Implementing the Database

Information Systems and Databases

Authors

Jose Silva - 84109
Luís Simões - 81282
Pedro Ferreira - 84166

November 24, 2022

Contents

1 Introduction

This project concerns the development of an information system supporting the day-to-day operations of a dental clinic. This second part tackles the creation of the database using an *SQL* language *MYSQL*, following a relational model similar to the one developed on the first part of this project.

After the creation, it is required to test and optimize it, in order to prove the suitability of the constructed database for the purpose of a dental clinic.

2 Create Database

First of all, it was done the drop off all tables to make sure that there wasn't any table already on the database that would conflict with the new one. These drops had to be done in specific order to ensure that all tables would be deleted, due to their relations between each others, regarding foreign keys.

Listing 1: Drop tables

```
drop table if exists procedure_charting;
drop table if exists teeth;
drop table if exists procedure_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 receptionist;
drop table if exists doctor;
drop table if exists nurse;
drop table if exists phone_number_employee;
drop table if exists employee;
```

While creating all the tables, the reaction to an update/deletion on the foreign keys was takes into account and in most cases it was chosen to do it on cascade. The only exception is regarding supervision reports where even if the trainee doctor is deleted the report will continue on the database.

Listing 2:

```
1
2
3  create table employee
4     (VAT char(10),
5     name varchar(255),
6     birth_date DATE,
7     street varchar(255),
8     city varchar(255),
9     zip char(9),
10    IBAN char(26) unique not null,
11    salary  numeric(20,2) CHECK (salary > 0),
12    primary key (VAT));
13 -- all employees are either receptionists, nurses or doctors
14
15 create table phone_number_employee
16    (phone char(10),
17    VAT char(10),
18    primary key(VAT, phone),
19    foreign key(VAT)
20     references employee(VAT) on update cascade on delete cascade);
21
22 create table receptionist
23    (VAT char(10),
24    primary key(VAT),
25    foreign key(VAT)
26     references employee(VAT) on update cascade on delete cascade);
27
28 create table doctor
29    (VAT char(10),
30    specialization varchar(255),
31    biography varchar(255),
32    email varchar(255) unique not null,
33    primary key(VAT),
34    foreign key(VAT)
35     references employee(VAT) on update cascade on delete cascade);
36 -- all doctors are either permanent or trainees
37
38 create table nurse
39    (VAT char(10),
40    primary key(VAT),
41    foreign key(VAT)
42     references employee(VAT) on update cascade on delete cascade);
43
44 create table client
45    (VAT char(10),
```

```
   name varchar(255),
   birth_date DATE,
   street varchar(255),
   city varchar(255),
   zip char(9),
   gender char(2),
   age int CHECK (age>0),
   primary key (VAT));
   --age = (current_date - birth_date).years

create table phone_number_client
   (phone char(10),
   VAT char(10),
   primary key(VAT,phone),
   foreign key(VAT)
    references client(VAT) on update cascade on delete cascade);

create table permanent_doctor
   (VAT char(10),
   primary key(VAT),
   foreign key(VAT)
    references doctor(VAT) on update cascade on delete cascade);

create table trainee_doctor
   (VAT char(10),
   supervisor char(10),
   foreign key(VAT)
    references doctor(VAT) on update cascade on delete cascade,
   foreign key(supervisor)
    references permanent_doctor(VAT) on update cascade on delete cascade,
   primary key (VAT) );

create table supervision_report
   (VAT char(10),
   date_timestamp timestamp,
   description varchar(255),
   evaluation int CHECK (evaluation <=5 AND evaluation >= 1),
   foreign key(VAT)
    references trainee_doctor(VAT) on update no action  on delete no action,
   primary key (VAT, date_timestamp)
   )ENGINE=MyISAM;

create table appointment
   (VAT_doctor char(10),
   date_timestamp timestamp,
   description varchar(255),
   VAT_client char(10),
```

```
93     foreign key(VAT_doctor)
94      references doctor(VAT) on update cascade on delete cascade,
95     foreign key(VAT_client)
96      references client(VAT) on update cascade on delete cascade,
97     primary key (VAT_doctor, date_timestamp));
98
99  create table consultation
100    (VAT_doctor char(10),
101    date_timestamp timestamp,
102    SOAP_S varchar(255),
103    SOAP_O varchar(255),
104    SOAP_A varchar(255),
105    SOAP_P varchar(255),
106    foreign key(VAT_doctor,date_timestamp)
107     references appointment(VAT_doctor, date_timestamp) on update cascade on delete
           no action,
108    primary key (VAT_doctor, date_timestamp)
109    );
110 -- Each primary key of consultation appears on consultation_assistant
111
112 create table consultation_assistant
113    (VAT_doctor char(10),
114    date_timestamp timestamp,
115    VAT_nurse char(10) not null,
116    foreign key(VAT_doctor,date_timestamp)
117     references appointment(VAT_doctor, date_timestamp) on update cascade on delete
           no action,
118    foreign key (VAT_nurse)
119     references nurse(VAT) on update cascade on delete no action,
120    primary key (VAT_doctor, date_timestamp));
121
122 create table diagnostic_code
123    (ID varchar(255),
124    description varchar(255),
125    primary key(ID)
126    );
127
128 create table diagnostic_code_relation
129    (ID1 varchar(255),
130    ID2 varchar(255),
131    type varchar(255),
132    foreign key(ID1)
133     references diagnostic_code(ID) on update cascade on delete cascade ,
134    foreign key(ID2)
135     references diagnostic_code(ID) on update cascade on delete cascade,
136    primary key (ID1, ID2));
137
```

```
138 create table consultation_diagnostic
139    (VAT_doctor char(10),
140    date_timestamp timestamp,
141    ID varchar(255),
142    foreign key (VAT_doctor, date_timestamp)
143     references consultation(VAT_doctor, date_timestamp) on update cascade on delete
            cascade,
144    foreign key (ID)
145     references diagnostic_code(ID) on update cascade on delete cascade,
146    primary key (VAT_doctor, date_timestamp, ID));
147
148 create table medication
149    (name varchar(255),
150    lab varchar(255),
151    primary key(name, lab));
152
153 create table prescription
154    (name varchar(255),
155    lab varchar(255),
156    VAT_doctor char(10),
157    date_timestamp timestamp,
158    ID varchar(255),
159    dosage varchar(255),
160    description varchar(255),
161    foreign key (VAT_doctor, date_timestamp,ID)
162     references consultation_diagnostic(VAT_doctor, date_timestamp,ID) on update
           cascade on delete cascade,
163    foreign key (name,lab)
164     references medication(name,lab) on update cascade on delete cascade,
165    primary key (name, VAT_doctor, date_timestamp, ID));
166
167 create table _procedure
168    (name varchar(255),
169    type varchar (255),
170    primary key (name));
171
172 create table procedure_in_consultation
173    (name varchar(255),
174    VAT_doctor char(10),
175    date_timestamp timestamp,
176    description varchar(255),
177    foreign key (VAT_doctor,date_timestamp)
178     references consultation(VAT_doctor, date_timestamp) on update cascade on delete
            cascade,
179    foreign key (name)
180     references _procedure(name) on update cascade on delete cascade,
181    primary key (name, VAT_doctor, date_timestamp)
```

```
182    );
183
184 create table procedure_radiology
185    (name varchar(255),
186    file varchar(255),
187    VAT_doctor char(10),
188    date_timestamp timestamp,
189    primary key (file, name, VAT_doctor, date_timestamp),
190    foreign key (VAT_doctor,date_timestamp)
191     references consultation_diagnostic(VAT_doctor,date_timestamp) on update cascade
           on delete cascade,
192    foreign key (name)
193     references _procedure(name) on update cascade on delete cascade );
194
195 create table teeth
196    (quadrant char(2),
197    number char(3),
198    name varchar(255),
199    primary key(quadrant, number));
200
201 create table procedure_charting
202    (name varchar(255),
203    VAT char(10),
204    date_timestamp timestamp,
205    quadrant char(2),
206    number char(3),
207    description varchar(255),
208    measure char(5),
209    foreign key(name, VAT, date_timestamp)
210       references procedure_in_consultation(name, VAT_doctor, date_timestamp) on
             update cascade on delete cascade,
211    foreign key (quadrant, number)
212       references teeth(quadrant, number) on update cascade on delete cascade,
213    primary key (name, VAT, date_timestamp, quadrant, number)
214    );
```

Integrity Constrains

Regarding the candidate keys, they have the constrain forcing the to be unique and not null.

All employees must have a positive salary that is checked by $CHECK(salary > 0)$, the employees should have be either a receptionist, doctor or nurse, however this constrain is just a comment on the table as the one saying that all doctors are either trainees or permanent and the one to derive the age of a client from his birth date. To make sure that the evaluation of a supervision report is a number in the range from 1 to 5 it is done $CHECK(evaluation <= 5 AND evaluation >= 1)$, the type of the value is integer.

To ensure that consultations are always assigned to at least one assistant nurse it is a comment on the consultation table and on the *consultation_assistant* the value of *VAT_nurse* cannot be

null. With this table format it is not possible to have more than one nurse per consultation since on the table *consultation_assistant* there would be two entries with the same key, to surpass this *VAT_nurse* would have to be part of the primary key.

3 Populate

To populate the database it were used the commands similar as the ones shown on the listing 3. A list of all the insertions is presented on appendix A.

Listing 3:

```
1  insert into employee values ('123456789', 'Jane Sweettooth','1990/12/17', 'rua', '
       cidade', '2780-255', 'PT50567891234567891234567', 900);
2
3  insert into phone_number_employee values ('912345678','123456789');
4
5  insert into receptionist values ('123457469');
6
7  insert into doctor values ('123456789', 'Expert em caries','Boa aluna, mas pessima
       a tirar sisos','Jane@bluetooth.com');
8
9  insert into nurse values ('123746789');
10
11 insert into client values ('999999999', 'Jose Bebe', '1990/12/17', 'rua3', 'cidade1
       ', '2780-255', 'M' , 26);
12
13 insert into phone_number_client values ('912345878','999999999');
14
15 insert into permanent_doctor values ('123456789');
16
17 insert into supervision_report values ('987654321', '2018/12/17', 'Boa moca a Julia
       ', 4);
18
19 insert into appointment values ('123456789', '2019/11/17 17:00:00', 'rotina', '
       999999999');
20
21 insert into consultation values ('123456789', '2019/11/17 17:00:00', 's', '
       gingivitis', 'a', 'p' );
22
23 insert into consultation_assistant values ('123456789', '2019/11/17 17:00:00', '
       123746789' );
24
25 insert into diagnostic_code values ('D105','constipacao dental');
26
27 insert into diagnostic_code_relation values ('D105','D106','dor aguda');
28
```

```
29 insert into consultation_diagnostic values ('123456789', '2019/11/17 17:00:00','D12
       ');
30
31 insert into medication values ('medication1','lab1');
32
33 insert into prescription values ('medication1','lab1','987654321', '2019/12/17
       17:00:00','D12','4 em 4 horas','nao esquecer');
34
35 insert into _procedure values ('d4 charting', 'dental charting');
36
37 insert into procedure_in_consultation values ('d4 charting', '123456789', '
       2019/11/17 17:00:00', 'arrancar');
38
39 insert into procedure_radiology values ('leg radiography','file1', '987656789', '
       2019/11/17 17:00:00');
40
41 insert into teeth values ('1','2','dente2');
42
43 insert into procedure_charting values ('d4 charting', '123456789', '2019/11/17
       17:00:00','1','1','jabcw','2');
```

4 Queries

4.1

In this query, there are tables whose keys are composed by more than one argument (appointment and consultation) that's why there are two comparisons between the two tables. This operation could also be done with a join.

Listing 4:

```
1 select distinct client.VAT, client.name, phone_number_client.phone
2 from client, consultation, employee, appointment , phone_number_client
3 where employee.name = "Jane Sweettooth" and employee.VAT = appointment.VAT_doctor
4 and appointment.date_timestamp = consultation.date_timestamp
5 and appointment.VAT_doctor=consultation.VAT_doctor  and appointment.VAT_client =
      client.VAT
6 and client.VAT  = phone_number_client.VAT
7 order by client.name;
```

4.2

In this query it was needed to read the same table two times pointing to two different values at the same time. That is why the table employee appears two times on the from of the query.

Listing 5:

```
(select emp_t.name as name_trainee, emp_t.VAT as VAT_trainee ,emp_d.name as
   name_doctor , emp_d.VAT as VAT_doctor, supervision_report.evaluation,
   supervision_report.description
from employee as emp_t, supervision_report, employee as emp_d, trainee_doctor
where emp_t.VAT = supervision_report.VAT and trainee_doctor.supervisor=emp_d.VAT
and supervision_report.evaluation < 3)
union
(select emp_t.name as name_trainee, emp_t.VAT as VAT_trainee ,emp_d.name as
   name_doctor , emp_d.VAT as VAT_doctor, supervision_report.evaluation,
   supervision_report.description
from employee as emp_t, supervision_report, employee as emp_d, trainee_doctor
where emp_t.VAT = supervision_report.VAT and trainee_doctor.supervisor=emp_d.VAT
   and supervision_report.description like '%insufficient%' );
```

4.3

The following query has a comparison to choose the most recent consultation of each client. Among these consultation there are selected the ones with the searching SOAP_O.

Listing 6:

```
select distinct client.name, client.city, client.VAT, consultation.SOAP_O
from client, appointment, consultation
where appointment.VAT_client = client.VAT and consultation.date_timestamp=
   appointment.date_timestamp
and appointment.VAT_doctor = consultation.VAT_doctor and (consultation.SOAP_O like
   '%gingivitis%' or consultation.SOAP_O like '%periodontitis%')
and consultation.date_timestamp >= all (
select consultation.date_timestamp
from consultation, appointment
where appointment.VAT_doctor = consultation.VAT_doctor and consultation.
   date_timestamp=appointment.date_timestamp
and client.VAT = appointment.VAT_client );
```

4.4

On this one, it was used a membership relation between two tables to check the appointments there were not in consultations.

Listing 7:

```
select client.name , client.VAT, client.street, client.city, client.zip
```

```
from client, appointment
where client.VAT = appointment.VAT_client and appointment.date_timestamp not in
(select consultation.date_timestamp from consultation);
```

4.5

On this example, the count is associated with a name (counter) with the purpose of using it to order the query result according to it.

Listing 8:

```
select diagnostic_code.ID, diagnostic_code.description, count( distinct
   prescription.name) as counter
from diagnostic_code, consultation_diagnostic, prescription
where  prescription.ID = diagnostic_code.ID and consultation_diagnostic.VAT_doctor=
   prescription.VAT_doctor and consultation_diagnostic.date_timestamp =
   prescription.date_timestamp and consultation_diagnostic.ID = prescription.ID
group by prescription.ID
order by counter asc
```

4.6

For Query 6, it is asked to do several averages of counts. In order to implement this, each count is computed on different tables built inside this query. The tables for each count are similar, they are compared to each consultation, in order to group entries belonging to the same consultation. In other words, each count represents the number of entries that belong to the same consultation, and then the average is done based on this numbers.

Query 6 was developed considering that each consultation could have more than one assistant nurse, nevertheless, with the proposed relational model, this doesn't happen, each consultation has only one nurse assign since *VAT_nurse* is not included on the primary key of the *consultation_assistant* table. The result is that the average for the nurses will be always 1.

Listing 9:

```
(select avg(nurses), avg(procedures), avg(diagnosis), avg(prescriptions) from
(select count(nurse.n) as nurses from
(select consultation_assistant.VAT_nurse as n, consultation_assistant.VAT_doctor as
    d, consultation_assistant.date_timestamp as t
from consultation_assistant, appointment, client
where appointment.date_timestamp = consultation_assistant.date_timestamp
and appointment.VAT_doctor = consultation_assistant.VAT_doctor
and extract(year from consultation_assistant.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age < 18) as nurse
group by nurse.d, nurse.t) as n1,
(select count(proc.n) as procedures from
```

```
(select procedure_in_consultation.name as n, procedure_in_consultation.VAT_doctor
   as d, procedure_in_consultation.date_timestamp as t
from procedure_in_consultation, appointment, client
where appointment.date_timestamp = procedure_in_consultation.date_timestamp
and appointment.VAT_doctor = procedure_in_consultation.VAT_doctor
and extract(year from procedure_in_consultation.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age < 18) as proc
group by proc.d, proc.t) as n2,
(select count(diagnostic.ID) as diagnosis from
(select consultation_diagnostic.ID as ID,
consultation_diagnostic.VAT_doctor as d, consultation_diagnostic.date_timestamp as
   t
from consultation_diagnostic, appointment, client
where appointment.date_timestamp = consultation_diagnostic.date_timestamp
and appointment.VAT_doctor = consultation_diagnostic.VAT_doctor
and extract(year from consultation_diagnostic.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age < 18) as diagnostic
group by diagnostic.d, diagnostic.t) as n3,
(select count(distinct concat(presc.n, presc.l)) as prescriptions from
(select prescription.name as n, prescription.lab as l,
prescription.VAT_doctor as d, prescription.date_timestamp as t
from prescription, appointment, client
where appointment.date_timestamp = prescription.date_timestamp
and appointment.VAT_doctor = prescription.VAT_doctor
and extract(year from prescription.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age < 18) as presc
group by presc.d, presc.t) as n4)
union all
(select avg(nurses), avg(procedures), avg(diagnosis), avg(prescriptions) from
(select count(nurse.n) as nurses from
(select consultation_assistant.VAT_nurse as n, consultation_assistant.VAT_doctor as
    d, consultation_assistant.date_timestamp as t
from consultation_assistant, appointment, client
where appointment.date_timestamp = consultation_assistant.date_timestamp
and appointment.VAT_doctor = consultation_assistant.VAT_doctor
and extract(year from consultation_assistant.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age >= 18) as nurse
group by nurse.d, nurse.t) as n1,
(select count(proc.n) as procedures from
(select procedure_in_consultation.name as n, procedure_in_consultation.VAT_doctor
   as d, procedure_in_consultation.date_timestamp as t
from procedure_in_consultation, appointment, client
where appointment.date_timestamp = procedure_in_consultation.date_timestamp
and appointment.VAT_doctor = procedure_in_consultation.VAT_doctor
and extract(year from procedure_in_consultation.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age >= 18) as proc
group by proc.d, proc.t) as n2,
```

```
(select count(diagnostic.ID) as diagnosis from
(select consultation_diagnostic.ID as ID,
consultation_diagnostic.VAT_doctor as d, consultation_diagnostic.date_timestamp as
   t
from consultation_diagnostic, appointment, client
where appointment.date_timestamp = consultation_diagnostic.date_timestamp
and appointment.VAT_doctor = consultation_diagnostic.VAT_doctor
and extract(year from consultation_diagnostic.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age >= 18) as diagnostic
group by diagnostic.d, diagnostic.t) as n3,
(select count(distinct concat(presc.n, presc.l)) as prescriptions from
(select prescription.name as n, prescription.lab as l,
prescription.VAT_doctor as d, prescription.date_timestamp as t
from prescription, appointment, client
where appointment.date_timestamp = prescription.date_timestamp
and appointment.VAT_doctor = prescription.VAT_doctor
and extract(year from prescription.date_timestamp) = '2019'
and client.VAT = appointment.VAT_client and client.age >= 18) as presc
group by presc.d, presc.t) as n4);
```

4.7

On this query, it is used two different names for prescription because information about the first search (p) was needed to filter the second one (p2).

Listing 10:

```
select p.ID, p.name, p.lab
from prescription as p
group by p.name
having count(p.name) >= all (
select count(p2.name)
from prescription as p2
where p2.ID = p.ID
group by p2.name )
```

4.8

On the query presented on the listing below it is used the function extract to compare the year from the timestamp with 2019. It is also used the *not in* to filter the results with the description mentioning infectious diseases. On this case the function extract would be more suitable, however *MYSQL* does not support it.

Listing 11:

```
select prescription.name , prescription.lab
from prescription, diagnostic_code
where prescription.ID = diagnostic_code.ID and extract(year from prescription.
   date_timestamp)='2019'
and diagnostic_code.description like '%dental cavities%' and (prescription.name,
   prescription.lab) not in
(select prescription.name , prescription.lab
from prescription, diagnostic_code
where prescription.ID = diagnostic_code.ID and extract(year from prescription.
   date_timestamp)='2019'
and diagnostic_code.description like '%infectious disease%')
group by prescription.name
order by prescription.name;
```

4.9

For this query it was asked to apply everything that was done before all on the same query.

Listing 12:

```
select distinct client.name , client.street, client.city, client.zip
from client, appointment
where client.VAT = appointment.VAT_client and extract(year from appointment.
   date_timestamp) = '2019'
and (client.name , client.street, client.city, client.zip) not in (
select client.name , client.street, client.city, client.zip
from client, appointment
where client.VAT = appointment.VAT_client and not exists (
        select 1 from appointment, consultation where appointment.VAT_doctor  =
           consultation.VAT_doctor
        and appointment.date_timestamp=consultation.date_timestamp)
)
```

5 Indexes

Regarding the first query, it could be used a index to have a better performance searching the doctor on the consultation table. On this query, the same search is done over and over again, for this reason, the most suitable index method would be a hash function, since hash functions work better for equality constrains.

On the second query, there are two index that can be added to increase its perform, one to the evaluation and other to the description. As far as the evaluation is concern, the search that is done is for more than one value (evaluation < 3) and the hash function is not adequate for these cases, the solution then is the B-Tree. Concerning the description, as it is a search of a word(s) among

sentences the most suitable index method is the inverted index, which will assign words as indexes to the entries where this word appears, in *MYSQL* this index is named as "*fulltext*".

All this indexes can be shown on the listing below.

Listing 13:

```
create index id_1 on consultation (VAT_doctor) using hash;

create index id_score on supervision_report (evaluation) using BTREE;

create fulltext index idx on supervision_report(description);
```

6 Changes

6.1

Listing 14:

```
update employee
set street = 'rua rovisco pais',city='lisboa',zip='0987'
where name = 'Jane Sweettooth';
```

6.2

Listing 15:

```
update employee
set salary = salary*1.05
where VAT in (
select VAT_doctor
from appointment
where extract(year from appointment.date_timestamp)='2019'
having count(VAT_doctor) >= 100  )
```

6.3

Listing 16:

```
delete from employee
where name = 'Jane Sweettooth'
```

6.4

Listing 17:

```
insert into diagnostic_code values ('D13','periodontitis');
update consultation_diagnostic
set ID = 'D13'
where (VAT_doctor,date_timestamp) in (
select pc.VAT, pc.date_timestamp
from  procedure_charting as pc
where pc.name='d4 charting'
group by pc.name
having avg(pc.measure)>4
);
```

All the changes to the database were trivial, the most difficult one is the one on the listing 17 where the group by function has to be there to be possible to calculate the average.

7 Views

7.1 dim_date

Listing 18:

```
create view dim_date as
select distinct date_timestamp "date", extract(day from date_timestamp) "day",
extract(month from date_timestamp) "month",
extract(year from date_timestamp) "year"
from consultation;
```

7.2 dim_client

Listing 19:

```
create view dim_client as
select VAT, gender, age
from client;
```

7.3 dim_location_client

Listing 20:

```
create view dim_location_client as
select distinct zip, city
from client;
```

The only difference between this first 3 views has to do with the keys of each one of them. The view *dim_ client* uses as primary key, a foreign key, thus it does not need the "distinct" restriction when selecting its elements. On the other two views if the "distinct" restriction is dropped, they may end up with repeated primary keys, which is not acceptable.

7.4 facts_consults

Listing 21:

```
create view facts_consults as
select dc.VAT "VAT", dd.date_timestamp "date", dlc.zip "zip",
count(distinct procedure_in_consultation.name) "num_procedures",
count(distinct prescription.name) "num_medications",
count(distinct consultation_diagnostic.ID) "num_diagnostic_codes"
from dim_client dc, dim_date dd, dim_location_client dlc,
procedure_in_consultation, prescription, consultation_diagnostic,
appointment, client, consultation
where dc.VAT = appointment.VAT_client
and dc.VAT = client.VAT
and dlc.zip = client.zip
and dd.date_timestamp = consultation.date_timestamp
and consultation.VAT_doctor = appointment.VAT_doctor
and dd.date_timestamp = appointment.date_timestamp
and dd.date_timestamp = procedure_in_consultation.date_timestamp
and dd.date_timestamp = prescription.date_timestamp
and dd.date_timestamp = consultation_diagnostic.date_timestamp
and appointment.VAT_doctor = procedure_in_consultation.VAT_doctor
and appointment.VAT_doctor = prescription.VAT_doctor
and appointment.VAT_doctor = consultation_diagnostic.VAT_doctor
and prescription.ID = consultation_diagnostic.ID
group by dc.VAT, dd.date_timestamp;
```

In *facts_ consults*, since it is required to search information in several different tables/views, there is a need to relate them. The *appointment* table provides the relation between each client and the date of the consultation and then it's easy to extract the procedures/prescriptions/diagnostics involved (there is the need to go through the *consultation* table, since the database does not guarantees that there are no clients which have made more than one appointment for the same date, with different doctors).

Each count has to have the "distinct" restriction, because this view has a lot of relations which may create repeated entries, thus, without the "distinct" it would count more entries then the ones

it was supposed to.

A Populate Tables

Listing 22: Populating tables

```
1
2    --            employees
3  insert into employee values ('123456789', 'Jane Sweettooth', '1990/12/17', 'rua', '
     cidade', '2780-255', 'PT5056789123456789123456 7', 900);
4  insert into employee values ('987654321', 'Julia Sweettooth', '1990/12/17', 'rua2',
      'cidade2', '2780-260', 'PT5056789123123489123456 7', 600);
5  insert into employee values ('123746789', 'Jane Dentedoce', '1980/12/17', 'rua2', '
     cidade3', '2770-255', 'PT5056789673456789123456 7', 1000);
6  insert into employee values ('987656789', 'Julio Isidro', '1200/12/17', 'rua', '
     cidade2', '2780-485', 'PT5012349123456789123466 6', 666.80);
7  insert into employee values ('123458889', 'Joao Baiao', '1805/12/17', 'rua7', '
     cidade3', '2780-777', 'PT5056789123456789543216 7', 9600);
8  insert into employee values ('123457469', 'Sara Rececao', '1254/12/17', 'rua6', '
     cidade74', '2740-777', 'PT5056689127456780543216 7', 100);
9
10     --  phone_number_employee
11 insert into phone_number_employee values ('912345678','123456789');
12 insert into phone_number_employee values ('962345678','987654321');
13 insert into phone_number_employee values ('962354678','123746789');
14 insert into phone_number_employee values ('953414378','987656789');
15 insert into phone_number_employee values ('998878574','123458889');
16
17
18 --              receptionist
19 insert into receptionist values ('123457469');
20
21
22 --              doctors
23 insert into doctor values ('123456789', 'Expert em caries', 'Boa aluna, mas pessima
      a tirar sisos','Jane@bluetooth.com');
24 insert into doctor values ('987654321', 'Expert em sisos', '3 meses na clinica e ja
      se fartou','Julia@bluetooth.com');
25 insert into doctor values ('987656789', 'Expert em piropos', 'Trabalha pouco fala
     muito','Julio@bluetooth.com');
26
27 --    nurses
28 insert into nurse values ('123746789');
29
30 --              clients
31 insert into client values ('999999999', 'Jose Bebe', '1990/12/17', 'rua3', 'cidade1
     ', '2780-255', 'M' , 26);
32 insert into client values ('888888888', 'Hugo Burro', '1980/12/17', 'rua1', '
     cidade5', '2780-255', 'M' , 26);
```

```
33 insert into client values ('777777777', 'Pedro Cebo', '1890/12/17', 'rua1', '
      cidade5', '2780-255', 'M' , 26);
34 insert into client values ('666666666', 'Filipe Bibe', '1890/12/17', 'rua1', '
      cidade5', '2780-255', 'M' , 26);
35
36    --    phone_number_client
37 insert into phone_number_client values ('912345878','999999999');
38 insert into phone_number_client values ('962345978','999999999');
39
40 --              permanent_doctors
41 insert into permanent_doctor values ('123456789');
42
43 --              trainee_doctors
44 insert into trainee_doctor values ('987654321','123456789');
45 insert into trainee_doctor values ('987656789', '123456789');
46
47 --              supervision_reports
48 insert into supervision_report values ('987654321', '2018/12/17', 'Boa moca a Julia
      ', 4);
49 insert into supervision_report values ('987656789', '2018/12/17', 'Mais piropos',
      2);
50 insert into supervision_report values ('987656789', '2017/12/17', 'insufficient',
      3);
51
52 --    appointments
53 insert into appointment values ('123456789', '2019/11/17 17:00:00', 'rotina', '
      999999999');
54 insert into appointment values ('123456789', '2019/12/17 17:00:00', 'follow-up', '
      999999999');
55 insert into appointment values ('987654321', '2019/11/17 17:00:00', 'rotina', '
      888888888');
56 insert into appointment values ('987654321', '2019/12/17 17:00:00', 'follow-up', '
      888888888');
57 insert into appointment values ('987656789', '2019/11/17 17:00:00', 'rotina', '
      777777777');
58 insert into appointment values ('987656789', '2019/12/17 17:00:00', 'follow-up', '
      777777777');
59 insert into appointment values ('987656789', '2019/10/17 17:00:00', 'follow-up', '
      666666666');
60 insert into appointment values ('123456789', '2019-7-13 16:38:00', 'follow-up', '
      777777777');
61 insert into appointment values ('123456789', '2019-12-25 13:58:00', 'rotina', '
      999999999');
62 insert into appointment values ('123456789', '2019-3-30 13:54:00', 'rotina', '
      888888888');
63 insert into appointment values ('123456789', '2019-1-13 14:46:00', 'rotina', '
      888888888');
```

```
64 insert into appointment values ('123456789', '2019-5-1 14:51:00', 'tratamento', '
       666666666');
65 insert into appointment values ('123456789', '2019-8-30 13:27:00', 'tratamento', '
       999999999');
66 insert into appointment values ('123456789', '2019-9-19 17:29:00', 'rotina', '
       777777777');
67 insert into appointment values ('123456789', '2019-9-26 13:42:00', 'rotina', '
       777777777');
68 insert into appointment values ('123456789', '2019-6-29 14:3:00', 'tratamento', '
       888888888');
69 insert into appointment values ('123456789', '2019-7-17 10:35:00', 'tratamento', '
       888888888');
70 insert into appointment values ('123456789', '2019-7-12 10:7:00', 'tratamento', '
       999999999');
71 insert into appointment values ('123456789', '2019-9-20 11:54:00', 'tratamento', '
       777777777');
72 insert into appointment values ('123456789', '2019-2-26 10:24:00', 'follow-up', '
       888888888');
73 insert into appointment values ('123456789', '2019-6-19 11:13:00', 'tratamento', '
       888888888');
74 insert into appointment values ('123456789', '2019-10-28 18:17:00', 'tratamento', '
       888888888');
75 insert into appointment values ('123456789', '2019-10-20 16:10:00', 'tratamento', '
       999999999');
76 insert into appointment values ('123456789', '2019-7-22 13:31:00', 'follow-up', '
       888888888');
77 insert into appointment values ('123456789', '2019-6-17 17:52:00', 'follow-up', '
       888888888');
78 insert into appointment values ('123456789', '2019-1-2 12:27:00', 'tratamento', '
       999999999');
79 insert into appointment values ('123456789', '2019-6-1 13:47:00', 'follow-up', '
       999999999');
80 insert into appointment values ('123456789', '2019-11-26 17:13:00', 'tratamento', '
       888888888');
81 insert into appointment values ('123456789', '2019-2-9 15:15:00', 'tratamento', '
       777777777');
82 insert into appointment values ('123456789', '2019-3-30 15:43:00', 'rotina', '
       999999999');
83 insert into appointment values ('123456789', '2019-5-17 11:15:00', 'rotina', '
       777777777');
84 insert into appointment values ('123456789', '2019-2-12 12:2:00', 'rotina', '
       666666666');
85 insert into appointment values ('123456789', '2019-8-10 18:9:00', 'tratamento', '
       999999999');
86 insert into appointment values ('123456789', '2019-11-13 11:29:00', 'tratamento', '
       666666666');
87 insert into appointment values ('123456789', '2019-6-27 17:35:00', 'follow-up', '
```

```
   999999999');
88 insert into appointment values ('123456789', '2019-5-20 17:17:00', 'rotina', '
   777777777');
89 insert into appointment values ('123456789', '2019-9-4 12:42:00', 'follow-up', '
   777777777');
90 insert into appointment values ('123456789', '2019-3-21 10:54:00', 'rotina', '
   777777777');
91 insert into appointment values ('123456789', '2019-11-12 16:17:00', 'tratamento', '
   666666666');
92 insert into appointment values ('123456789', '2019-8-28 14:40:00', 'follow-up', '
   999999999');
93 insert into appointment values ('123456789', '2019-2-9 15:16:00', 'rotina', '
   666666666');
94 insert into appointment values ('123456789', '2019-8-4 15:42:00', 'follow-up', '
   666666666');
95 insert into appointment values ('123456789', '2019-7-4 16:12:00', 'rotina', '
   777777777');
96 insert into appointment values ('123456789', '2019-10-16 17:6:00', 'tratamento', '
   666666666');
97 insert into appointment values ('123456789', '2019-2-27 16:48:00', 'follow-up', '
   666666666');
98 insert into appointment values ('123456789', '2019-5-6 14:1:00', 'follow-up', '
   888888888');
99 insert into appointment values ('123456789', '2019-5-1 18:24:00', 'tratamento', '
   666666666');
100 insert into appointment values ('123456789', '2019-10-16 10:37:00', 'rotina', '
   888888888');
101 insert into appointment values ('123456789', '2019-12-17 11:15:00', 'follow-up', '
   666666666');
102 insert into appointment values ('123456789', '2019-4-30 18:55:00', 'follow-up', '
   777777777');
103 insert into appointment values ('123456789', '2019-9-13 11:35:00', 'rotina', '
   666666666');
104 insert into appointment values ('123456789', '2019-9-30 10:3:00', 'follow-up', '
   777777777');
105 insert into appointment values ('123456789', '2019-6-6 14:35:00', 'rotina', '
   777777777');
106 insert into appointment values ('123456789', '2019-9-21 18:4:00', 'tratamento', '
   777777777');
107 insert into appointment values ('123456789', '2019-9-1 13:47:00', 'tratamento', '
   777777777');
108 insert into appointment values ('123456789', '2019-3-26 12:49:00', 'rotina', '
   999999999');
109 insert into appointment values ('123456789', '2019-12-7 14:29:00', 'tratamento', '
   888888888');
110 insert into appointment values ('123456789', '2019-2-15 12:34:00', 'tratamento', '
   888888888');
```

```
111 insert into appointment values ('123456789', '2019-4-18 14:11:00', 'rotina', '
    666666666');
112 insert into appointment values ('123456789', '2019-1-10 17:41:00', 'tratamento', '
    888888888');
113 insert into appointment values ('123456789', '2019-10-23 15:36:00', 'rotina', '
    888888888');
114 insert into appointment values ('123456789', '2019-7-26 10:42:00', 'rotina', '
    999999999');
115 insert into appointment values ('123456789', '2019-4-22 17:53:00', 'rotina', '
    999999999');
116 insert into appointment values ('123456789', '2019-9-17 14:51:00', 'tratamento', '
    666666666');
117 insert into appointment values ('123456789', '2019-3-2 13:53:00', 'follow-up', '
    666666666');
118 insert into appointment values ('123456789', '2019-3-8 18:19:00', 'tratamento', '
    999999999');
119 insert into appointment values ('123456789', '2019-12-16 14:13:00', 'rotina', '
    888888888');
120 insert into appointment values ('123456789', '2019-5-27 15:17:00', 'follow-up', '
    666666666');
121 insert into appointment values ('123456789', '2019-5-13 14:53:00', 'tratamento', '
    666666666');
122 insert into appointment values ('123456789', '2019-11-7 18:4:00', 'rotina', '
    999999999');
123 insert into appointment values ('123456789', '2019-1-19 17:55:00', 'tratamento', '
    666666666');
124 insert into appointment values ('123456789', '2019-5-29 10:0:00', 'rotina', '
    666666666');
125 insert into appointment values ('123456789', '2019-8-24 18:40:00', 'follow-up', '
    999999999');
126 insert into appointment values ('123456789', '2019-1-8 15:25:00', 'follow-up', '
    777777777');
127 insert into appointment values ('123456789', '2019-10-10 14:38:00', 'follow-up', '
    666666666');
128 insert into appointment values ('123456789', '2019-5-22 15:18:00', 'rotina', '
    888888888');
129 insert into appointment values ('123456789', '2019-7-20 14:22:00', 'follow-up', '
    888888888');
130 insert into appointment values ('123456789', '2019-8-7 12:0:00', 'rotina', '
    999999999');
131 insert into appointment values ('123456789', '2019-10-29 18:12:00', 'tratamento', '
    666666666');
132 insert into appointment values ('123456789', '2019-11-14 17:23:00', 'follow-up', '
    777777777');
133 insert into appointment values ('123456789', '2019-7-28 13:27:00', 'rotina', '
    888888888');
134 insert into appointment values ('123456789', '2019-3-29 10:14:00', 'follow-up', '
```

```
    999999999');
135 insert into appointment values ('123456789', '2019-1-9 12:1:00', 'follow-up', '
    777777777');
136 insert into appointment values ('123456789', '2019-4-23 15:10:00', 'tratamento', '
    999999999');
137 insert into appointment values ('123456789', '2019-12-9 16:26:00', 'follow-up', '
    777777777');
138 insert into appointment values ('123456789', '2019-3-24 13:5:00', 'tratamento', '
    666666666');
139 insert into appointment values ('123456789', '2019-5-21 18:9:00', 'tratamento', '
    999999999');
140 insert into appointment values ('123456789', '2019-5-27 17:0:00', 'follow-up', '
    777777777');
141 insert into appointment values ('123456789', '2019-4-2 13:27:00', 'rotina', '
    999999999');
142 insert into appointment values ('123456789', '2019-4-4 17:12:00', 'follow-up', '
    999999999');
143 insert into appointment values ('123456789', '2019-10-1 13:34:00', 'rotina', '
    666666666');
144 insert into appointment values ('123456789', '2019-5-22 14:3:00', 'follow-up', '
    999999999');
145 insert into appointment values ('123456789', '2019-10-9 17:0:00', 'rotina', '
    888888888');
146 insert into appointment values ('123456789', '2019-6-22 15:7:00', 'follow-up', '
    888888888');
147 insert into appointment values ('123456789', '2019-10-6 17:46:00', 'tratamento', '
    777777777');
148 insert into appointment values ('123456789', '2019-11-12 16:28:00', 'rotina', '
    888888888');
149 insert into appointment values ('123456789', '2019-2-21 14:18:00', 'tratamento', '
    999999999');
150 insert into appointment values ('123456789', '2019-9-21 16:35:00', 'tratamento', '
    999999999');
151 insert into appointment values ('123456789', '2019-2-7 17:19:00', 'rotina', '
    777777777');
152 insert into appointment values ('123456789', '2019-6-26 18:42:00', 'follow-up', '
    888888888');
153 insert into appointment values ('123456789', '2019-3-5 12:52:00', 'follow-up', '
    777777777');
154 insert into appointment values ('123456789', '2019-3-14 14:24:00', 'follow-up', '
    666666666');
155 insert into appointment values ('123456789', '2019-9-29 16:5:00', 'rotina', '
    666666666');
156 insert into appointment values ('123456789', '2019-1-17 18:13:00', 'tratamento', '
    777777777');
157 insert into appointment values ('123456789', '2019-9-13 17:32:00', 'rotina', '
    999999999');
```

```
158 insert into appointment values ('123456789', '2019-4-15 11:55:00', 'follow-up', '
    666666666');
159 insert into appointment values ('123456789', '2019-4-28 15:57:00', 'tratamento', '
    999999999');
160 insert into appointment values ('123456789', '2019-12-26 13:45:00', 'rotina', '
    999999999');
161 insert into appointment values ('123456789', '2019-10-10 13:12:00', 'follow-up', '
    777777777');
162 insert into appointment values ('123456789', '2019-3-4 13:32:00', 'tratamento', '
    666666666');
163 insert into appointment values ('123456789', '2019-4-13 11:58:00', 'follow-up', '
    888888888');
164 insert into appointment values ('123456789', '2019-12-15 16:38:00', 'tratamento', '
    666666666');
165 insert into appointment values ('123456789', '2019-9-20 16:15:00', 'tratamento', '
    777777777');
166 insert into appointment values ('123456789', '2019-2-16 11:48:00', 'rotina', '
    999999999');
167 insert into appointment values ('123456789', '2019-7-12 15:19:00', 'tratamento', '
    666666666');
168 insert into appointment values ('123456789', '2019-6-13 12:51:00', 'rotina', '
    777777777');
169 insert into appointment values ('123456789', '2019-2-13 11:54:00', 'tratamento', '
    777777777');
170
171 --    consultations
172 insert into consultation values ('123456789', '2019/11/17 17:00:00', 's', '
    gingivitis', 'a', 'p' );
173 insert into consultation values ('123456789', '2019/12/17 17:00:00', 's', '
    periodontitis', 'a', 'p' );
174 insert into consultation values ('987654321', '2019/12/17 17:00:00', 's', '
    gingivitis', 'a', 'p' );
175 insert into consultation values ('987654321', '2019/11/17 17:00:00', 's', '
    periodontitis', 'a', 'p' );
176 insert into consultation values ('987656789', '2019/11/17 17:00:00', 's', 'o', 'a',
     'p' );
177 insert into consultation values ('987656789', '2019/12/17 17:00:00', 's', 'o', 'a',
     'p' );
178
179 --    consultation_assistants
180 insert into consultation_assistant values ('123456789', '2019/11/17 17:00:00', '
    123746789' );
181 insert into consultation_assistant values ('123456789', '2019/12/17 17:00:00', '
    123746789' );
182 insert into consultation_assistant values ('987654321', '2019/11/17 17:00:00', '
    123746789' );
183 insert into consultation_assistant values ('987654321', '2019/12/17 17:00:00', '
```

```
    123746789' );
184 insert into consultation_assistant values ('987656789', '2019/11/17 17:00:00', '
    123746789' );
185 insert into consultation_assistant values ('987656789', '2019/12/17 17:00:00', '
    123746789' );
186
187 -- diagnostic_code
188 insert into diagnostic_code values ('D105','constipacao dental');
189 insert into diagnostic_code values ('D106','dor no dente');
190 insert into diagnostic_code values ('D200','dentes tortos');
191 insert into diagnostic_code values ('D204','dentes muito tortos');
192 insert into diagnostic_code values ('D000','esta a fingir');
193 insert into diagnostic_code values ('D501','infectious disease');
194 insert into diagnostic_code values ('D502','dental cavities');
195 insert into diagnostic_code values ('D12','gingivitis');
196
197
198 -- diagnostic_code_relation
199
200 insert into diagnostic_code_relation values ('D105','D106','dor aguda');
201 insert into diagnostic_code_relation values ('D200','D204','aparelho');
202
203 -- consultation_diagnostic
204
205 insert into consultation_diagnostic values ('123456789', '2019/11/17 17:00:00','D12
    ');
206 insert into consultation_diagnostic values ('987654321', '2019/12/17 17:00:00','D12
    ');
207 insert into consultation_diagnostic values ('987656789', '2019/11/17 17:00:00','
    D204');
208 insert into consultation_diagnostic values ('987654321', '2019/11/17 17:00:00','D12
    ');
209 insert into consultation_diagnostic values ('123456789', '2019/12/17 17:00:00','D12
    ');
210 insert into consultation_diagnostic values ('987656789', '2019/12/17 17:00:00','
    D502');
211
212 -- medication
213
214 insert into medication values ('palmada','mae');
215 insert into medication values ('medication1','lab1');
216 insert into medication values ('medication2','lab1');
217
218 -- prescription
219
220 insert into prescription values ('palmada','mae','123456789', '2019/11/17 17:00:00'
    ,'D12','qdo se porta mal','bem dado');
```

```
221 insert into prescription values ('medication1','lab1','987654321', '2019/12/17
       17:00:00','D12','4 em 4 horas','nao esquecer');
222 insert into prescription values ('medication2','lab1','987656789', '2019/11/17
       17:00:00','D204','2 em 2 horas','nao esquecer');
223 insert into prescription values ('medication2','lab1','987654321','2019/11/17
       17:00:00','D12','2 em 2 horas','nao esquecer');
224 insert into prescription values ('medication2','lab1','123456789', '2019/12/17
       17:00:00','D12','2 em 2 horas','nao esquecer');
225 insert into prescription values ('medication1','lab1','987656789', '2019/12/17
       17:00:00','D502','2 em 2 horas','nao esquecer');
226
227 -- procedure
228
229 insert into _procedure values ('d4 charting', 'dental charting');
230 insert into _procedure values ('leg radiography', 'x-ray');
231 insert into _procedure values ('arm radiography', 'x-ray');
232
233 -- procedure_in_consultation
234
235 insert into procedure_in_consultation values ('d4 charting', '123456789', '
       2019/11/17 17:00:00', 'arrancar');
236 insert into procedure_in_consultation values ('d4 charting', '123456789', '
       2019/12/17 17:00:00', 'arrancar');
237 insert into procedure_in_consultation values ('leg radiography', '987656789', '
       2019/11/17 17:00:00', 'fotografar');
238 insert into procedure_in_consultation values ('leg radiography', '123456789', '
       2019/11/17 17:00:00', 'correu mal');
239 insert into procedure_in_consultation values ('leg radiography', '123456789', '
       2019/12/17 17:00:00', 'partido');
240 insert into procedure_in_consultation values ('arm radiography', '123456789', '
       2019/11/17 17:00:00', 'fraturado');
241
242 -- procedure radiology
243
244 insert into procedure_radiology values ('leg radiography','file1', '987656789', '
       2019/11/17 17:00:00');
245 insert into procedure_radiology values ('leg radiography','file 2', '123456789', '
       2019/11/17 17:00:00');
246 insert into procedure_radiology values ('leg radiography','file 3', '123456789', '
       2019/11/17 17:00:00');
247 insert into procedure_radiology values ('arm radiography','file4', '123456789', '
       2019/11/17 17:00:00');
248
249
250 -- teeth
251 insert into teeth values ('1','1','dente1');
252 insert into teeth values ('1','2','dente2');
```

```
253 insert into teeth values ('1','3','dente3');
254 insert into teeth values ('2','1','dente1');
255 insert into teeth values ('2','2','dente2');
256 insert into teeth values ('2','3','dente3');
257
258
259 -- procedure charting
260 insert into procedure_charting values ('d4 charting', '123456789', '2019/11/17
       17:00:00','1','1','jabcw','2');
261 insert into procedure_charting values ('d4 charting', '123456789', '2019/11/17
       17:00:00','1','2','ajc','5');
262 insert into procedure_charting values ('d4 charting', '123456789', '2019/11/17
       17:00:00','1','3','ajc','10');
263 insert into procedure_charting values ('d4 charting', '123456789', '2019/12/17
       17:00:00','2','3','ajc','3');
264 insert into procedure_charting values ('d4 charting', '123456789', '2019/12/17
       17:00:00','1','1','ajc','3');
265 insert into procedure_charting values ('d4 charting', '123456789', '2019/12/17
       17:00:00','2','2','ajc','2');
```