similar to the receptionist one.

1.6 client

```
CREATE TABLE client
(
    VAT CHAR(10),
    name VARCHAR(64) NOT NULL,
    birth_date DATE NOT NULL,
    street VARCHAR(32) NOT NULL,
    city VARCHAR(32) NOT NULL,
    zip CHAR(8) NOT NULL,
    gender VARCHAR(6) NOT NULL,
```

```
    age INTEGER(3) NOT NULL,
    PRIMARY KEY(VAT),
    CHECK(age BETWEEN 0 AND 150)
);
```

This table is similar to the employee one, but some columns differ. In this one the gender and age are stored, but the IBAN and salary are not. The age is an `INTEGER(3)` and the `CHECK` statement assures that it must range from 0 to 150.

1.7 phone_number_client

```
CREATE TABLE phone_number_client
(
    VAT CHAR(10) NOT NULL,
    phone CHAR(9),
    PRIMARY KEY(phone),
    FOREIGN KEY(VAT) REFERENCES client(VAT) ON DELETE CASCADE ON
       UPDATE CASCADE
);
```

This table is equivalent to the phone_number_employee, but it `REFERENCES` the client table.

1.8 permanent_doctor

```
CREATE TABLE permanent_doctor
(
    VAT CHAR(10) NOT NULL,
    years INTEGER(3) NOT NULL,
    FOREIGN KEY(VAT) REFERENCES doctor(VAT) ON DELETE CASCADE ON
       UPDATE CASCADE,
    UNIQUE(VAT),
    CHECK(years BETWEEN 0 AND 100)
);
```

This is a specialization of doctor. Beside the VAT column that `REFERENCES` the doctor table, it has a column which stores the number of years of practice, which is an `INTEGER` between 0 and 100.

1.9 trainee_doctor

```
CREATE TABLE trainee_doctor
(
    VAT CHAR(10) NOT NULL,
    supervisor CHAR(10) NOT NULL,
    FOREIGN KEY(VAT) REFERENCES doctor(VAT) ON DELETE CASCADE ON
       UPDATE CASCADE,
    FOREIGN KEY(supervisor) REFERENCES permanent_doctor(VAT) ON
       DELETE CASCADE ON UPDATE CASCADE,
    UNIQUE(VAT)
);
```

It is a specialization of doctor. It also REFERENCES a row in the permanent_doctor table which corresponds to the supervisor of the trainee doctor.

1.10 supervision_report

```
CREATE TABLE supervision_report
(
    VAT CHAR(10) NOT NULL,
    date_timestamp DATETIME NOT NULL,
    description VARCHAR(512) NOT NULL,
    evaluation INTEGER(1) NOT NULL,
    PRIMARY KEY(date_timestamp),
    FOREIGN KEY(VAT) REFERENCES trainee_doctor(VAT) ON DELETE CASCADE
        ON UPDATE CASCADE,
    CHECK(evaluation BETWEEN 1 AND 5)
);
```

Every report has as PRIMARY KEY a DATETIME value, which corresponds to a string that contains a date and time. The DATETIME type was chosen over TIMESTAMP since the database does not lead with different time zones. Also, every report REFERENCES a trainee doctor and it is assured that the evaluation score is contained BETWEEN 1 AND 5.

1.11 appointment

```
CREATE TABLE appointment
(
    VAT_doctor CHAR(10),
    date_timestamp DATETIME,
    description VARCHAR(255) NOT NULL,
    VAT_client CHAR(10) NOT NULL,
    PRIMARY KEY(VAT_doctor, date_timestamp),
    FOREIGN KEY(VAT_doctor) REFERENCES doctor(VAT) ON DELETE CASCADE
       ON UPDATE CASCADE,
```

```
    FOREIGN KEY(VAT_client) REFERENCES client(VAT) ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

The PRIMARY KEY in this table will be the pair VAT_doctor and date_timestamp, since there cannot be two appointments with the same doctor at the same time.

1.12 consultation

```
CREATE TABLE consultation
(
    VAT_doctor CHAR(10),
    date_timestamp DATETIME NOT NULL,
    SOAP_S VARCHAR(512),
    SOAP_O VARCHAR(512),
    SOAP_A VARCHAR(512),
    SOAP_P VARCHAR(512),
    FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES appointment(
        VAT_doctor, date_timestamp) ON DELETE CASCADE ON UPDATE
        CASCADE,
    UNIQUE(VAT_doctor, date_timestamp)
);
```

Each consultation REFERENCES only one appointment, therefore, the pair VAT_doctor and date_timestamp of each consultation is UNIQUE, since it is a PRIMARY KEY in the table this pair REFERENCES.

1.13 consultation_assistant

```
CREATE TABLE consultation_assistant
(
    VAT_doctor CHAR(10),
    date_timestamp DATETIME NOT NULL,
    VAT_nurse CHAR(10) NOT NULL,
    FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES consultation(
        VAT_doctor, date_timestamp) ON DELETE CASCADE ON UPDATE
        CASCADE,
    FOREIGN KEY(VAT_nurse) REFERENCES nurse(VAT) ON DELETE CASCADE ON
         UPDATE CASCADE,
    UNIQUE(VAT_doctor, date_timestamp, VAT_nurse)
);
```

Again, since the same nurse cannot assist in two appointments at the same time, the group VAT_doctor, date_timestamp, VAT_nurse is UNIQUE.

1.14 diagnostic_code

```
CREATE TABLE diagnostic_code
(
    ID VARCHAR(7),
    description VARCHAR(255) NOT NULL,
    PRIMARY KEY(ID)
);
```

1.15 diagnostic_code_relation

```
CREATE TABLE diagnostic_code_relation
(
    ID1 VARCHAR(7) NOT NULL,
    ID2 VARCHAR(7) NOT NULL,
    type VARCHAR(64) NOT NULL,
    FOREIGN KEY(ID1) REFERENCES diagnostic_code(ID) ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY(ID2) REFERENCES diagnostic_code(ID) ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

This table reflects the relation between two diagnostic codes, therefore, each ID will be a `FOREIGN KEY` that `REFERENCES` the diagnostic_code table.

One could assume that the relation between two diagnostic codes is `UNIQUE`, but it was assumed that two diagnostic codes can have more than one relation between them.

1.16 consultation_diagnostic

```
CREATE TABLE consultation_diagnostic
(
    VAT_doctor CHAR(10),
    date_timestamp DATETIME NOT NULL,
    ID VARCHAR(7) NOT NULL,
    FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES consultation(
       VAT_doctor, date_timestamp) ON DELETE CASCADE ON UPDATE
       CASCADE,
    FOREIGN KEY(ID) REFERENCES diagnostic_code(ID) ON DELETE CASCADE
       ON UPDATE CASCADE,
    UNIQUE(VAT_doctor, date_timestamp, ID)
);
```

In each consultation there can be more than one diagnostic code, therefore, the ID parameter is not UNIQUE, but the same ID cannot appear twice in the same consultation, therefore, the group VAT_doctor, date_timestamp, ID is UNIQUE.

1.17 medication

```
CREATE TABLE medication
(
    name VARCHAR(64),
    lab VARCHAR(64),
    PRIMARY KEY(name, lab)
);
```

1.18 prescription

```
CREATE TABLE prescription
(
    name VARCHAR(64) NOT NULL,
    lab VARCHAR(64) NOT NULL,
    VAT_doctor CHAR(10),
    date_timestamp DATETIME NOT NULL,
    ID VARCHAR(7) NOT NULL,
    dosage VARCHAR(32) NOT NULL,
    description VARCHAR(64) NOT NULL,
    FOREIGN KEY(name, lab) REFERENCES medication(name, lab) ON DELETE
        CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(VAT_doctor, date_timestamp, ID) REFERENCES
       consultation_diagnostic(VAT_doctor, date_timestamp, ID) ON
       DELETE CASCADE ON UPDATE CASCADE,
    UNIQUE(name, lab, VAT_doctor, date_timestamp, ID)
);
```

The prescription table REFERENCES a diagnostic of a consultation and also a medication. It includes the dosage and the description of that medication intake. Each medication can only be prescribed once per consultation diagnostic, therefore, the group name, lab, VAT_doctor, date_timestamp, ID is UNIQUE.

1.19 procedure

```
CREATE TABLE procedure_
(
```

```
    name VARCHAR(64),
    type VARCHAR(64) NOT NULL,
    PRIMARY KEY(name)
);
```

The name for this table is procedure_ since `CREATE PROCEDURE` conflicts with the SQL syntax.

1.20 procedure_in_consultation

```
CREATE TABLE procedure_in_consultation
(
    name VARCHAR(64) NOT NULL,
    VAT_doctor CHAR(10),
    date_timestamp DATETIME NOT NULL,
    description VARCHAR(255) NOT NULL,
    FOREIGN KEY(name) REFERENCES procedure_(name) ON DELETE CASCADE
       ON UPDATE CASCADE,
    FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES consultation(
       VAT_doctor, date_timestamp) ON DELETE CASCADE ON UPDATE
       CASCADE,
    UNIQUE(name, VAT_doctor, date_timestamp)
);
```

In each consultation the same procedure can be performed only one, therefore, the group name VAT_doctor and date_timestamp should be `UNIQUE`.

1.21 procedure_in_radiology

```
CREATE TABLE procedure_in_radiology
(
    name VARCHAR(64) NOT NULL,
    file VARCHAR(255),
    VAT_doctor CHAR(10),
    date_timestamp DATETIME NOT NULL,
    PRIMARY KEY(file),
    FOREIGN KEY(name) REFERENCES procedure_in_consultation(name) ON
       DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES
       procedure_in_consultation(VAT_doctor, date_timestamp) ON
       DELETE CASCADE ON UPDATE CASCADE,
    UNIQUE(name, VAT_doctor, date_timestamp)
);
```

Since in each row in the procedure_in_radiology table corresponds to only one procedure in consultation, like previously, each group name VAT_doctor and date_timestamp should be `UNIQUE`.

1.22 teeth

```
CREATE TABLE teeth
(
    quadrant VARCHAR(11),
    number INTEGER(1),
    name VARCHAR(32) NOT NULL,
    PRIMARY KEY(quadrant, number),
    CHECK(number BETWEEN 1 AND 8)
);
```

According to the Palmer Notation Numbering System, the maximum number of teeth per quadrant is 8, therefore, an `INTEGER(1)` is used and it is assured that it ranges between 1 and 8.

1.23 procedure_charting

```
CREATE TABLE procedure_charting
(
    name VARCHAR(64) NOT NULL,
    VAT CHAR(10),
    date_timestamp DATETIME NOT NULL,
    quadrant VARCHAR(11) NOT NULL,
    number INTEGER(1) NOT NULL,
    description VARCHAR(255) NOT NULL,
    measure NUMERIC(3,1) NOT NULL,
    FOREIGN KEY(name) REFERENCES procedure_in_consultation(name) ON
       DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(VAT, date_timestamp) REFERENCES
       procedure_in_consultation(VAT_doctor, date_timestamp) ON
       DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(quadrant, number) REFERENCES teeth(quadrant, number)
       ON DELETE CASCADE ON UPDATE CASCADE,
    UNIQUE(name, VAT, date_timestamp, quadrant, number),
    CHECK(measure>=0)
);
```

The measure value is a `NUMERIC VALUE` of 3 digits total and 1 digit to the right of the decimal point. It is assured that it is positive, therefore, it ranges between 0.0 and 99,9.

Every tooth is measured only once per charting procedure, therefore, the combination name, VAT (VAT_doctor), date_timestamp, quadrant and number must be `UNIQUE`.

2 Populate the Database

Our database was populated having in mind the capability to assess the well-functioning of all the queries and updates used throughout this project. Having said that, the database was populated as following:

- 101 Clients
- 30 Employees:
  - 10 Doctors:
    - 5 Permanent doctors
    - 5 Trainee doctors
  - 5 Receptionists
  - 5 Nurses
- 1 unique phone number for each client and employee.
- 5 supervision reports, one for each trainee doctor.
- 353 scheduled appointments.
- 300 performed consultations.
- For each consultation either 1 or 2 nurses where assigned.
- 10 diagnostic codes.
- 2 relations were established between diagnostic codes.
- For each consultation 1 diagnostic code was assigned.
- 6 medications.
- To 9 diagnostic codes, a prescription was made. Each prescription has a 1 medication associated. Different diagnostic codes can be treated with the same medication. Also, some medications are used to treated the same conditions.
- 6 types of procedures were created:

  - Tooth extractions:
    * Extraction of impacted tooth
    * Extraction of extra teeth
  - Application of dental restoratives:
    * Treating enamel erosion
    * Treating bacterial infection
  - Radiography exams:
    * Maxillary molar periapical radiography
    * Mandible X-ray
    * Wisdom tooth X-ray
  - Dental Charting:
    * Periodontal charting
  - Root canal treatments:
    * Dental filling
    * Treatment of cavities
  - Orthodontics:
    * Dental braces

- For each consultation 1 procedure was assigned.
- 62 of the performed procedures where of the type *Radiography exams*.
- 32 of the performed procedures where of the type *Dental Charting*.

3 SQL Queries

The SQL queries developed to solve the given information needs is presented and explained below.

3.1 Query 1

```
SELECT DISTINCT client.VAT AS "Client VAT", client.name AS "Name",
    phone_number_client.phone AS "Phone"
FROM client, phone_number_client, appointment, consultation, employee
WHERE appointment.date_timestamp = consultation.date_timestamp
    AND employee.name = 'Jane Sweettooth'
    AND employee.VAT = appointment.VAT_doctor
```

```
    AND client.VAT = appointment.VAT_client
    AND phone_number_client.VAT = client.VAT;
ORDER BY client.name ASC;
```

In this query, are listed the VAT, name and phone number of all clients that had consultations with the doctor named *Jane Sweettooth*.

First, it is assured that the appointment became a consultation by equaling their timestamp. This means the patient showed up. Then, the name of the employee of that consultation is specified, *Jane Sweettooth*, the VAT of the employee of the consultation is the VAT of the doctor of the appointment, the VAT of the client is equal to both the VAT of the client of the appointment and the VAT associated with the client's phone number.

3.2 Query 2

```
SELECT etd.name AS "Trainee Dr Name", td.VAT AS "Trainee Dr VAT", epd
    .name AS "Permanent Dr Name", supervision_report.evaluation AS "
    Evaluation Score", supervision_report.description AS "Report
    Description"
FROM (employee etd INNER JOIN trainee_doctor td ON etd.VAT = td.VAT)
    ,
    (employee epd INNER JOIN permanent_doctor pd ON epd.VAT = pd.VAT)
        ,
    supervision_report
WHERE (supervision_report.evaluation < 3 OR LOWER(supervision_report.
    description) LIKE '%insufficient%')
    AND td.VAT= supervision_report.VAT
    AND td.supervisor = pd.VAT
ORDER BY supervision_report.evaluation DESC;
```

In this query, it is listed the name and the VAT of the trainee doctor, the name of the permanent doctor, the evaluation and description of all reports that scored less than 3 or were *insufficient*.

To achieve this, a `INNER JOIN` between the employee table and trainee_doctor, and the employee table and the permanent_doctor one, were applied. These employee tables were associated with their respective entities (trainee doctor and permanent doctor) through their VAT.

This allows to match each trainee doctor and each permanent doctor with its corresponding employee row.

Then, the conditions that these reports had to respect were specified: score lower than 3 or having the term *insufficient*. The trainee doctor's VAT associated with the report is matched with the td table and returns the trainee doctor. The attribute supervisor of the trainee doctor is matched with the pd table and returns the permanent doctor.

3.3 Query 3

```
SELECT client.name AS "Client Name", client.city AS "City", client.
    VAT AS "VAT"
FROM client,
    (SELECT appointment.VAT_client, consultation.SOAP_O, MAX(
        appointment.date_timestamp)
    FROM (appointment
        INNER JOIN consultation
        ON appointment.date_timestamp = consultation.date_timestamp
            AND appointment.VAT_doctor = consultation.VAT_doctor)
    GROUP BY appointment.VAT_client) AS con
WHERE (LOWER(con.SOAP_O) LIKE '%gingivitis%' OR LOWER(con.SOAP_O)
    LIKE '%periodontitis%')
    AND con.VAT_client = client.VAT;
```

In this query, it is listed the name, city and VAT of the clients who had the Objective part of their SOAP notes containing the words *gingivitis* or *periodontitis* in their most recent consultation.

To select the most recent consultation, a `INNER JOIN` was applied between the appointment table and consultation one `ON` the timestamp and VAT_doctor. Then, the results were grouped by appointment.VAT_client, and the MAX command was used in the appointment.date_timestamp column to select the biggest timestamp which corresponds to the last entry. This value, the client and the objective part of the SOAP were displayed in a temporary table treated `AS` con.

Then the rows in which the SOAP_O value contains one of the terms are matched with the client by their VAT.o the client VAT.

3.4 Query 4

```
SELECT DISTINCT client.name AS "Client Name", client.VAT AS "VAT",
    client.city AS "City", client.street AS "Street", client.zip AS "
    ZIP code"
FROM client
    INNER JOIN appointment
    ON appointment.VAT_client = client.VAT
WHERE VAT_client NOT IN (SELECT DISTINCT appointment.VAT_client
FROM appointment
    INNER JOIN consultation
    ON appointment.date_timestamp = consultation.date_timestamp
        AND appointment.VAT_doctor = consultation.VAT_doctor);
```

In this query, its is listed the name, VAT, city, street and ZIP code of all clients that have had appointments but never had a consultation.

To achieve this, it is assured that the client's VAT is not present in the appointments that turn into consultations. This is accomplished by creating a temporary table in which it is selected the client's VAT of all clients that associated with appointments that turn into consultations and select the clients' VAT that are `NOT IN` this temporary table.

3.5 Query 5

```
SELECT diagnostic_code.ID AS "Diagnostic Code", diagnostic_code.
    description AS "Description", COUNT(DISTINCT medication.name) AS "
    Number of distinct medications"
FROM diagnostic_code
    LEFT JOIN (prescription
    INNER JOIN medication
    ON (prescription.name = medication.name
        AND prescription.lab = medication.lab))
    ON diagnostic_code.ID = prescription.ID
GROUP BY(diagnostic_code.ID)
ORDER BY COUNT(DISTINCT medication.name) ASC;
```

In this query, it is list the diagnostic code ID, description and number of different medications used to treat a condition.

First, a `INNER JOIN` is applied between the prescription table and the medication one, in order to get the corresponding medication information of each prescription. Then a `LEFT JOIN` is applied between the diagnostic_code table and the `INNER JOIN` result, in order to correspond each diagnostic code to the prescriptions associated to it. If there is not any prescription, the value presented is `NULL`.

Then the `GROUP BY` command is applied to group the rows by diagnostic_code.ID and the `COUNT` is used to count the number of `DISTINCT` medication.name in each group. The results are ordered in ascending order by the this value.

3.6 Query 6

```
SELECT Age AS "0: <=18 | 1: >18", AVG(temp.count_nurses) AS "Average
    Nurses", AVG(temp.count_procedures) AS "Average Procedures", AVG(
    temp.count_diagnostics) AS "Average Diagnostics", AVG(temp.
    count_prescriptions) AS "Average Prescriptions"
FROM
    (SELECT client.age>18 AS Age, COUNT(DISTINCT
        consultation_assistant.VAT_nurse) AS count_nurses, COUNT(
        DISTINCT procedure_in_consultation.name) AS count_procedures,
        COUNT(DISTINCT consultation_diagnostic.ID) AS
```

```
        count_diagnostics, COUNT(DISTINCT prescription.name,
        prescription.lab) AS count_prescriptions
    FROM client
        INNER JOIN appointment
        ON (client.VAT = appointment.VAT_client)
        INNER JOIN consultation
        ON (appointment.VAT_doctor = consultation.VAT_doctor
            AND appointment.date_timestamp = consultation.
                date_timestamp)
        LEFT JOIN consultation_assistant
        ON (consultation.VAT_doctor = consultation_assistant.
            VAT_doctor
            AND consultation.date_timestamp = consultation_assistant.
                date_timestamp)
        LEFT JOIN procedure_in_consultation
        ON (consultation.VAT_doctor = procedure_in_consultation.
            VAT_doctor
            AND consultation.date_timestamp =
                procedure_in_consultation.date_timestamp)
        LEFT JOIN consultation_diagnostic
        ON (consultation.VAT_doctor = consultation_diagnostic.
            VAT_doctor
            AND consultation.date_timestamp = consultation_diagnostic
                .date_timestamp)
        LEFT JOIN prescription
        ON (consultation.VAT_doctor = prescription.VAT_doctor
            AND consultation.date_timestamp = prescription.
                date_timestamp)
    WHERE YEAR(consultation.date_timestamp) = 2019
    GROUP BY consultation.VAT_doctor, consultation.date_timestamp) AS
        temp
GROUP BY Age;
```

In this query, the clients of 2019 are divided into 2 age groups, less than or equal to 18 and greater than 18. For each group the average number of nurses, procedures, diagnostic codes and prescriptions involved in consultations is listed.

A `INNER JOIN` is applied between the appointment and consultation tables in order to match the information in each one of this tables. Then a `LEFT JOIN` is applied between this and the consultation_assistant, procedure_in_consultation, consultation_diagnostic, and prescription tables. This assures a correspondence between each consultation and the other tables if it exists; if not, the `NULL` value is presented. Only the rows corresponding to 2019 are selected. A `GROUP BY` consultation.VAT_doctor and consultation.date_timestamp is applied to these rows, which results in each row corresponding to one consultation and each one containing all the information regarding the nurses, procedures, diagnostic codes, and prescriptions involved. To count the number of each, a `COUNT(DISTINCT [COLUMN])` was applied to: consultation_assistant.VAT_nurse, procedure_in_consultation.name, consulta-

tion_diagnostic.ID, and pair prescription.name and prescription.lab.

In order to divide the results of each consultation between the two groups, the client information corresponding to each consultation is matched and the `SELECT` statement contains: client.age¿18 `AS` Age. Therefore, for each consultation row, the value of this column is 0 if the consultation corresponds to a client with age less than or equal to 18, and 1 otherwise.

Then, this temporary table obtained is grouped by this Age, and the average of each column is calculated using `AVG`, resulting in the desired table.

3.7 Query 7

```
SELECT temp.ID AS "Diagnostic Code", temp.name "Most used medication"
FROM
    (SELECT diagnostic_code.ID, medication.name
    FROM diagnostic_code
        LEFT JOIN (prescription
        INNER JOIN medication
        ON (prescription.name = medication.name
            AND prescription.lab = medication.lab))
        ON diagnostic_code.ID = prescription.ID
    GROUP BY diagnostic_code.ID, medication.name
    ORDER BY COUNT(medication.name) DESC) AS temp
GROUP BY temp.ID;
```

In this query, the name of the most common medication used to treat each diagnostic code is listed.

Similarly to the Query 5, a `INNER JOIN` is applied between prescription and medication, and `LEFT JOIN` is applied between diagnostic_code and the `INNER JOIN` result. Then, a `GROUP BY` is applied that groups rows with the same diagnostic_code.ID and medication.name. By applying `COUNT(MEDICATION.NAME)`, the number of times each medication was prescribed for each diagnostic code is obtained. Therefore, by sorting by `COUNT(MEDICATION.NAME)` in a descending order the medications used more often will appear first. There might still be multiple medications associated to each diagnostic code, then, by applying a `GROUP BY(TEMP.ID)` only the first occurrence of each diagnostic_code.ID is displayed, which corresponds to the most common medication prescribed to it.

3.8 Query 8

```
SELECT DISTINCT prescription.name AS "Medication Name", prescription.
    lab AS "Lab"
```

```
FROM prescription, diagnostic_code
WHERE prescription.ID = diagnostic_code.ID
    AND diagnostic_code.description LIKE '%dental cavities%'
    AND diagnostic_code.description NOT LIKE '%infectious disease%'
    AND YEAR(prescription.date_timestamp) = 2019
ORDER BY prescription.name;
```

In this query, it is listed the name and laboratory of medications that were used to treat *dental cavities* but not *infectious diseases* in 2019.

The tables prescription and diagnostic code were used to achieve this goal, since each prescription relates a medication and a diagnostic code. Then, only the rows that verify the conditions mentioned above were selected.

3.9 Query 9

```
SELECT DISTINCT client.name AS "Client Name", client.city AS "City",
   client.street AS "Street", client.zip AS "ZIP code"
FROM client
    INNER JOIN appointment
    ON appointment.VAT_client = client.VAT
WHERE YEAR(appointment.date_timestamp) = 2019
    AND VAT_client IN (SELECT DISTINCT appointment.VAT_client
    FROM appointment
        INNER JOIN consultation
        ON appointment.date_timestamp = consultation.date_timestamp
            AND appointment.VAT_doctor = consultation.VAT_doctor);
```

This query is the opposite of the query 4. It is listed the name, city, street and zip of all clients that never missed an appointment (all appointments turned into consultations) in 2019.

Everything is the same for this query, except that it is assured that the YEAR of the appointment is 2019 and instead it is selected the clients which VAT is `IN` the temporary table containing the VAT of every client that did not miss a consultation.

4 Database Indexes Suggestions

4.1 Index 1

```
CREATE INDEX indexName ON employee(name);
```

Since this query goes through the employee table searching for the doctor named *Jane Sweettooth*, the table can be indexed by the attribute name, this way the query

can find immediately the VAT corresponding to a specific doctor, in this case, *Jane Sweettooth*.

4.2 Index 2

```
CREATE INDEX indexScore ON supervision_report(evaluation);
CREATE FULLTEXT INDEX ON supervision_report(description);
```

This query goes through the supervision_report table searching for reports with a evaluation score less than 3 or the term *insufficient* in its description. Therefore, in order to speed up the first search, an `INDEX` could be created on the evaluation column, making it much faster to access evaluation reports of a specific evaluation.

Then, a `FULLTEXT INDEX` could be created `ON` the description column. If one wants to apply this `INDEX`, the following statement of this query must be change from:

```
LOWER(supervision_report.description) LIKE '%insufficient%')
```

To:

```
CONTAINS(supervision_report.description, "insufficient")
```

5 Changes in the Database

5.1 Change 1

```
UPDATE employee
SET city = "Santiago do Cacem", street = "Rua de Mirobriga", zip = "
    7540-109"
WHERE name = "Jane Sweettooth"
```

5.2 Change 2

```
UPDATE employee,
    (SELECT VAT
    FROM employee
        INNER JOIN appointment
        ON employee.VAT = appointment.VAT_doctor
    WHERE YEAR(appointment.date_timestamp) = 2019
    GROUP BY employee.VAT
    HAVING COUNT(appointment.date_timestamp)>100) AS raise
SET salary = salary*1.05
WHERE employee.VAT = raise.VAT
```

A temporary table called raise is created in which are listed the VAT corresponding to employees that deserve a raise. The employee table is matched with the appointment one with an `INNER JOIN`, the year 2019 is selected, and a `GROUP BY` VAT is applied, this way, in each row we get a different VAT and, in it, all the appointments corresponding to it. By applying `COUNT` appointment.date_timestamp, the number of appointments is calculated and then assured that it is greater than 100 with the `HAVING` term.

5.3 Change 3

```
DELETE FROM diagnostic_code
WHERE ID IN (
    SELECT consultation_diagnostic.ID
    FROM appointment
        INNER JOIN consultation
        ON appointment.date_timestamp = consultation.date_timestamp
            AND appointment.VAT_doctor = consultation.VAT_doctor
        INNER JOIN consultation_diagnostic
        ON consultation.date_timestamp = consultation_diagnostic.
           date_timestamp
            AND consultation.VAT_doctor = consultation_diagnostic.
               VAT_doctor
        LEFT JOIN employee
        ON employee.VAT = consultation.VAT_doctor
            AND employee.name NOT LIKE "Jane Sweettooth"
GROUP BY consultation_diagnostic.ID
HAVING COUNT(DISTINCT employee.name) = 0)

DELETE FROM procedure_
WHERE name IN (
    SELECT procedure_in_consultation.name
    FROM appointment
        INNER JOIN consultation
        ON appointment.date_timestamp = consultation.date_timestamp
            AND appointment.VAT_doctor = consultation.VAT_doctor
        INNER JOIN procedure_in_consultation
        ON consultation.date_timestamp = procedure_in_consultation.
           date_timestamp
            AND consultation.VAT_doctor = procedure_in_consultation.
               VAT_doctor
        LEFT JOIN employee
        ON employee.VAT = consultation.VAT_doctor
            AND employee.name NOT LIKE "Jane Sweettooth"
GROUP BY procedure_in_consultation.name
HAVING COUNT(DISTINCT employee.VAT) = 0)

DELETE FROM employee
```

```
34 WHERE name = "Jane Sweettooth";
```

Since, when creating the table, `ON DELETE CASCADE` was used, when the employee *Jane Sweettooth* is removed, every row that contains her VAT is also deleted.

However, it is also asked to delete from database the diagnostic codes and procedures associated only to her. For example, if *Jane Sweettooth* is the only doctor able to perform a procedure, if she leaves the clinic, that procedure should be eliminated.

To do so, a temporary table containing the consultation IDs of the rows associated only to her is created. All the consultations diagnostics are gathered and a `LEFT JOIN` with the employee table was done `ON` the employee.VAT and the name `NOT LIKE` *Jane Sweettooth*, so the ones associated to the doctor *Jane Sweettooth* appear as null. The results are grouped by consultation_diagnostic.ID and a `COUNT` of `DISTINCT` employee.VAT is performed. If the only doctor associated to that diagnostic code is *Jane Sweettooth*, the `COUNT` will return 0. Therefore, the ones who return 0 are the ones selected and deleted.

The same method is applied to the procedure table.

5.4 Change 4

```
1 INSERT INTO diagnostic_code(ID,description)
2 SELECT * FROM (SELECT 'DC_11', 'Periodontitis') AS tmp
3 WHERE NOT EXISTS (
4     SELECT description FROM diagnostic_code WHERE LOWER(description)
          LIKE '%periodontitis%'
5 ) LIMIT 1;
```

Add a new row to the diagnostic_code table corresponding to *periodontitis* if it does not already exist in it.

```
1  UPDATE consultation_diagnostic,
2      (SELECT consultation_diagnostic.date_timestamp,
           consultation_diagnostic.VAT_doctor
3      FROM consultation_diagnostic
4          INNER JOIN diagnostic_code
5          ON consultation_diagnostic.ID = diagnostic_code.ID
6          INNER JOIN procedure_charting
7          ON consultation_diagnostic.date_timestamp =
               procedure_charting.date_timestamp
8              AND consultation_diagnostic.VAT_doctor =
                   procedure_charting.VAT
9      WHERE LOWER(diagnostic_code.description) LIKE '%gingivitis%'
10     GROUP BY procedure_charting.VAT, procedure_charting.
           date_timestamp
11     HAVING AVG(procedure_charting.measure)>4) AS consultations
```

```
12 SET consultation_diagnostic.ID =
13     (SELECT DISTINCT ID
14     FROM diagnostic_code
15     WHERE LOWER(description) LIKE '%periodontitis%'
16     LIMIT 1)
17 WHERE consultation_diagnostic.date_timestamp = consultations.
       date_timestamp
18     AND consultation_diagnostic.VAT_doctor = consultations.VAT_doctor
           ;
```

An `INNER JOIN` is applied between the consultation_diagnostic, diagnostic_code and procedure_charting tables. Then, only the rows containing the term *gingivitis* in the diagnostic_code.description are selected. Then, the rows are grouped by VAT_doctor and date_timestamp in order to obtain a different consultation in each row. The average gap measure for each consultation is done by applying `AVG` to procedure_charting.measure and make sure that it is greater than 4 (mm) in the `HAVING` statement.

6 Views Over the Tables

6.1 View 1

```
1 CREATE OR REPLACE VIEW dim_date AS
2 SELECT consultation.date_timestamp AS date_timestamp, YEAR(
      consultation.date_timestamp) AS "Year", MONTH(consultation.
      date_timestamp) AS "Month", DAY(consultation.date_timestamp) AS "
      Day"
3 FROM consultation;
```

This view creates a table which contains every consultation on the first column and then the year, month and day of each consultation on adjacent columns.

6.2 View 2

```
1 CREATE OR REPLACE  VIEW dim_client AS
2 SELECT client.VAT AS VAT, client.gender AS "Gender", client.age AS "
      Age"
3 FROM client;
```

This view creates a table with the VAT, gender and age of each client of our clinic.

6.3 View 3

```sql
CREATE OR REPLACE  VIEW dim_location_client AS
SELECT DISTINCT client.zip AS ZIP, client.city AS "City"
FROM client;
```

This view creates a table with the zip code and city of each client of our clinic. A `DISTINCT` was used in order to not contain repeated ZIP codes.

6.4 View 4

```sql
CREATE OR REPLACE VIEW facts_consults AS
SELECT dim_client.VAT AS "Client VAT", dim_date.date_timestamp AS "
    Date", dim_location_client.ZIP AS "ZIP", COUNT(DISTINCT
    procedure_in_consultation.name) AS "# Procedures", COUNT(DISTINCT
    prescription.name, prescription.lab) AS "# Medication", COUNT(
    DISTINCT consultation_diagnostic.ID) AS "# Diagnostic Codes"
FROM dim_client
        INNER JOIN client
        ON dim_client.VAT = client.VAT
        INNER JOIN dim_location_client
        ON client.zip = dim_location_client.zip
        INNER JOIN appointment
        ON dim_client.VAT = appointment.VAT_client
        INNER JOIN dim_date
        ON appointment.date_timestamp = dim_date.date_timestamp
        INNER JOIN consultation
        ON appointment.date_timestamp = consultation.date_timestamp
                AND appointment.VAT_doctor = consultation.VAT_doctor
        LEFT JOIN consultation_diagnostic
        ON consultation.date_timestamp = consultation_diagnostic.
           date_timestamp
                AND consultation.VAT_doctor = consultation_diagnostic
                   .VAT_doctor
        LEFT JOIN prescription
        ON consultation.date_timestamp = prescription.date_timestamp
                AND consultation.VAT_doctor = prescription.VAT_doctor
        LEFT JOIN procedure_in_consultation
        ON consultation.date_timestamp = procedure_in_consultation.
           date_timestamp
                AND consultation.VAT_doctor =
                   procedure_in_consultation.VAT_doctor
GROUP BY dim_client.VAT, appointment.date_timestamp;
```

This query creates a table with a different row for each consultation with the date of the consultation, the VAT and zip code of the client and the associated number of procedures, number of diagnostic codes, and number of prescribed medications.

A `INNER JOIN` was used to match the tables client, dim_location_client, appointment, dim_date, consultation, since there is a direct match between the rows of these

tables. Then a `LEFT JOIN` was used to match with consultation_diagnostic, prescription and procedure_in_consultation, since there might not be any corresponding match between the previously obtained tables and these ones.

Then, a `GROUP BY` dim_client.VAT and appointment.date_timestamp was used to group the information regarding each. Then a `COUNT` was applied to the `DISTINCT` values of columns procedure_in_consultation.name, pair prescription.name and prescription.lab, and consultation_diagnostic.ID.

7 Results

7.1 Queries

7.1.1 Query 1

Client VAT	Name	Phone
5071193401	Abe Casely	914674663
5375272189	Abram Rodden	916640904
5205804402	Alistair Boydon	918594218
3037318718	Andriana Bach	915168085
3054429349	Aylmer Geeson	913387296
5680263126	Barde Nicklen	916733473
1384288459	Barron Riccio	911953742
8236190259	Basil Sammon	918829933
1564926875	Bekki Houlson	911754530
5484250919	Bernardo Lynas	910306491
2261570434	Blinnie Halm	910171172
6409151986	Briano Boice	916782661
2512980241	Cherilynn Braham	913707049
1495182097	Clarey Lampert	913346849
5168802859	Claudio Di Carli	911998536
6101114711	Cobb Lehrle	918921015
1065284477	Cordy Addyman	914289428
4834682164	Cosetta Wapol	912445846
8924571573	Culver McGuire	914428166
5880863332	Delly McKinie	910787581
8632411626	Dewitt Pipping	914195712
1288938172	Didi Fossett	912822931
7593875652	Dorree Harvison	918589128
9366456197	Dru Crosio	919265438
2125409049	Elvyn Lasselle	916865928
8689053273	Emmery MacGowing	910795261
8301432130	Ericha Ilyenko	917264907
3260652693	Everard Inderwick	916106890
9309326893	Ewell Tesdale	912044860
4666833838	Florencia Haveline	918412275
5403873299	Franklin Leindecker	911633208
1443765898	Gaston May	917857497
1125478009	Gates Touson	913984615
2785024496	Gena Tadgell	916785986
3014867017	Gerianne Fairbank	910367419
5819093380	Glenn Brimman	918819997
8847783511	Gnni Erington	915995604
4192964808	Guglielma Donnachie	912767436
1315824037	Gusta Faucherand	910087890
5057692187	Harley Demare	911036263
3045325542	Helge Rickert	914108816
1251928457	Helyn Giraudat	919038772
6037486263	Holt Osipov	919263196
6324221520	Jana Poff	912699511
5153824867	Jard Goligher	912729002
9262710939	Jewel Frowen	913570788
8372266270	Jobi Louden	919932200
8237760990	Josefa Dregan	916690199
8104963722	Josy Crowest	917002258
4175670708	Joyan Heibl	917850236

```
| 9261507324 | Julieta Weblin      | 917053602 |
| 1774361024 | Kendre Kynforth     | 911175646 |
| 7416995863 | Kingston Fowlds     | 911485354 |
| 4321553298 | Konrad Forsey       | 910279719 |
| 2380246766 | Laurena Simmell     | 917882709 |
| 9916815067 | Lavinia Jurca       | 917056774 |
| 9666407194 | Libbey Dilliway     | 914560129 |
| 7710870439 | Loni Kunneke        | 913217491 |
| 7678417951 | Lorianna Leithgoe   | 913637872 |
| 7412563835 | Lotta Stoodale      | 912406184 |
| 9043514344 | Lurline Duffie      | 918710461 |
| 7807464581 | Marna Rickasse      | 910616523 |
| 8272343344 | Merle Sporton       | 915074389 |
| 2542736826 | Murdoch Semon       | 915369267 |
| 6633747219 | Nap Cockerton       | 914270294 |
| 1944111515 | Natka Jeavons       | 914398321 |
| 3333638031 | Neilla Cushworth    | 912751887 |
| 6978678991 | Nolly Morais        | 917105699 |
| 7763386444 | Obed Van den Bosch  | 916663071 |
| 3721688004 | Page Devanney       | 916628484 |
| 3594045692 | Pat Fulbrook        | 915290213 |
| 4681803112 | Purcell Dickin      | 913269713 |
| 3269972780 | Raf Gowlett         | 913628945 |
| 1889223122 | Reeta Hufton        | 917844858 |
| 3252775826 | Reinald Agron       | 913970379 |
| 7129224524 | Rockey Exall        | 919040272 |
| 2756537095 | Roldan Steinson     | 910002817 |
| 7158549937 | Ronni De Vries      | 919913328 |
| 3027642977 | Ruthann Tomek       | 915453515 |
| 7277773552 | See Feron           | 919139485 |
| 6164397671 | Sonnie Loade        | 917954364 |
| 3787344317 | Sonnie Stubbeley    | 912875852 |
| 7587186491 | Stephana Purvis     | 912307543 |
| 8443316525 | Taber Swains        | 916750141 |
| 3044991851 | Terencio Munnis     | 910284692 |
| 1347404557 | Tiena Derisley      | 919741400 |
| 6071361884 | Tobin Christensen   | 911805923 |
| 4517801011 | Tonnie Stellin      | 914740699 |
| 6382296314 | Torr Yurkevich      | 912047587 |
| 4444304768 | Tracy Cromly        | 913436510 |
| 3852397258 | Verina Lacheze      | 910505847 |
| 9795584884 | Zack Sartain        | 913743370 |
+------------+---------------------+-----------+
92 rows in set (0.04 sec)
```

7.1.2 Query 2

```
+-----------------+----------------+-------------------+------------------+---------------------+
| Trainee Dr Name | Trainee Dr VAT | Permanent Dr Name | Evaluation Score | Report Description |
+-----------------+----------------+-------------------+------------------+---------------------+
| Mike Oaxmall    | 5224001850     | Mike Ratch        |                2 | Insufficient        |
| Hugh Jass       | 7782456248     | Ben Dover         |                1 | Weak                |
+-----------------+----------------+-------------------+------------------+---------------------+
2 rows in set (0.02 sec)
```

7.1.3 Query 3

```
+------------------+----------------+------------+
| Client Name      | City           | VAT        |
+------------------+----------------+------------+
| Clarey Lampert   | Vila de Rei    | 1495182097 |
| Cherilynn Braham | Quartier Morin | 2512980241 |
| Gena Tadgell     | Sydney         | 2785024496 |
| Andriana Bach    | Kremidivka     | 3037318718 |
| Sonnie Stubbeley | Huochezhan     | 3787344317 |
| Cosetta Wapol    | Jargalant      | 4834682164 |
| Briano Boice     | Invercargill   | 6409151986 |
| Nolly Morais     | Houston        | 6978678991 |
| Basil Sammon     | Huerta Grande  | 8236190259 |
+------------------+----------------+------------+
9 rows in set (0.01 sec)
```

7.1.4 Query 4

```
+--------------+------------+-------+----------+----------+
| Client Name  | VAT        | City  | Street   | ZIP code |
+--------------+------------+-------+----------+----------+
| Carlos Lopes | 6969696969 | Porto | Maratona | 1700-136 |
+--------------+------------+-------+----------+----------+
1 row in set (0.01 sec)
```

7.1.5 Query 5

```
+-----------------+--------------------------+---------------------------------+
| Diagnostic Code | Description              | Number of distinct medications |
+-----------------+--------------------------+---------------------------------+
| DC_10           | Crooked teeth            |                               0 |
| DC_5            | Periodontitis            |                               1 |
| DC_7            | Infectious disease       |                               1 |
| DC_2            | Gingivitis               |                               1 |
| DC_6            | Impacted tooth           |                               2 |
| DC_1            | Broken tooth             |                               2 |
| DC_8            | Structural problems      |                               2 |
| DC_9            | Enamel erosion           |                               2 |
| DC_3            | Cyst due to Wisdom tooth |                               2 |
| DC_4            | Dental Cavities          |                               2 |
+-----------------+--------------------------+---------------------------------+
10 rows in set (0.02 sec)
```

7.1.6 Query 6

```
+------------------+----------------+--------------------+----------------------+------------------------+
| 0: <=18 | 1: >18 | Average Nurses | Average Procedures | Average Diagnostics | Average Prescriptions |
+------------------+----------------+--------------------+----------------------+------------------------+
|                0 |         1.6389 |             1.0000 |               1.0000 |                 0.9444 |
|                1 |         1.6288 |             1.0000 |               1.0000 |                 0.8939 |
+------------------+----------------+--------------------+----------------------+------------------------+
2 rows in set (0.04 sec)
```

7.1.7 Query 7

```
+-----------------+-----------------------+
| Diagnostic Code | Most used medication |
+-----------------+-----------------------+
| DC_1            | Tylenol               |
| DC_10           | NULL                  |
| DC_2            | PerioGard             |
| DC_3            | Tylenol               |
| DC_4            | Fluoride              |
| DC_5            | PerioGard             |
| DC_6            | Tylenol               |
| DC_7            | Metronidazole         |
| DC_8            | Calpol                |
| DC_9            | Fluoride              |
+-----------------+-----------------------+
10 rows in set (0.02 sec)
```

7.1.8 Query 8

Medication Name	Lab
Fluoride	Lilly
Hydroxyapatite	Roche

2 rows in set (0.02 sec)

7.1.9 Query 9

Client Name	City	Street	ZIP code
Cordy Addyman	Tycher	Cordelia	1700-089
Gates Touson	Jaciara	Stuart	1700-108
Helyn Giraudat	Lisbon	Lunder	1700-073
Didi Fossett	Sasar	Hermina	1700-040
Gusta Faucherand	Kamuli	Myrtle	1700-077
Tiena Derisley	Beicheng	Bellgrove	1700-125
Maribeth Pardey	Yaguaraparo	Pennsylvania	1700-056
Barron Riccio	Xiangzikou	Banding	1700-082
Gaston May	Kalmar	Dovetail	1700-041
Clarey Lampert	Vila de Rei	Oak Valley	1700-109
Bekki Houlson	Sengeti	Homewood	1700-059
Kendre Kynforth	Sung Noen	Karstens	1700-065
Reeta Hufton	Rustaq	Starling	1700-135
Natka Jeavons	Gazanjyk	Oneill	1700-081
Elvyn Lasselle	El Ocotana	Thompson	1700-058
Normie Fudge	Kidodi	Monterey	1700-055
Blinnie Halm	Nogent-sur-Marne	Lindbergh	1700-096
Laurena Simmell	Pomerode	2nd	1700-043
Cherilynn Braham	Quartier Morin	Knutson	1700-083
Murdoch Semon	Ranilug	Hanson	1700-112
Roldan Steinson	Guantang	Pepper Wood	1700-093
Gena Tadgell	Sydney	Scoville	1700-066
Gerianne Fairbank	Saffle	Goodland	1700-116
Ruthann Tomek	Hammam Sousse	Dixon	1700-062
Andriana Bach	Kremidivka	Fordem	1700-067
Terencio Munnis	Nyurba	La Follette	1700-042
Helge Rickert	San Juan de Mata	Vidon	1700-097
Aylmer Geeson	Youcheng	Ruskin	1700-120
Dalila Baxandall	Kista	Westport	1700-054
Reinald Agron	Ifon	Nelson	1700-106
Everard Inderwick	Bykov	Lawn	1700-087
Raf Gowlett	Wangkung	Blaine	1700-060
Neilla Cushworth	Zhongshan	Iowa	1700-121
Pat Fulbrook	Ocongate	Express	1700-122
Page Devanney	Vilcan	Moland	1700-075
Sonnie Stubbeley	Huochezhan	Pierstorff	1700-099
Verina Lacheze	Abelheira	Ludington	1700-064
Joyan Heibl	Guadalupe	Coolidge	1700-100
Guglielma Donnachie	Sangmalima	Jackson	1700-134
Konrad Forsey	La Roxas	Milwaukee	1700-085
Tracy Cromly	Davila	Manufacturers	1700-114
Tonnie Stellin	Wenlin	Springview	1700-123
Florencia Haveline	Risalpur	Johnson	1700-124
Purcell Dickin	Daxing	Mandrake	1700-133
Cosetta Wapol	Jargalant	Hallows	1700-131
Harley Demare	Sulaq	Linden	1700-069
Abe Casely	Levuka	Commercial	1700-104
Jard Goligher	Prakhon Chai	Mcguire	1700-091
Claudio Di Carli	Karangkancana	Jenifer	1700-098
Alistair Boydon	Kuala Lumpur	Debs	1700-079
Justine Ashwell	Heshang	Monument	1700-049
Abram Rodden	Loreto	Kensington	1700-127
Franklin Leindecker	Lazaro Cardenas	Portage	1700-063
Bernardo Lynas	Taupo	Brentwood	1700-038
Barde Nicklen	Chervonohryhorivka	Katie	1700-046
Glenn Brimman	Hecheng	Carioca	1700-119
Delly McKinie	Dashiqiao	Erie	1700-070
Holt Osipov	Palalang	Merchant	1700-117
Tobin Christensen	Ukata	Rieder	1700-047
Cobb Lehrle	Santiago Vazquez	Division	1700-111

```
| Sonnie Loade         | Kedungdoro          | Homewood      | 1700-045 |
| Jana Poff            | Changsha            | Marquette     | 1700-126 |
| Torr Yurkevich       | Fresno              | Northport     | 1700-128 |
| Briano Boice         | Invercargill        | Londonderry   | 1700-103 |
| Nap Cockerton        | Gajah               | Anzinger      | 1700-084 |
| Nolly Morais         | Houston             | Havey         | 1700-130 |
| Rockey Exall         | Manzil Zalafah      | Graceland     | 1700-090 |
| Ramsey Rickert       | Manalongon          | Anhalt        | 1700-053 |
| Ronni De Vries       | Candelaria          | Magdeline     | 1700-094 |
| See Feron            | Ejigbo              | Hovde         | 1700-068 |
| Swen Playdon         | Mombasa             | Ryan          | 1700-052 |
| Lotta Stoodale       | Laboulaye           | Cordelia      | 1700-095 |
| Kingston Fowlds      | Toledo              | Basil         | 1700-071 |
| Stephana Purvis      | Qiongshan           | Raven         | 1700-115 |
| Dorree Harvison      | Miracema            | Shoshone      | 1700-076 |
| Lorianna Leithgoe    | Yako                | Schiller      | 1700-088 |
| Loni Kunneke         | Toritama            | Bluestem      | 1700-101 |
| Obed Van den Bosch   | Huyunxiang          | Kipling       | 1700-092 |
| Marna Rickasse       | Tharyarwady         | Nancy         | 1700-048 |
| Josy Crowest         | Bell Ville          | Helena        | 1700-037 |
| Basil Sammon         | Huerta Grande       | Independence  | 1700-105 |
| Josefa Dregan        | Adygeysk            | Lake View     | 1700-080 |
| Merle Sporton        | Qianfoling          | Holy Cross    | 1700-061 |
| Ericha Ilyenko       | Chengjiao           | Northport     | 1700-044 |
| Jobi Louden          | Ifo                 | Old Gate      | 1700-036 |
| Dorie Gilliam        | Cuijiamatou         | Dennis        | 1700-051 |
| Taber Swains         | Motrico             | Sutherland    | 1700-072 |
| Dewitt Pipping       | Sigaozhuang         | Holmberg      | 1700-102 |
| Emmery MacGowing     | Valerik             | Vermont       | 1700-113 |
| Gnni Erington        | Basco               | Cambridge     | 1700-132 |
| Culver McGuire       | Gobernador Costa    | Scoville      | 1700-110 |
| Lurline Duffie       | Puutuga             | Kinsman       | 1700-107 |
| Milt Oughtright      | Mujiangping         | Prairieview   | 1700-050 |
| Julieta Weblin       | Dubafsari           | Blaine        | 1700-118 |
| Jewel Frowen         | Tigaon              | Mallard       | 1700-086 |
| Ewell Tesdale        | Ikaalinen           | Susan         | 1700-074 |
| Dru Crosio           | Sanxing             | Steensland    | 1700-078 |
| Libbey Dilliway      | Butwal              | Pankratz      | 1700-039 |
| Zack Sartain         | Hengli              | Pankratz      | 1700-129 |
| Lavinia Jurca        | Ramotswa            | Farwell       | 1700-057 |
+----------------------+---------------------+---------------+----------+
100 rows in set (0.02 sec)
```

7.2 Changes

7.2.1 Change 1

```
BEFORE
+------------+-----------------+----------+--------+----------+
| VAT        | name            | street   | city   | zip      |
+------------+-----------------+----------+--------+----------+
| 3129297397 | Jane Sweettooth | Ferreira | Sintra | 1800-148 |
+------------+-----------------+----------+--------+----------+
1 row in set (0.01 sec)

Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

AFTER
+------------+-----------------+------------------+--------------------+----------+
| VAT        | name            | street           | city               | zip      |
+------------+-----------------+------------------+--------------------+----------+
| 3129297397 | Jane Sweettooth | Rua de Mirobriga | Santiago do Cacem | 7540-109 |
+------------+-----------------+------------------+--------------------+----------+
1 row in set (0.02 sec)
```

7.2.2 Change 2

```
BEFORE
+------------+---------+--------------+
```

```
| VAT        | salary  | Appointments |
+------------+---------+--------------+
| 3129297397 | 3500.00 |          159 |
| 3360575526 | 3500.00 |           48 |
| 3601490264 | 2500.00 |           49 |
| 7103563823 | 2500.00 |           49 |
| 7706462800 | 3500.00 |           48 |
+------------+---------+--------------+
5 rows in set (0.01 sec)

Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

AFTER
+------------+---------+--------------+
| VAT        | salary  | Appointments |
+------------+---------+--------------+
| 3129297397 | 3675.00 |          159 |
| 3360575526 | 3500.00 |           48 |
| 3601490264 | 2500.00 |           49 |
| 7103563823 | 2500.00 |           49 |
| 7706462800 | 3500.00 |           48 |
+------------+---------+--------------+
5 rows in set (0.01 sec)
```

7.2.3 Change 3

```
BEFORE
+--------------------+----------------------+
| Total Appointments | Total Consultations |
+--------------------+----------------------+
|                353 |                  300 |
+--------------------+----------------------+
1 row in set (0.05 sec)

Query OK, 1 row affected (0.23 sec)

AFTER
+--------------------+----------------------+
| Total Appointments | Total Consultations |
+--------------------+----------------------+
|                194 |                  187 |
+--------------------+----------------------+
1 row in set (0.02 sec)
```

7.2.4 Change 4

```
BEFORE
+------+---------------+-------------+
| ID   | description   | Average Gap |
+------+---------------+-------------+
| DC_2 | Gingivitis    |     4.31250 |
| DC_2 | Gingivitis    |     4.09375 |
| DC_2 | Gingivitis    |     4.62500 |
| DC_2 | Gingivitis    |     4.31250 |
| DC_2 | Gingivitis    |     4.28125 |
| DC_5 | Periodontitis |     4.65625 |
| DC_5 | Periodontitis |     4.28125 |
| DC_5 | Periodontitis |     4.50000 |
| DC_2 | Gingivitis    |     4.31250 |
| DC_2 | Gingivitis    |     4.37500 |
| DC_5 | Periodontitis |     4.56250 |
| DC_2 | Gingivitis    |     4.21875 |
| DC_5 | Periodontitis |     4.06250 |
| DC_5 | Periodontitis |     4.03125 |
| DC_5 | Periodontitis |     4.18750 |
| DC_5 | Periodontitis |     4.28125 |
| DC_5 | Periodontitis |     4.31250 |
| DC_2 | Gingivitis    |     4.12500 |
| DC_2 | Gingivitis    |     4.37500 |
| DC_5 | Periodontitis |     4.12500 |
+------+---------------+-------------+
```

```
20 rows in set (0.02 sec)

Query OK, 10 rows affected (0.02 sec)
Rows matched: 10  Changed: 10  Warnings: 0

AFTER
+------+---------------+-------------+
| ID   | description   | Average Gap |
+------+---------------+-------------+
| DC_5 | Periodontitis |     4.31250 |
| DC_5 | Periodontitis |     4.09375 |
| DC_5 | Periodontitis |     4.62500 |
| DC_5 | Periodontitis |     4.31250 |
| DC_5 | Periodontitis |     4.28125 |
| DC_5 | Periodontitis |     4.65625 |
| DC_5 | Periodontitis |     4.28125 |
| DC_5 | Periodontitis |     4.50000 |
| DC_5 | Periodontitis |     4.31250 |
| DC_5 | Periodontitis |     4.37500 |
| DC_5 | Periodontitis |     4.56250 |
| DC_5 | Periodontitis |     4.21875 |
| DC_5 | Periodontitis |     4.06250 |
| DC_5 | Periodontitis |     4.03125 |
| DC_5 | Periodontitis |     4.18750 |
| DC_5 | Periodontitis |     4.28125 |
| DC_5 | Periodontitis |     4.31250 |
| DC_5 | Periodontitis |     4.12500 |
| DC_5 | Periodontitis |     4.37500 |
| DC_5 | Periodontitis |     4.12500 |
+------+---------------+-------------+
20 rows in set (0.02 sec)
```

7.3 Views

7.3.1 View 1

```
+---------------------+------+-------+------+
| date_timestamp      | Year | Month | Day  |
+---------------------+------+-------+------+
| 2019-01-03 17:36:43 | 2019 |     1 |    3 |
| 2019-01-03 21:06:58 | 2019 |     1 |    3 |
| 2019-01-06 23:11:38 | 2019 |     1 |    6 |
| 2019-01-07 04:38:23 | 2019 |     1 |    7 |
| 2019-01-09 04:30:43 | 2019 |     1 |    9 |
| 2019-01-13 18:53:03 | 2019 |     1 |   13 |
| 2019-01-14 19:42:36 | 2019 |     1 |   14 |
| 2019-01-16 14:05:34 | 2019 |     1 |   16 |
| 2019-01-18 23:29:12 | 2019 |     1 |   18 |
| 2019-01-27 08:48:03 | 2019 |     1 |   27 |
| 2019-01-27 15:19:27 | 2019 |     1 |   27 |
| 2019-01-28 16:46:11 | 2019 |     1 |   28 |
| 2019-01-30 06:31:54 | 2019 |     1 |   30 |
| 2019-02-07 23:53:52 | 2019 |     2 |    7 |
| 2019-02-10 01:03:25 | 2019 |     2 |   10 |
| 2019-02-10 09:08:19 | 2019 |     2 |   10 |
| 2019-02-14 20:53:52 | 2019 |     2 |   14 |
| 2019-02-17 08:19:03 | 2019 |     2 |   17 |
| 2019-02-17 15:07:57 | 2019 |     2 |   17 |
| 2019-02-18 22:01:04 | 2019 |     2 |   18 |
| 2019-02-19 18:21:01 | 2019 |     2 |   19 |
| 2019-02-22 14:09:46 | 2019 |     2 |   22 |
| 2019-02-23 17:44:48 | 2019 |     2 |   23 |
| 2019-02-24 09:42:29 | 2019 |     2 |   24 |
| 2019-02-25 22:04:12 | 2019 |     2 |   25 |
| 2019-02-26 11:47:45 | 2019 |     2 |   26 |
| 2019-03-01 07:51:55 | 2019 |     3 |    1 |
| 2019-03-06 04:43:48 | 2019 |     3 |    6 |
| 2019-03-06 06:30:33 | 2019 |     3 |    6 |
| 2019-03-09 19:14:55 | 2019 |     3 |    9 |
| 2019-03-10 01:39:05 | 2019 |     3 |   10 |
| 2019-03-16 21:55:29 | 2019 |     3 |   16 |
| 2019-03-18 21:53:21 | 2019 |     3 |   18 |
| 2019-03-20 08:09:40 | 2019 |     3 |   20 |
| 2019-03-26 11:17:47 | 2019 |     3 |   26 |
| 2019-03-27 19:02:39 | 2019 |     3 |   27 |
| 2019-03-28 19:35:08 | 2019 |     3 |   28 |
```

2019-04-01 00:21:29	2019	4	1
2019-04-02 03:52:43	2019	4	2
2019-04-05 06:13:14	2019	4	5
2019-04-06 20:14:12	2019	4	6
2019-04-07 18:59:54	2019	4	7
2019-04-09 10:31:37	2019	4	9
2019-04-09 15:24:52	2019	4	9
2019-04-10 02:58:43	2019	4	10
2019-04-12 13:26:17	2019	4	12
2019-04-17 08:35:30	2019	4	17
2019-04-21 14:02:38	2019	4	21
2019-04-21 20:01:45	2019	4	21
2019-04-22 05:10:00	2019	4	22
2019-04-24 08:45:27	2019	4	24
2019-04-24 12:32:51	2019	4	24
2019-04-29 09:42:46	2019	4	29
2019-04-29 12:48:59	2019	4	29
2019-05-04 20:35:09	2019	5	4
2019-05-06 15:16:05	2019	5	6
2019-05-07 09:31:14	2019	5	7
2019-05-09 06:04:50	2019	5	9
2019-05-12 02:14:25	2019	5	12
2019-05-13 21:14:05	2019	5	13
2019-05-17 12:56:04	2019	5	17
2019-05-19 05:07:34	2019	5	19
2019-05-21 15:47:51	2019	5	21
2019-05-22 03:53:47	2019	5	22
2019-05-24 18:59:43	2019	5	24
2019-05-27 05:22:11	2019	5	27
2019-05-31 14:10:52	2019	5	31
2019-06-03 21:37:25	2019	6	3
2019-06-04 22:31:04	2019	6	4
2019-06-06 04:03:23	2019	6	6
2019-06-14 17:58:37	2019	6	14
2019-06-16 12:38:15	2019	6	16
2019-06-16 23:59:53	2019	6	16
2019-06-17 11:18:57	2019	6	17
2019-06-18 06:17:12	2019	6	18
2019-06-24 12:51:25	2019	6	24
2019-06-25 04:52:20	2019	6	25
2019-06-27 06:39:44	2019	6	27
2019-06-28 23:40:57	2019	6	28
2019-06-29 19:32:10	2019	6	29
2019-06-30 22:59:34	2019	6	30
2019-07-02 03:57:37	2019	7	2
2019-07-02 23:27:08	2019	7	2
2019-07-07 04:57:07	2019	7	7
2019-07-07 15:01:33	2019	7	7
2019-07-10 18:44:25	2019	7	10
2019-07-20 06:38:10	2019	7	20
2019-07-21 02:33:48	2019	7	21
2019-07-26 16:32:48	2019	7	26
2019-07-28 23:05:50	2019	7	28
2019-07-30 22:44:15	2019	7	30
2019-08-02 18:02:16	2019	8	2
2019-08-06 15:20:14	2019	8	6
2019-08-08 15:28:49	2019	8	8
2019-08-13 22:44:49	2019	8	13
2019-08-19 07:03:00	2019	8	19
2019-08-27 08:45:11	2019	8	27
2019-08-28 19:16:48	2019	8	28
2019-09-03 11:17:42	2019	9	3
2019-09-03 14:37:13	2019	9	3
2019-09-06 00:52:44	2019	9	6
2019-09-14 10:00:47	2019	9	14
2019-09-15 20:56:44	2019	9	15
2019-09-22 05:42:14	2019	9	22
2019-09-25 03:30:58	2019	9	25
2019-09-26 00:16:50	2019	9	26
2019-10-08 00:04:54	2019	10	8
2019-10-08 22:12:48	2019	10	8
2019-10-13 11:36:04	2019	10	13
2019-10-17 05:26:29	2019	10	17
2019-10-19 18:14:28	2019	10	19
2019-10-20 19:14:31	2019	10	20
2019-10-31 09:09:21	2019	10	31
2019-01-23 19:25:27	2019	1	23
2019-01-25 04:24:30	2019	1	25
2019-02-18 02:11:29	2019	2	18
2019-02-19 07:18:17	2019	2	19
2019-02-24 07:40:44	2019	2	24
2019-03-06 20:41:46	2019	3	6
2019-03-11 19:39:16	2019	3	11
2019-03-20 04:27:07	2019	3	20

2019-03-24 00:20:36	2019	3	24
2019-03-26 19:00:38	2019	3	26
2019-03-31 01:55:03	2019	3	31
2019-04-03 18:12:55	2019	4	3
2019-04-05 04:55:00	2019	4	5
2019-04-14 05:27:20	2019	4	14
2019-04-28 22:10:05	2019	4	28
2019-05-05 10:59:18	2019	5	5
2019-05-06 00:25:34	2019	5	6
2019-05-11 15:20:33	2019	5	11
2019-05-23 16:14:09	2019	5	23
2019-05-25 22:49:00	2019	5	25
2019-05-30 06:52:17	2019	5	30
2019-06-02 18:07:44	2019	6	2
2019-06-05 04:09:05	2019	6	5
2019-06-14 01:50:29	2019	6	14
2019-06-15 17:56:10	2019	6	15
2019-06-20 09:52:35	2019	6	20
2019-06-30 05:35:32	2019	6	30
2019-07-02 05:14:56	2019	7	2
2019-07-06 08:20:22	2019	7	6
2019-07-08 14:21:12	2019	7	8
2019-07-17 02:31:28	2019	7	17
2019-07-17 13:22:37	2019	7	17
2019-08-06 15:33:46	2019	8	6
2019-08-08 10:17:42	2019	8	8
2019-08-11 08:09:13	2019	8	11
2019-08-21 22:16:41	2019	8	21
2019-09-04 12:25:40	2019	9	4
2019-09-04 23:14:56	2019	9	4
2019-09-05 02:37:15	2019	9	5
2019-09-11 03:34:25	2019	9	11
2019-09-15 12:52:02	2019	9	15
2019-09-25 15:59:27	2019	9	25
2019-09-29 16:56:23	2019	9	29
2019-10-17 06:49:13	2019	10	17
2019-10-17 19:42:46	2019	10	17
2019-10-28 19:50:25	2019	10	28
2019-01-01 09:32:19	2019	1	1
2019-01-09 07:55:32	2019	1	9
2019-01-31 01:13:28	2019	1	31
2019-01-31 17:10:21	2019	1	31
2019-02-17 06:43:04	2019	2	17
2019-02-17 19:03:19	2019	2	17
2019-02-22 16:18:51	2019	2	22
2019-02-23 21:24:23	2019	2	23
2019-02-27 04:25:43	2019	2	27
2019-03-03 19:56:33	2019	3	3
2019-03-23 11:22:36	2019	3	23
2019-03-25 02:48:17	2019	3	25
2019-03-26 16:07:43	2019	3	26
2019-03-28 01:55:42	2019	3	28
2019-04-04 18:41:03	2019	4	4
2019-04-26 05:09:10	2019	4	26
2019-04-27 20:14:16	2019	4	27
2019-05-09 21:28:21	2019	5	9
2019-05-11 01:15:05	2019	5	11
2019-05-15 01:47:28	2019	5	15
2019-05-18 16:23:12	2019	5	18
2019-05-22 11:32:09	2019	5	22
2019-05-28 17:39:21	2019	5	28
2019-06-17 21:12:50	2019	6	17
2019-06-22 18:09:49	2019	6	22
2019-07-11 17:14:36	2019	7	11
2019-07-19 12:25:25	2019	7	19
2019-07-27 22:42:32	2019	7	27
2019-07-28 19:01:40	2019	7	28
2019-08-02 07:29:23	2019	8	2
2019-08-07 23:36:01	2019	8	7
2019-08-10 20:43:26	2019	8	10
2019-08-14 07:58:17	2019	8	14
2019-08-19 23:25:13	2019	8	19
2019-08-20 13:12:26	2019	8	20
2019-08-25 21:58:01	2019	8	25
2019-08-29 09:32:57	2019	8	29
2019-09-02 03:40:39	2019	9	2
2019-09-13 15:07:03	2019	9	13
2019-09-14 01:13:33	2019	9	14
2019-09-20 21:04:06	2019	9	20
2019-09-30 13:51:05	2019	9	30
2019-10-04 12:29:16	2019	10	4
2019-10-05 15:17:47	2019	10	5
2019-10-07 08:12:00	2019	10	7
2019-10-07 22:46:08	2019	10	7

2019-10-19 06:10:03	2019	10	19
2019-01-07 01:35:09	2019	1	7
2019-01-15 21:47:28	2019	1	15
2019-01-30 13:15:26	2019	1	30
2019-02-03 09:07:46	2019	2	3
2019-03-08 06:51:42	2019	3	8
2019-04-01 15:50:06	2019	4	1
2019-04-11 07:56:30	2019	4	11
2019-04-11 12:01:19	2019	4	11
2019-04-14 01:58:29	2019	4	14
2019-04-15 02:09:47	2019	4	15
2019-04-20 19:43:49	2019	4	20
2019-04-24 01:40:47	2019	4	24
2019-04-24 21:33:14	2019	4	24
2019-04-27 02:09:14	2019	4	27
2019-05-01 03:21:33	2019	5	1
2019-05-22 21:43:23	2019	5	22
2019-05-26 12:55:45	2019	5	26
2019-05-28 18:03:55	2019	5	28
2019-05-30 00:09:52	2019	5	30
2019-06-10 12:30:40	2019	6	10
2019-06-25 19:38:10	2019	6	25
2019-06-29 15:58:34	2019	6	29
2019-06-29 21:27:28	2019	6	29
2019-06-30 19:27:36	2019	6	30
2019-07-04 04:10:40	2019	7	4
2019-07-04 05:17:11	2019	7	4
2019-07-07 05:08:29	2019	7	7
2019-07-21 16:26:58	2019	7	21
2019-07-26 03:36:21	2019	7	26
2019-08-02 03:23:43	2019	8	2
2019-08-02 15:24:07	2019	8	2
2019-08-04 11:33:09	2019	8	4
2019-08-06 03:30:28	2019	8	6
2019-08-09 00:20:10	2019	8	9
2019-08-19 15:27:32	2019	8	19
2019-08-20 04:17:24	2019	8	20
2019-08-25 08:24:32	2019	8	25
2019-08-26 15:17:45	2019	8	26
2019-08-27 00:57:54	2019	8	27
2019-09-11 06:17:25	2019	9	11
2019-09-21 07:34:05	2019	9	21
2019-09-30 10:19:19	2019	9	30
2019-10-04 22:08:15	2019	10	4
2019-10-05 11:18:55	2019	10	5
2019-10-06 05:40:17	2019	10	6
2019-10-23 15:19:34	2019	10	23
2019-10-24 00:21:36	2019	10	24
2019-01-06 04:57:10	2019	1	6
2019-01-07 05:03:22	2019	1	7
2019-01-11 07:17:06	2019	1	11
2019-01-15 09:03:04	2019	1	15
2019-01-18 20:29:04	2019	1	18
2019-01-20 13:38:15	2019	1	20
2019-01-23 14:00:58	2019	1	23
2019-01-25 21:27:28	2019	1	25
2019-01-28 07:44:43	2019	1	28
2019-02-04 13:01:22	2019	2	4
2019-02-09 01:00:51	2019	2	9
2019-02-11 09:43:18	2019	2	11
2019-02-15 06:21:46	2019	2	15
2019-02-16 10:51:27	2019	2	16
2019-02-19 18:56:42	2019	2	19
2019-03-07 02:12:48	2019	3	7
2019-03-26 13:31:06	2019	3	26
2019-03-31 05:44:53	2019	3	31
2019-04-01 18:37:50	2019	4	1
2019-04-11 10:26:03	2019	4	11
2019-04-11 12:21:44	2019	4	11
2019-04-14 12:18:12	2019	4	14
2019-04-19 11:44:28	2019	4	19
2019-04-26 14:54:25	2019	4	26
2019-05-01 09:16:23	2019	5	1
2019-05-02 02:23:26	2019	5	2
2019-05-05 23:57:48	2019	5	5
2019-05-10 06:43:36	2019	5	10
2019-06-03 03:45:13	2019	6	3
2019-06-17 08:39:52	2019	6	17
2019-06-24 02:24:32	2019	6	24
2019-07-03 18:14:03	2019	7	3
2019-07-04 02:12:34	2019	7	4
2019-07-05 23:26:29	2019	7	5
2019-07-08 01:19:30	2019	7	8
2019-07-09 04:01:34	2019	7	9

```
| 2019-07-24 21:07:31 | 2019 |      7 |   24 |
| 2019-08-11 05:12:59 | 2019 |      8 |   11 |
| 2019-08-13 04:03:30 | 2019 |      8 |   13 |
| 2019-08-21 20:08:15 | 2019 |      8 |   21 |
| 2019-08-26 21:13:21 | 2019 |      8 |   26 |
| 2019-08-31 06:14:49 | 2019 |      8 |   31 |
| 2019-10-02 11:14:47 | 2019 |     10 |    2 |
| 2019-10-04 20:12:03 | 2019 |     10 |    4 |
| 2019-10-10 04:07:54 | 2019 |     10 |   10 |
| 2019-10-21 15:36:59 | 2019 |     10 |   21 |
| 2019-10-30 02:08:03 | 2019 |     10 |   30 |
+---------------------+------+--------+------+
300 rows in set (0,10 sec)
```

7.3.2 View 2

```
+------------+--------+-----+
| VAT        | Gender | Age |
+------------+--------+-----+
| 1065284477 | Female |  21 |
| 1125478009 | Female |  19 |
| 1251928457 | Female |  57 |
| 1288938172 | Female |  20 |
| 1315824037 | Female |  50 |
| 1347404557 | Female |  57 |
| 1377212573 | Female |  15 |
| 1384288459 | Male   |  53 |
| 1443765898 | Male   |  17 |
| 1495182097 | Female |  71 |
| 1564926875 | Female |  44 |
| 1774361024 | Female |  28 |
| 1889223122 | Female |  64 |
| 1944111515 | Female |  26 |
| 2125409049 | Male   |  14 |
| 2177633016 | Male   |  71 |
| 2261570434 | Female |  16 |
| 2380246766 | Female |  75 |
| 2512980241 | Female |  72 |
| 2542736826 | Male   |  19 |
| 2756537095 | Male   |  63 |
| 2785024496 | Female |  24 |
| 3014867017 | Female |  27 |
| 3027642977 | Female |  71 |
| 3037318718 | Female |  14 |
| 3044991851 | Male   |  45 |
| 3045325542 | Female |  46 |
| 3054429349 | Male   |  25 |
| 3114584891 | Female |  15 |
| 3252775826 | Male   |  61 |
| 3260652693 | Male   |  44 |
| 3269972780 | Female |  77 |
| 3333638031 | Female |  51 |
| 3594045692 | Male   |  44 |
| 3721688004 | Female |  20 |
| 3787344317 | Female |  18 |
| 3852397258 | Female |  37 |
| 4175670708 | Female |  69 |
| 4192964808 | Female |  24 |
| 4321553298 | Male   |  62 |
| 4444304768 | Female |  57 |
| 4517801011 | Male   |  22 |
| 4666833838 | Female |  72 |
| 4681803112 | Male   |  37 |
| 4834682164 | Female |  70 |
| 5057692187 | Female |  72 |
| 5071193401 | Male   |  50 |
| 5153824867 | Male   |  71 |
| 5168802859 | Male   |  51 |
| 5205804402 | Male   |  28 |
| 5222487955 | Female |  50 |
| 5375272189 | Male   |  76 |
| 5403873299 | Male   |  60 |
| 5484250919 | Male   |  39 |
| 5680263126 | Male   |  14 |
| 5819093380 | Female |  67 |
| 5880863332 | Female |  54 |
| 6037486263 | Male   |  63 |
| 6071361884 | Male   |  58 |
```

```
| 6101114711 | Male   |  76 |
| 6164397671 | Male   |  39 |
| 6324221520 | Female |  68 |
| 6382296314 | Male   |  23 |
| 6409151986 | Male   |  62 |
| 6633747219 | Male   |  51 |
| 6969696969 | Male   |  69 |
| 6978678991 | Male   |  22 |
| 7129224524 | Male   |  75 |
| 7144038478 | Male   |  40 |
| 7158549937 | Female |  24 |
| 7277773552 | Male   |  21 |
| 7370461688 | Male   |  30 |
| 7412563835 | Female |  66 |
| 7416995863 | Male   |  41 |
| 7587186491 | Female |  41 |
| 7593875652 | Female |  41 |
| 7678417951 | Female |  56 |
| 7710870439 | Female |  67 |
| 7763386444 | Male   |  58 |
| 7807464581 | Female |  60 |
| 8104963722 | Female |  54 |
| 8236190259 | Male   |  59 |
| 8237760990 | Female |  79 |
| 8272343344 | Male   |  25 |
| 8301432130 | Female |  50 |
| 8372266270 | Female |  16 |
| 8379611058 | Female |  60 |
| 8443316525 | Male   |  61 |
| 8632411626 | Male   |  57 |
| 8689053273 | Male   |  49 |
| 8847783511 | Female |  21 |
| 8924571573 | Male   |  63 |
| 9043514344 | Female |  51 |
| 9159797588 | Male   |  17 |
| 9261507324 | Female |  18 |
| 9262710939 | Female |  32 |
| 9309326893 | Male   |  40 |
| 9366456197 | Female |  38 |
| 9666407194 | Female |  51 |
| 9795584884 | Male   |  55 |
| 9916815067 | Female |  65 |
+------------+--------+-----+
101 rows in set (0,06 sec)
```

7.3.3 View 3

```
+----------+--------------------+
| ZIP      | City               |
+----------+--------------------+
| 1700-089 | Tycher             |
| 1700-108 | Jaciara            |
| 1700-073 | Lisbon             |
| 1700-040 | Sasar              |
| 1700-077 | Kamuli             |
| 1700-125 | Beicheng           |
| 1700-056 | Yaguaraparo        |
| 1700-082 | Xiangzikou         |
| 1700-041 | Kalmar             |
| 1700-109 | Vila de Rei        |
| 1700-059 | Sengeti            |
| 1700-065 | Sung Noen          |
| 1700-135 | Rustaq             |
| 1700-081 | Gazanjyk           |
| 1700-058 | El Ocotana         |
| 1700-055 | Kidodi             |
| 1700-096 | Nogent-sur-Marne   |
| 1700-043 | Pomerode           |
| 1700-083 | Quartier Morin     |
| 1700-112 | Ranilug            |
| 1700-093 | Guantang           |
| 1700-066 | Sydney             |
| 1700-116 | Saffle             |
| 1700-062 | Hammam Sousse      |
| 1700-067 | Kremidivka         |
| 1700-042 | Nyurba             |
| 1700-097 | San Juan de Mata   |
| 1700-120 | Youcheng           |
```

```
| 1700-054 | Kista              |
| 1700-106 | Ifon               |
| 1700-087 | Bykov              |
| 1700-060 | Wangkung           |
| 1700-121 | Zhongshan          |
| 1700-122 | Ocongate           |
| 1700-075 | Vilcan             |
| 1700-099 | Huochezhan         |
| 1700-064 | Abelheira          |
| 1700-100 | Guadalupe          |
| 1700-134 | Sangmalima         |
| 1700-085 | La Roxas           |
| 1700-114 | Davila             |
| 1700-123 | Wenlin             |
| 1700-124 | Risalpur           |
| 1700-133 | Daxing             |
| 1700-131 | Jargalant          |
| 1700-069 | Sulaq              |
| 1700-104 | Levuka             |
| 1700-091 | Prakhon Chai       |
| 1700-098 | Karangkancana      |
| 1700-079 | Kuala Lumpur       |
| 1700-049 | Heshang            |
| 1700-127 | Loreto             |
| 1700-063 | Lazaro Cardenas    |
| 1700-038 | Taupo              |
| 1700-046 | Chervonohryhorivka |
| 1700-119 | Hecheng            |
| 1700-070 | Dashiqiao          |
| 1700-117 | Palalang           |
| 1700-047 | Ukata              |
| 1700-111 | Santiago Vazquez   |
| 1700-045 | Kedungdoro         |
| 1700-126 | Changsha           |
| 1700-128 | Fresno             |
| 1700-103 | Invercargill       |
| 1700-084 | Gajah              |
| 1700-136 | Porto              |
| 1700-130 | Houston            |
| 1700-090 | Manzil Zalafah     |
| 1700-053 | Manalongon         |
| 1700-094 | Candelaria         |
| 1700-068 | Ejigbo             |
| 1700-052 | Mombasa            |
| 1700-095 | Laboulaye          |
| 1700-071 | Toledo             |
| 1700-115 | Qiongshan          |
| 1700-076 | Miracema           |
| 1700-088 | Yako               |
| 1700-101 | Toritama           |
| 1700-092 | Huyunxiang         |
| 1700-048 | Tharyarwady        |
| 1700-037 | Bell Ville         |
| 1700-105 | Huerta Grande      |
| 1700-080 | Adygeysk           |
| 1700-061 | Qianfoling         |
| 1700-044 | Chengjiao          |
| 1700-036 | Ifo                |
| 1700-051 | Cuijiamatou        |
| 1700-072 | Motrico            |
| 1700-102 | Sigaozhuang        |
| 1700-113 | Valerik            |
| 1700-132 | Basco              |
| 1700-110 | Gobernador Costa   |
| 1700-107 | Puutuga            |
| 1700-050 | Mujiangping        |
| 1700-118 | Dubafsari          |
| 1700-086 | Tigaon             |
| 1700-074 | Ikaalinen          |
| 1700-078 | Sanxing            |
| 1700-039 | Butwal             |
| 1700-129 | Hengli             |
| 1700-057 | Ramotswa           |
+----------+--------------------+
101 rows in set (0,04 sec)
```

7.3.4 View 4

Client VAT	Date	ZIP	# Procedures	# Medication	# Diagnostic Codes
1065284477	2019-02-19 18:21:01	1700-089	1	0	1
1065284477	2019-03-24 00:20:36	1700-089	1	1	1
1065284477	2019-06-06 04:03:23	1700-089	1	1	1
1065284477	2019-06-20 09:52:35	1700-089	1	1	1
1125478009	2019-04-21 14:02:38	1700-108	1	1	1
1125478009	2019-09-21 07:34:05	1700-108	1	1	1
1251928457	2019-04-14 05:27:20	1700-073	1	1	1
1251928457	2019-06-14 01:50:29	1700-073	1	1	1
1251928457	2019-07-19 12:25:25	1700-073	1	1	1
1251928457	2019-08-02 18:02:16	1700-073	1	0	1
1288938172	2019-05-23 16:14:09	1700-040	1	1	1
1288938172	2019-07-07 04:57:07	1700-040	1	1	1
1288938172	2019-08-06 03:30:28	1700-040	1	1	1
1315824037	2019-01-23 19:25:27	1700-077	1	1	1
1315824037	2019-01-25 04:24:30	1700-077	1	1	1
1315824037	2019-02-22 16:18:51	1700-077	1	1	1
1315824037	2019-09-25 03:30:58	1700-077	1	1	1
1347404557	2019-06-03 21:37:25	1700-125	1	1	1
1377212573	2019-02-18 02:11:29	1700-056	1	1	1
1377212573	2019-03-08 06:51:42	1700-056	1	1	1
1377212573	2019-08-19 15:27:32	1700-056	1	1	1
1384288459	2019-02-23 17:44:48	1700-082	1	0	1
1384288459	2019-04-06 20:14:12	1700-082	1	1	1
1384288459	2019-04-14 12:18:12	1700-082	1	1	1
1384288459	2019-07-08 01:19:30	1700-082	1	1	1
1443765898	2019-01-16 14:05:34	1700-041	1	1	1
1443765898	2019-01-25 21:27:28	1700-041	1	1	1
1443765898	2019-03-31 01:55:03	1700-041	1	1	1
1495182097	2019-05-06 00:25:34	1700-109	1	1	1
1495182097	2019-06-16 12:38:15	1700-109	1	1	1
1564926875	2019-02-17 08:19:03	1700-059	1	1	1
1564926875	2019-02-27 04:25:43	1700-059	1	1	1
1564926875	2019-04-26 05:09:10	1700-059	1	0	1
1564926875	2019-06-25 19:38:10	1700-059	1	0	1
1774361024	2019-05-04 20:35:09	1700-065	1	1	1
1774361024	2019-06-05 04:09:05	1700-065	1	1	1
1774361024	2019-09-05 02:37:15	1700-065	1	1	1
1774361024	2019-09-20 21:04:06	1700-065	1	1	1
1889223122	2019-08-28 19:16:48	1700-135	1	1	1
1944111515	2019-03-06 04:43:48	1700-081	1	1	1
1944111515	2019-03-16 21:55:29	1700-081	1	1	1
1944111515	2019-06-15 17:56:10	1700-081	1	1	1
1944111515	2019-07-17 13:22:37	1700-081	1	1	1
2125409049	2019-02-23 21:24:23	1700-058	1	1	1
2125409049	2019-07-09 04:01:34	1700-058	1	1	1
2125409049	2019-08-26 21:13:21	1700-058	1	1	1
2125409049	2019-09-22 05:42:14	1700-058	1	1	1
2177633016	2019-07-11 17:14:36	1700-055	1	1	1
2177633016	2019-07-21 16:26:58	1700-055	1	1	1
2177633016	2019-08-29 09:32:57	1700-055	1	1	1
2261570434	2019-04-24 21:33:14	1700-096	1	1	1
2261570434	2019-07-26 03:36:21	1700-096	1	1	1
2261570434	2019-08-13 22:44:49	1700-096	1	0	1
2261570434	2019-10-13 11:36:04	1700-096	1	1	1
2380246766	2019-03-03 19:56:33	1700-043	1	1	1
2380246766	2019-04-10 02:58:43	1700-043	1	1	1
2380246766	2019-04-27 02:09:14	1700-043	1	1	1
2512980241	2019-02-17 06:43:04	1700-083	1	1	1
2512980241	2019-03-26 16:07:43	1700-083	1	1	1
2512980241	2019-06-29 19:32:10	1700-083	1	1	1
2512980241	2019-07-07 15:01:33	1700-083	1	1	1
2542736826	2019-05-30 00:09:52	1700-112	1	1	1
2542736826	2019-07-26 16:32:48	1700-112	1	0	1
2756537095	2019-02-19 07:18:17	1700-093	1	1	1
2756537095	2019-03-01 07:51:55	1700-093	1	1	1
2756537095	2019-03-28 19:35:08	1700-093	1	1	1
2756537095	2019-08-08 10:17:42	1700-093	1	0	1
2785024496	2019-06-18 06:17:12	1700-066	1	1	1
2785024496	2019-07-04 05:17:11	1700-066	1	0	1
2785024496	2019-10-10 04:07:54	1700-066	1	1	1
2785024496	2019-10-21 15:36:59	1700-066	1	1	1
3014867017	2019-02-22 14:09:46	1700-116	1	1	1
3014867017	2019-05-26 12:55:45	1700-116	1	0	1
3027642977	2019-02-10 09:08:19	1700-062	1	1	1
3027642977	2019-06-03 03:45:13	1700-062	1	1	1
3027642977	2019-07-04 02:12:34	1700-062	1	1	1
3027642977	2019-10-04 22:08:15	1700-062	1	1	1
3037318718	2019-04-09 15:24:52	1700-067	1	1	1
3037318718	2019-05-05 10:59:18	1700-067	1	1	1
3037318718	2019-05-28 17:39:21	1700-067	1	1	1
3037318718	2019-09-02 03:40:39	1700-067	1	1	1

3044991851	2019-04-01 00:21:29	1700-042	1	1	1
3044991851	2019-08-21 20:08:15	1700-042	1	1	1
3044991851	2019-10-19 06:10:03	1700-042	1	1	1
3045325542	2019-01-30 06:31:54	1700-097	1	1	1
3045325542	2019-05-09 06:04:50	1700-097	1	1	1
3045325542	2019-08-11 08:09:13	1700-097	1	1	1
3045325542	2019-10-28 19:50:25	1700-097	1	1	1
3054429349	2019-01-13 18:53:03	1700-120	1	1	1
3054429349	2019-09-11 06:17:25	1700-120	1	1	1
3114584891	2019-01-18 20:29:04	1700-054	1	1	1
3114584891	2019-03-07 02:12:48	1700-054	1	0	1
3114584891	2019-08-02 07:29:23	1700-054	1	1	1
3252775826	2019-02-09 01:00:51	1700-106	1	1	1
3252775826	2019-06-24 12:51:25	1700-106	1	1	1
3260652693	2019-01-03 17:36:43	1700-087	1	1	1
3260652693	2019-03-20 08:09:40	1700-087	1	0	1
3260652693	2019-05-18 16:23:12	1700-087	1	1	1
3260652693	2019-08-19 23:25:13	1700-087	1	1	1
3269972780	2019-05-22 21:43:23	1700-060	1	1	1
3269972780	2019-06-25 04:52:20	1700-060	1	1	1
3269972780	2019-07-05 23:26:29	1700-060	1	1	1
3269972780	2019-08-20 04:17:24	1700-060	1	1	1
3333638031	2019-03-26 19:00:38	1700-121	1	1	1
3333638031	2019-04-22 05:10:00	1700-121	1	1	1
3594045692	2019-04-09 10:31:37	1700-122	1	1	1
3594045692	2019-08-11 05:12:59	1700-122	1	1	1
3721688004	2019-01-09 07:55:32	1700-075	1	1	1
3721688004	2019-04-05 04:55:00	1700-075	1	1	1
3721688004	2019-06-17 21:12:50	1700-075	1	1	1
3721688004	2019-09-26 00:16:50	1700-075	1	1	1
3787344317	2019-02-17 19:03:19	1700-099	1	1	1
3787344317	2019-03-23 11:22:36	1700-099	1	1	1
3787344317	2019-04-02 03:52:43	1700-099	1	1	1
3787344317	2019-10-17 05:26:29	1700-099	1	1	1
3852397258	2019-02-19 18:56:42	1700-064	1	1	1
3852397258	2019-07-21 02:33:48	1700-064	1	1	1
3852397258	2019-10-06 05:40:17	1700-064	1	0	1
3852397258	2019-10-24 00:21:36	1700-064	1	1	1
4175670708	2019-04-24 08:45:27	1700-100	1	1	1
4175670708	2019-08-04 11:33:09	1700-100	1	0	1
4175670708	2019-08-26 15:17:45	1700-100	1	1	1
4192964808	2019-06-14 17:58:37	1700-134	1	1	1
4321553298	2019-05-11 15:20:33	1700-085	1	1	1
4321553298	2019-07-02 05:14:56	1700-085	1	1	1
4321553298	2019-07-28 23:05:50	1700-085	1	1	1
4321553298	2019-10-08 22:12:48	1700-085	1	1	1
4444304768	2019-02-25 22:04:12	1700-114	1	1	1
4444304768	2019-03-26 13:31:06	1700-114	1	1	1
4517801011	2019-05-12 02:14:25	1700-123	1	1	1
4517801011	2019-05-22 11:32:09	1700-123	1	0	1
4666833838	2019-08-27 08:45:11	1700-124	1	1	1
4681803112	2019-06-28 23:40:57	1700-133	1	1	1
4834682164	2019-05-31 14:10:52	1700-131	1	1	1
5057692187	2019-01-27 15:19:27	1700-069	1	1	1
5057692187	2019-09-04 12:25:40	1700-069	1	1	1
5057692187	2019-09-25 15:59:27	1700-069	1	1	1
5057692187	2019-09-30 13:51:05	1700-069	1	1	1
5071193401	2019-05-13 21:14:05	1700-104	1	1	1
5071193401	2019-08-02 03:23:43	1700-104	1	1	1
5071193401	2019-08-27 00:57:54	1700-104	1	1	1
5153824867	2019-04-27 20:14:16	1700-091	1	1	1
5153824867	2019-04-29 12:48:59	1700-091	1	1	1
5153824867	2019-06-22 18:09:49	1700-091	1	1	1
5153824867	2019-10-19 18:14:28	1700-091	1	1	1
5168802859	2019-04-12 13:26:17	1700-098	1	1	1
5168802859	2019-04-29 09:42:46	1700-098	1	0	1
5168802859	2019-05-02 02:23:26	1700-098	1	1	1
5168802859	2019-08-31 06:14:49	1700-098	1	1	1
5205804402	2019-03-06 06:30:33	1700-079	1	1	1
5205804402	2019-03-25 02:48:17	1700-079	1	0	1
5205804402	2019-08-06 15:20:14	1700-079	1	1	1
5205804402	2019-10-07 22:46:08	1700-079	1	1	1
5222487955	2019-03-20 04:27:07	1700-049	1	1	1
5222487955	2019-07-24 21:07:31	1700-049	1	1	1
5222487955	2019-09-15 12:52:02	1700-049	1	1	1
5375272189	2019-02-24 09:42:29	1700-127	1	1	1
5403873299	2019-04-04 18:41:03	1700-063	1	1	1
5403873299	2019-07-20 06:38:10	1700-063	1	1	1
5403873299	2019-08-06 15:33:46	1700-063	1	1	1
5403873299	2019-10-07 08:12:00	1700-063	1	1	1
5484250919	2019-04-26 14:54:25	1700-038	1	1	1
5484250919	2019-05-27 05:22:11	1700-038	1	1	1
5484250919	2019-08-10 20:43:26	1700-038	1	1	1
5680263126	2019-01-15 09:03:04	1700-046	1	1	1

5680263126	2019-05-11 01:15:05	1700-046	1	1	1
5680263126	2019-05-21 15:47:51	1700-046	1	1	1
5819093380	2019-01-01 09:32:19	1700-119	1	1	1
5819093380	2019-09-03 11:17:42	1700-119	1	0	1
5880863332	2019-05-05 23:57:48	1700-070	1	1	1
5880863332	2019-05-10 06:43:36	1700-070	1	1	1
5880863332	2019-06-17 11:18:57	1700-070	1	1	1
5880863332	2019-06-30 19:27:36	1700-070	1	1	1
6037486263	2019-01-09 04:30:43	1700-117	1	1	1
6037486263	2019-04-28 22:10:05	1700-117	1	1	1
6071361884	2019-01-07 01:35:09	1700-047	1	1	1
6071361884	2019-01-18 23:29:12	1700-047	1	1	1
6071361884	2019-09-14 01:13:33	1700-047	1	0	1
6101114711	2019-04-07 18:59:54	1700-111	1	1	1
6101114711	2019-05-09 21:28:21	1700-111	1	1	1
6164397671	2019-01-27 08:48:03	1700-045	1	1	1
6164397671	2019-06-02 18:07:44	1700-045	1	1	1
6164397671	2019-07-03 18:14:03	1700-045	1	0	1
6324221520	2019-01-03 21:06:58	1700-126	1	1	1
6382296314	2019-04-17 08:35:30	1700-128	1	1	1
6409151986	2019-01-31 17:10:21	1700-103	1	1	1
6409151986	2019-05-24 18:59:43	1700-103	1	1	1
6409151986	2019-10-04 12:29:16	1700-103	1	1	1
6633747219	2019-03-09 19:14:55	1700-084	1	1	1
6633747219	2019-04-11 12:01:19	1700-084	1	1	1
6633747219	2019-08-25 08:24:32	1700-084	1	1	1
6633747219	2019-09-03 14:37:13	1700-084	1	1	1
6978678991	2019-03-10 01:39:05	1700-130	1	1	1
7129224524	2019-01-11 07:17:06	1700-090	1	1	1
7129224524	2019-04-24 12:32:51	1700-090	1	1	1
7129224524	2019-08-13 04:03:30	1700-090	1	0	1
7129224524	2019-09-06 00:52:44	1700-090	1	1	1
7144038478	2019-04-01 18:37:50	1700-053	1	1	1
7144038478	2019-06-30 05:35:32	1700-053	1	0	1
7144038478	2019-10-17 19:42:46	1700-053	1	1	1
7158549937	2019-01-28 07:44:43	1700-094	1	1	1
7158549937	2019-03-26 11:17:47	1700-094	1	1	1
7158549937	2019-06-24 02:24:32	1700-094	1	1	1
7158549937	2019-06-27 06:39:44	1700-094	1	1	1
7277773552	2019-01-07 05:03:22	1700-068	1	1	1
7277773552	2019-01-30 13:15:26	1700-068	1	1	1
7277773552	2019-05-01 03:21:33	1700-068	1	1	1
7277773552	2019-05-17 12:56:04	1700-068	1	1	1
7370461688	2019-04-11 07:56:30	1700-052	1	1	1
7370461688	2019-07-04 04:10:40	1700-052	1	1	1
7370461688	2019-09-29 16:56:23	1700-052	1	0	1
7412563835	2019-05-22 03:53:47	1700-095	1	1	1
7412563835	2019-07-27 22:42:32	1700-095	1	1	1
7412563835	2019-07-28 19:01:40	1700-095	1	1	1
7412563835	2019-09-15 20:56:44	1700-095	1	1	1
7416995863	2019-01-31 01:13:28	1700-071	1	1	1
7416995863	2019-02-14 20:53:52	1700-071	1	1	1
7416995863	2019-02-24 07:40:44	1700-071	1	0	1
7416995863	2019-03-28 01:55:42	1700-071	1	1	1
7587186491	2019-07-02 23:27:08	1700-115	1	1	1
7587186491	2019-09-13 15:07:03	1700-115	1	1	1
7593875652	2019-01-06 04:57:10	1700-076	1	1	1
7593875652	2019-05-28 18:03:55	1700-076	1	1	1
7593875652	2019-06-29 21:27:28	1700-076	1	1	1
7593875652	2019-08-08 15:28:49	1700-076	1	1	1
7678417951	2019-02-18 22:01:04	1700-088	1	1	1
7678417951	2019-10-05 11:18:55	1700-088	1	1	1
7678417951	2019-10-23 15:19:34	1700-088	1	1	1
7678417951	2019-10-31 09:09:21	1700-088	1	1	1
7710870439	2019-05-25 22:49:00	1700-101	1	1	1
7710870439	2019-07-10 18:44:25	1700-101	1	1	1
7710870439	2019-09-04 23:14:56	1700-101	1	1	1
7763386444	2019-04-15 02:09:47	1700-092	1	1	1
7763386444	2019-04-20 19:43:49	1700-092	1	1	1
7763386444	2019-07-02 03:57:37	1700-092	1	1	1
7763386444	2019-09-14 10:00:47	1700-092	1	1	1
7807464581	2019-03-27 19:02:39	1700-048	1	1	1
7807464581	2019-08-09 00:20:10	1700-048	1	1	1
7807464581	2019-10-17 06:49:13	1700-048	1	1	1
8104963722	2019-01-23 14:00:58	1700-037	1	1	1
8104963722	2019-06-30 22:59:34	1700-037	1	0	1
8104963722	2019-07-17 02:31:28	1700-037	1	1	1
8236190259	2019-01-28 16:46:11	1700-105	1	1	1
8236190259	2019-05-30 06:52:17	1700-105	1	0	1
8237760990	2019-03-18 21:53:21	1700-080	1	1	1
8237760990	2019-06-16 23:59:53	1700-080	1	1	1
8237760990	2019-07-07 05:08:29	1700-080	1	0	1
8237760990	2019-09-30 10:19:19	1700-080	1	1	1
8272343344	2019-04-03 18:12:55	1700-061	1	1	1

8272343344	2019-04-05 06:13:14	1700-061	1	1	1
8272343344	2019-07-06 08:20:22	1700-061	1	1	1
8272343344	2019-08-07 23:36:01	1700-061	1	1	1
8301432130	2019-01-06 23:11:38	1700-044	1	1	1
8301432130	2019-04-24 01:40:47	1700-044	1	1	1
8301432130	2019-07-08 14:21:12	1700-044	1	1	1
8372266270	2019-01-07 04:38:23	1700-036	1	1	1
8372266270	2019-01-15 21:47:28	1700-036	1	1	1
8372266270	2019-03-11 19:39:16	1700-036	1	1	1
8379611058	2019-04-14 01:58:29	1700-051	1	1	1
8379611058	2019-08-14 07:58:17	1700-051	1	1	1
8379611058	2019-08-25 21:58:01	1700-051	1	1	1
8443316525	2019-02-07 23:53:52	1700-072	1	1	1
8443316525	2019-04-11 10:26:03	1700-072	1	1	1
8443316525	2019-06-10 12:30:40	1700-072	1	1	1
8443316525	2019-06-29 15:58:34	1700-072	1	0	1
8632411626	2019-02-16 10:51:27	1700-102	1	1	1
8632411626	2019-04-11 12:21:44	1700-102	1	1	1
8632411626	2019-06-04 22:31:04	1700-102	1	1	1
8689053273	2019-02-17 15:07:57	1700-113	1	1	1
8689053273	2019-08-21 22:16:41	1700-113	1	1	1
8847783511	2019-08-19 07:03:00	1700-132	1	1	1
8924571573	2019-04-21 20:01:45	1700-110	1	1	1
8924571573	2019-06-17 08:39:52	1700-110	1	1	1
9043514344	2019-02-10 01:03:25	1700-107	1	1	1
9043514344	2019-08-20 13:12:26	1700-107	1	1	1
9159797588	2019-04-19 11:44:28	1700-050	1	1	1
9159797588	2019-05-15 01:47:28	1700-050	1	1	1
9159797588	2019-10-30 02:08:03	1700-050	1	1	1
9261507324	2019-02-11 09:43:18	1700-118	1	1	1
9261507324	2019-10-20 19:14:31	1700-118	1	1	1
9262710939	2019-01-14 19:42:36	1700-086	1	1	1
9262710939	2019-02-04 13:01:22	1700-086	1	1	1
9262710939	2019-07-30 22:44:15	1700-086	1	1	1
9262710939	2019-10-04 20:12:03	1700-086	1	0	1
9309326893	2019-05-01 09:16:23	1700-074	1	1	1
9309326893	2019-05-07 09:31:14	1700-074	1	1	1
9309326893	2019-08-02 15:24:07	1700-074	1	1	1
9309326893	2019-10-02 11:14:47	1700-074	1	1	1
9366456197	2019-02-03 09:07:46	1700-078	1	0	1
9366456197	2019-02-15 06:21:46	1700-078	1	1	1
9366456197	2019-03-31 05:44:53	1700-078	1	1	1
9366456197	2019-10-08 00:04:54	1700-078	1	1	1
9666407194	2019-02-26 11:47:45	1700-039	1	1	1
9666407194	2019-04-01 15:50:06	1700-039	1	1	1
9666407194	2019-10-05 15:17:47	1700-039	1	1	1
9795584884	2019-05-19 05:07:34	1700-129	1	1	1
9916815067	2019-01-20 13:38:15	1700-057	1	1	1
9916815067	2019-03-06 20:41:46	1700-057	1	1	1
9916815067	2019-05-06 15:16:05	1700-057	1	1	1
9916815067	2019-09-11 03:34:25	1700-057	1	1	1

300 rows in set (0,19 sec)