

Sistema de Informação e Base de Dados

Assignment 2 – Implementing the Database

Year 2019/2020

Group 60

Work done by:

- Paz De Pablo-Romero Parias, nº 94852*
- Francisco Javier Gómez Lozano, nº 94874*
- Francisco Rabaça Moller Freiria, nº 97236*

SQL instructions to create the database

```
SET FOREIGN_KEY_CHECKS = 0;

drop table if exists phone_number_employee;

drop table if exists recepcionist;

drop table if exists doctor;

drop table if exists nurse;

drop table if exists phone_number_client;

drop table if exists permanent_doctor;

drop table if exists trainee_doctor;

drop table if exists supervision_report;

drop table if exists appointment;

drop table if exists consultation;

drop table if exists consultation_assistant;

drop table if exists diagnostic_code_relation;

drop table if exists consultation_diagnostic;

drop table if exists prescription;

drop table if exists procedure_in_consultation;

drop table if exists procedure_radiology;

drop table if exists procedure_charting;

drop table if exists employee;

drop table if exists client;

drop table if exists diagnostic_code;

drop table if exists medication;

drop table if exists proceduree;

drop table if exists teeth;

SET FOREIGN_KEY_CHECKS = 1;
```

create table employee

```
(VAT varchar(11),  
  name varchar(255),  
      birth_date date,  
      street varchar(255),  
      city varchar(100),  
      zip varchar(10),  
      IBAN varchar(25),  
      salary integer,  
      primary key(VAT),  
      unique(IBAN),  
      check(salary>=0));
```

SELECT receptionist.VAT,

nurse.VAT,

doctor.VAT,

employee.VAT

FROM employee

INNER JOIN receptionist ON receptionist.VAT = employee.VAT

INNER JOIN nurse ON nurse.VAT = employee.VAT

INNER JOIN doctor ON doctor.VAT = employee.VAT

create table phone_number_employee

```
(VAT varchar(11),  
  phone varchar(20),  
      primary key(VAT, phone),  
      foreign key(VAT) references employee(VAT) on delete cascade);
```

create table recepcionist

(VAT varchar(11),
primary key(VAT),
foreign key(VAT) references employee(VAT) on delete cascade);

create table doctor

(VAT varchar(11),
specialization varchar (40),
biografy varchar(255),
email varchar(255),
primary key(VAT),
foreign key(VAT) references employee(VAT) on delete cascade,
unique (email));

SELECT permanent_doctor.VAT,

trainee_doctor.VAT,

doctor.VAT

FROM doctor

left outer join permanent_doctor ON permanent_doctor.VAT = doctor.VAT

left outer join trainee_doctor ON trainee_doctor.VAT = doctor.VAT

--integraty constrain related with specialization of employee 'IC: All employees are
either receptionists, nurses or doctors'

create table nurse

(VAT varchar(11),
primary key(VAT),
foreign key(VAT) references employee(VAT) on delete cascade);

create table client

```
(VAT varchar(11),  
name varchar(255),  
birth_date date,  
street varchar(255),  
city varchar(100),  
zip varchar(10),  
gender varchar(9),  
age integer,  
primary key(VAT));
```

create table phone_number_client

```
(VAT varchar(11),  
phone varchar(20),  
primary key(VAT,phone),  
foreign key(VAT) references client(VAT) on delete cascade);
```

create table permanent_doctor

```
(VAT varchar(11),  
years integer,  
primary key(VAT),  
foreign key(VAT) references doctor(VAT) on delete cascade);
```

create table trainee_doctor

```
(VAT varchar(11),  
supervisor varchar(11),  
primary key(VAT),  
foreign key(VAT) references doctor(VAT) on delete cascade,
```

```
foreign key(supervisor) references permanent_doctor(VAT) on delete  
cascade);
```

--As supervisor doesn't appears within permanent_doctor we suppose
that it refers to VAT of permanent_doctor.

```
create table supervision_report  
  
    (VAT varchar(11),  
     date_timestamp timestamp,  
     description varchar(255),  
     evaluation integer,  
     primary key(VAT,date_timestamp),  
     foreign key(VAT) references trainee_doctor(VAT) on delete cascade,  
     check(evaluation>=1),  
     check(evaluation<=5));
```

```
create table appointment  
  
    (VAT_doctor varchar(11),  
     date_timestamp timestamp,  
     description varchar(255),  
     VAT_client varchar(11),  
     primary key(VAT_doctor,date_timestamp),  
     foreign key(VAT_doctor) references doctor(VAT) on delete cascade,  
     foreign key(VAT_client) references client(VAT) on delete cascade);
```

```
create table consultation  
  
    (VAT_doctor varchar(11),  
     date_timestamp timestamp,  
     SOAP_S varchar(255),
```

```
SOAP_O varchar(255),
SOAP_A varchar(255),
SOAP_P varchar(255),
primary key(VAT_doctor,date_timestamp),
foreign key(VAT_doctor, date_timestamp) references
appointment(VAT_doctor, date_timestamp) on delete cascade);
```

```
create table consultation_assistant
```

```
(VAT_doctor varchar(11),
date_timestamp timestamp,
VAT_nurse varchar(11),
primary key(VAT_doctor,date_timestamp),
foreign key(VAT_doctor) references appointment(VAT_doctor) on delete
cascade,
foreign key(date_timestamp) references consultation(date_timestamp)
on delete cascade,
foreign key(VAT_nurse) references nurse(VAT) on delete cascade);
```

```
select consultation_assistant.VAT_nurse,
nurse.VAT
from consultation_assistant
LEFT JOIN consultation_assistant on consultation_assistant.VAT_nurse=nurse.VAT
--'IC: Consultations are always assigned to at least one assistant nurse'
```

```
create table diagnostic_code
```

```
(ID integer,
description varchar(255),
primary key(ID));
```

create table diagnostic_code_relation

```
(ID1 integer,  
ID2 integer,  
typee varchar(255),  
primary key(ID1,ID2),  
foreign key(ID1) references diagnostic_code(ID) on delete cascade,  
foreign key(ID2) references diagnostic_code(ID) on delete cascade);
```

create table consultation_diagnostic

```
(VAT_doctor varchar(11),  
date_timestamp timestamp,  
ID integer,  
primary key(VAT_doctor,date_timestamp,ID),  
foreign key(VAT_doctor) references consultation(VAT_doctor) on delete  
cascade,  
foreign key(date_timestamp) references consultation(date_timestamp)  
on delete cascade,  
foreign key(ID) references diagnostic_code(ID) on delete cascade);
```

create table medication

```
(name varchar(255),  
lab varchar(255),  
primary key(name, lab));
```

create table prescription

```
(name varchar(255),  
lab varchar(255),  
VAT_doctor varchar(11),  
date_timestamp timestamp,
```



```
ID integer,  
dosage integer,  
description varchar(255),  
primary key(name,lab,VAT_doctor,date_timestamp,ID),  
foreign key(VAT_doctor) references consultation_diagnostic(VAT_doctor)  
on delete cascade,  
foreign key(date_timestamp) references  
consultation_diagnostic(date_timestamp) on delete cascade,  
foreign key(ID) references consultation_diagnostic(ID) on delete cascade,  
foreign key(name) references medication(name) on delete cascade,  
foreign key(lab) references medication(lab) on delete cascade);
```

create table proceduree

```
(name varchar(255),  
typee varchar(255),  
primary key(name));
```

create table procedure_in_consultation

```
(name varchar(255),  
VAT_doctor varchar(11),  
date_timestamp timestamp,  
description varchar(255),  
primary key(name,VAT_doctor,date_timestamp),  
foreign key(name) references proceduree(name) on delete cascade,  
foreign key(VAT_doctor) references consultation(VAT_doctor) on delete  
cascade,  
foreign key(date_timestamp) references consultation(date_timestamp)  
on delete cascade);
```

```
create table procedure_radiology
    (name varchar(255),
    filee varchar(255),
    VAT_doctor varchar(11),
    date_timestamp timestamp,
    primary key(name,filee,VAT_doctor,date_timestamp),
    foreign key(name) references procedure_in_consultation(name) on
delete cascade,
    foreign key(VAT_doctor) references
procedure_in_consultation(VAT_doctor) on delete cascade,
    foreign key(date_timestamp) references
procedure_in_consultation(date_timestamp) on delete cascade);
```

```
create table teeth
    (quadrant integer,
    numbeer integer,
    name varchar(255),
    primary key(quadrant,numbeer));
```

```
create table procedure_charting
    (name varchar(255),
    VAT varchar(11),
    date_timestamp timestamp,
    quadrant numeric(2,0),
    numbeer numeric(2,0),
    description varchar(255),
    measure varchar(255),
    primary key(name,VAT,date_timestamp,quadrant,numbeer),
```

foreign key(name) references procedure_in_consultation(name) on
delete cascade,

foreign key(VAT) references procedure_in_consultation(VAT_doctor) on
delete cascade,

foreign key(date_timestamp) references
procedure_in_consultation(date_timestamp) on delete cascade,

foreign key(quadrant) references teeth(quadrant) on delete cascade,

foreign key(numbeer) references teeth(numbeer) on delete cascade);

SQL script to populate the tables of the relational database

insert into employee values ('PT123456100', 'John Door', 1996-06-06, 'Bairro das Portas N02', 'Lisboa', '2700-252', 'PT50002700000001234567222', 1000);

insert into employee values ('PT123456789', 'António Santos', 1997-10-08, 'Rua das Flores N32', 'Lisboa', '2700-832', 'PT50002700000001234567833', 1000);

insert into employee values ('PT123456788', 'Beatriz Salvado', 1994-09-07, 'Rua dos Corvos N23', 'Lisboa', '2725-254', 'PT50002700000009834567444', 1100);

insert into employee values ('PT254696301', 'Bernardo Cruz', 1995-03-28, 'Rua dos Condes N54', 'Lisboa', '2775-157', 'PT50002700000001549567833', 1100);

insert into employee values ('PT251627561', 'Carla Mendes', 1993-05-18, 'Rua dos Padres N89', 'Lisboa', '2765-749', 'PT50002700000001537467833', 1050);

insert into employee values ('PT248787421', 'Eva Dias', 1992-12-27, 'Rua dos Pecados N01', 'Lisboa', '2700-442', 'PT50002700000001534167412', 1150);

insert into employee values ('PT178781451', 'Igor Criador', 1980-04-08, 'Rua dos Recados N46', 'Lisboa', '2700-751', 'PT50002700000001537467063', 2400);

insert into employee values ('PT156548357', 'Jane Sweettooth', 1979-06-02, 'Rua da Causa N13', 'Lisboa', '2745-361', 'PT500027000000015345075641', 2500);

insert into employee values ('PT158696188', 'Joao Felix', 1992-09-21, 'Rua dos Jovens N01', 'Lisboa', '2700-681', 'PT50002700000001534167471', 1600);

insert into employee values ('PT254673667', 'Helder Postiga', 1991-10-02, 'Rua dos Perdidos N42', 'Lisboa', '2700-592', 'PT50002700000001537467122', 1500);

insert into employee values ('PT254136856', 'Hugo Almeida', 1989-02-13, 'Rua dos Malucos N33', 'Lisboa', '2745-403', 'PT500027000000015345075963', 1500);

insert into phone_number_employee values ('PT123456789', '+351924569321');

insert into phone_number_employee values ('PT123456788', '+351963214567');

insert into phone_number_employee values ('PT254696301', '+351913258467');

insert into phone_number_employee values ('PT251627561', '+351936573214');

```
insert into phone_number_employee values ('PT248787421','+351921456832');
insert into phone_number_employee values ('PT178781451','+351965547183');
insert into phone_number_employee values ('PT156548357','+351921473598');
insert into phone_number_employee values ('PT158696188','+351963258741');
insert into phone_number_employee values ('PT254673667','+351935642352');
insert into phone_number_employee values ('PT254136856','+351916548743');
```

```
insert into recepcionist values ('PT123456100');
```

```
insert into doctor values ('PT178781451','Oral pathology','A lot of
Experience','igorthecreator@gmail.com');
```

```
insert into doctor values ('PT156548357','Dento-maxillofacial radiology','Recognized by
the order','toothachecalljane@gmail.com');
```

```
insert into doctor values ('PT158696188','Endodontic','Trainee','joaofelis@gmail.com');
```

```
insert into doctor values
('PT254673667','Orthodontic','Trainee','helderaleijado@gmail.com');
```

```
insert into doctor values
('PT254136856','Periodontics','Trainee','almeidaturco@gmail.com');
```

```
insert into nurse values ('PT123456789');
```

```
insert into nurse values ('PT123456788');
```

```
insert into nurse values ('PT254696301');
```

```
insert into nurse values ('PT251627561');
```

```
insert into nurse values ('PT248787421');
```

```
insert into client values ('PT258469131','Anciento Antunes',1998-04-23,'Bairro Lilás
N12','Lisboa','2700-149','Male',20);
```

```
insert into client values ('PT257469822','Ariana Pequena',1997-07-14,'Bairro Laranja
N08','Lisboa','2700-211','Female',21);
```

insert into client values ('PT267894513','Beatriz Bento',1996-01-27,'Bairro Vermelho N75','Lisboa','2700-446','Female',22);

insert into client values ('PT243567144','Clara Estrela',1995-03-12,'Bairro Branco N15','Lisboa','2700-156','Female',22);

insert into client values ('PT248569325','Estrela Silva',1999-11-07,'Bairro Amarelo N41','Lisboa','2700-325','Female',24);

insert into client values ('PT198523656','Francisco Antunes',1994-09-22,'Bairro Branco N04','Lisboa','2700-459','Male',40);

insert into client values ('PT169696527','Guilherme Marques',1997-11-16,'Bairro Castanho N37','Lisboa','2700-553','Male',10);

insert into client values ('PT178964338','Hugo Morais',1991-01-31,'Bairro Violeta N58','Lisboa','2700-541','Male',9);

insert into client values ('PT233455679','Inês Pereira',1996-06-22,'Bairro Rosa N33','Lisboa','2700-333','Female',22);

insert into client values ('PT222555990','Inês Saraiva',1998-01-07,'Bairro Preta N69','Lisboa','2700-696','Female',25);

insert into phone_number_client values ('PT258469131','+351923564121');

insert into phone_number_client values ('PT257469822','+351924563892');

insert into phone_number_client values ('PT267894513','+351914561383');

insert into phone_number_client values ('PT243567144','+351936712354');

insert into phone_number_client values ('PT248569325','+351925379955');

insert into phone_number_client values ('PT198523656','+351917239336');

insert into phone_number_client values ('PT169696527','+351932754697');

insert into phone_number_client values ('PT178964338','+351965321898');

insert into phone_number_client values ('PT233455679','+351925631569');

insert into phone_number_client values ('PT222555990','+351935641230');

insert into permanent_doctor values ('PT178781451',7);

insert into permanent_doctor values ('PT156548357',8);

insert into trainee_doctor values ('PT158696188','PT178781451');

insert into trainee_doctor values ('PT254673667','PT178781451');

insert into trainee_doctor values ('PT254136856','PT156548357');

insert into supervision_report values ('PT158696188',2019-04-12
10:21:12.06,'Insufficient Work',2);

insert into supervision_report values ('PT254673667',2019-06-12 11:20:12.04,'Keep
working',3);

insert into supervision_report values ('PT254136856',2019-08-12 19:22:12.02,'Nice
Job',5);

insert into appointment values ('PT178781451',2019-10-14
09:00:30.15,'Bleaching','PT258469131');

insert into appointment values ('PT178781451',2019-10-15 11:00:30.25,'Caries
treatment','PT257469822');

insert into appointment values ('PT178781451',2019-10-16 14:00:30.35,'Wisdom tooth
removal','PT267894513');

insert into appointment values ('PT178781451',2019-10-17 15:00:30.45,'Dental
Checking','PT243567144');

insert into appointment values ('PT156548357',2019-10-17
16:30:30.55,'Braces','PT248569325');

insert into appointment values ('PT156548357',2019-10-18
09:31:30.65,'Dentures','PT198523656');

insert into appointment values ('PT156548357',2019-10-18
12:00:30.75,'Veneers','PT169696527');

insert into appointment values ('PT156548357',2019-10-19
09:00:30.85,'Sealants','PT178964338');

insert into appointment values ('PT158696188',2019-10-20
09:40:40.95,'Braces','PT233455679');

insert into appointment values ('PT254673667',2019-10-21
09:00:30.05,'Extractions','PT222555990');

insert into appointment values ('PT254136856',2019-10-21
12:14:30.15,'Bonding','PT243567144');

insert into appointment values ('PT178781451',2019-12-04 10:00:30.25,'Gum Surgery','PT258469131');

insert into appointment values ('PT156548357',2019-12-05 09:00:30.35,'Root Canals','PT257469822');

insert into consultation values ('PT178781451',2019-10-14 09:00:30.15,'Bad Treatment before','crooked teeth', 'Small carie', 'More consultations');

insert into consultation values ('PT178781451',2019-10-15 11:00:30.25,'Bad Treatment before','crooked teeth', 'Abscess', 'More consultations');

insert into consultation values ('PT178781451',2019-10-17 15:00:30.45,'Bad Treatment before','crooked teeth', 'Small carie', 'More consultations');

insert into consultation values ('PT156548357',2019-10-17 16:30:30.55,'Bad Treatment before','crooked teeth', 'Abscess', 'More consultations');

insert into consultation values ('PT156548357',2019-10-18 09:31:30.65,'Bad Treatment before','crooked teeth', 'Small carie', 'More consultations');

insert into consultation values ('PT156548357',2019-10-18 12:00:30.75,'Bad Treatment before','crooked teeth', 'Abscess', 'More consultations');

insert into consultation values ('PT158696188',2019-10-20 09:40:40.95,'Bad Treatment before','crooked teeth', 'Small carie', 'More consultations');

insert into consultation values ('PT254673667',2019-10-21 09:00:30.05,'Bad Treatment before','crooked teeth', 'Abscess', 'More consultations');

insert into consultation_assistant values ('PT178781451',2019-10-14 09:00:30.15,'PT123456789');

insert into consultation_assistant values ('PT178781451',2019-10-15 11:00:30.25,'PT123456788');

insert into consultation_assistant values ('PT178781451',2019-10-17 15:00:30.45,'PT254696301');

insert into consultation_assistant values ('PT156548357',2019-10-17 16:30:30.55,'PT251627561');

insert into consultation_assistant values ('PT156548357',2019-10-18 09:31:30.65,'PT248787421');

insert into consultation_assistant values ('PT156548357',2019-10-18
12:00:30.75,'PT123456789');

insert into consultation_assistant values ('PT158696188',2019-10-20
09:40:40.95,'PT123456788');

insert into consultation_assistant values ('PT254673667',2019-10-21
09:00:30.05,'PT254696301');

insert into diagnostic_code values (0001,'Gengivist');

insert into diagnostic_code values (0002,'Attrition ');

insert into diagnostic_code values (0003,'Caries');

insert into diagnostic_code values (0004,'Recurrent Caries');

insert into diagnostic_code values (0005,'Concussion Injuries');

insert into diagnostic_code values (0006,'Enamel Hypoplasia');

insert into diagnostic_code values (0007,'Fluorosis');

insert into diagnostic_code values (0008,'Irreversible Pulpitis');

insert into diagnostic_code values (0009,'Malocclusion');

insert into diagnostic_code values (0010,'Poor Oral Hygiene');

insert into diagnostic_code values (0011,'Lymphoma');

insert into diagnostic_code values (0012,'Oral Cancer');

insert into diagnostic_code_relation values (0003,0004,'No teeth care');

insert into diagnostic_code_relation values (0001,0007,'Weak gums');

insert into diagnostic_code_relation values (0011,0012,'Cancer');

insert into consultation_diagnostic values ('PT178781451',2019-10-14
09:00:30.15,0002);

insert into consultation_diagnostic values ('PT178781451',2019-10-15
11:00:30.25,0004);

insert into consultation_diagnostic values ('PT178781451',2019-10-17
15:00:30.45,0006);

insert into consultation_diagnostic values ('PT156548357',2019-10-17
16:30:30.55,0008);

insert into consultation_diagnostic values ('PT156548357',2019-10-18
09:31:30.65,0010);

insert into consultation_diagnostic values ('PT156548357',2019-10-18
12:00:30.75,0012);

insert into consultation_diagnostic values ('PT158696188',2019-10-20
09:40:40.95,0001);

insert into consultation_diagnostic values ('PT254673667',2019-10-21
09:00:30.05,0003);

insert into medication values ('Ibuprofeno ','Lab1');

insert into medication values ('Acetaminophen ','Lab2');

insert into medication values ('Orabase ','Lab3');

insert into medication values ('Ambesol ','Lab4');

insert into medication values ('Chloraseptic ','Lab5');

insert into medication values ('Xylocaine ','Lab6');

insert into prescription values ('Ibuprofeno','Lab1','PT178781451',2019-10-14
09:00:30.15,0002,400,'Every 8h for 3 days');

insert into prescription values ('Ibuprofeno','Lab1','PT178781451',2019-10-15
11:00:30.25,0004,400,'Every 6h for 3 days');

insert into prescription values ('Ambesol','Lab4','PT178781451',2019-10-17
15:00:30.45,0006,600,'Every day at morning for 3 days');

insert into prescription values ('Acetaminophen','Lab2','PT156548357',2019-10-17
16:30:30.55,0008,600,'Every day at morning for 5 days');

insert into prescription values ('Orabase','Lab3','PT156548357',2019-10-18
09:31:30.65,0010,600,'Every day 1 week');

insert into prescription values ('Chloraseptic','Lab5','PT156548357',2019-10-18
12:00:30.75,0012,400,'Every 12h for 4 days');

insert into prescription values ('Orabase','Lab3','PT158696188',2019-10-20
09:40:40.95,0001,600,'Every day at morning for 2 days');

insert into prescription values ('Xylocaine','Lab6','PT254673667',2019-10-21
09:00:30.05,0003,400,'Every day at morning for 3 days');

insert into proceduree values ('Caries treatment','No evasive');

insert into proceduree values ('Wisdom tooth removal','Evasive');

insert into proceduree values ('Bleaching','No evasive');

insert into proceduree values ('Dental Checking','No evasive');

insert into proceduree values ('Braces','No evasive');

insert into proceduree values ('Dentures','No evasive');

insert into proceduree values ('Veneers','Evasive');

insert into proceduree values ('Sealants','No evasive');

insert into proceduree values ('Extractions','Evasive');

insert into proceduree values ('Bonding','Evasive');

insert into proceduree values ('Gum Surgery','Evasive');

insert into proceduree values ('Root Canals','No evasive');

insert into proceduree values ('Recomend Specialist','No evasive');

insert into procedure_in_consultation values ('Dental Checking','PT178781451',2019-
10-14 09:00:30.15,'Gone well');

insert into procedure_in_consultation values ('Caries treatment','PT178781451',2019-
10-15 11:00:30.25,'Gone well');

insert into procedure_in_consultation values ('Dentures','PT178781451',2019-10-17
15:00:30.45,'Advice about Carie');

insert into procedure_in_consultation values ('Root Canals','PT156548357',2019-10-17
16:30:30.55,'Gone well');

insert into procedure_in_consultation values ('Bonding','PT156548357',2019-10-18
09:31:30.65,'Gone well');

insert into procedure_in_consultation values ('Recomend
Specialist','PT156548357',2019-10-18 12:00:30.75,'Detect cancer');

insert into procedure_in_consultation values ('Bleaching','PT158696188',2019-10-20
09:40:40.95,'Gone well');

insert into procedure_in_consultation values ('Caries treatment','PT254673667',2019-10-21 09:00:30.05,'Small bleeding');

insert into procedure_radiology values ('Caries treatment','file1','PT178781451',2019-10-15 11:00:30.25);

insert into procedure_radiology values ('Dentures','file2','PT178781451',2019-10-17 15:00:30.45);

insert into procedure_radiology values ('Root Canals','file3','PT156548357',2019-10-17 16:30:30.55);

insert into procedure_radiology values ('Caries treatment','file4','PT254673667',2019-10-21 09:00:30.05);

insert into teeth values (1,1,'Third Molar');

insert into teeth values (1,2,'Second Molar');

insert into teeth values (1,3,'First Molar');

insert into teeth values (1,4,'Second Bicuspid');

insert into teeth values (1,5,'First Bicuspid');

insert into teeth values (1,6,'Cuspid');

insert into teeth values (1,7,'Lateral Incisor');

insert into teeth values (1,8,'Central Incisor');

insert into teeth values (2,9,'Central Incisor');

insert into teeth values (2,10,'Lateral Incisor');

insert into teeth values (2,11,'Cuspid');

insert into teeth values (2,12,'First Bicuspid');

insert into teeth values (2,13,'Second Bicuspid');

insert into teeth values (2,14,'First Molar');

insert into teeth values (2,15,'Second Molar');

insert into teeth values (2,16,'Third Molar');

insert into teeth values (3,17,'Third Molar');

insert into teeth values (3,18,'Second Molar');

insert into teeth values (3,19,'First Molar');
insert into teeth values (3,20,'Second Bicuspid');
insert into teeth values (3,21,'First Bicuspid');
insert into teeth values (3,22,'Cuspid');
insert into teeth values (3,23,'Lateral Incisor');
insert into teeth values (3,24,'Central Incisor');
insert into teeth values (4,25,'Central Incisor');
insert into teeth values (4,26,'Lateral Incisor');
insert into teeth values (4,27,'Cuspid');
insert into teeth values (4,28,'First Bicuspid');
insert into teeth values (4,29,'Second Bicuspid');
insert into teeth values (4,30,'First Molar');
insert into teeth values (4,31,'Second Molar');
insert into teeth values (4,32,'Third Molar');

insert into procedure_charting values ('Caries treatment','PT178781451',2019-10-15
11:00:30.25,1,7,'Small carie','Removed');
insert into procedure_charting values ('Root Canals','PT156548357',2019-10-17
16:30:30.55,2,14,'Abscess','Medicated');
insert into procedure_charting values ('Caries treatment','PT254673667',2019-10-21
09:00:30.05,3,19,'Big carie','Removed');

SQL queries for given information [1]

Select c.VAT,c.name,p.phone

From employee e, doctor d, client c, phone_number_client p, consultation co,
appointment a

Where a.VAT_client=c.VAT

and co.VAT_doctor=a.VAT_doctor

and a.VAT_doctor=d.VAT

and d.VAT=e.VAT

and e.name= 'Jane Sweettooth'

Order by c.name

SQL queries for given information [2]

((Select t.VAT, s.evaluation, s.description, e.name

From trainee_doctor t, supervision_report s, employee e

Where (s.evaluation < 3 or s.description like '%insufficient%')

and t.VAT=d.VAT

and d.VAT=e.VAT)

intersect

(Select e.name

From trainee_doctor t, supervision_report s, employee e, permanente_doctor pe

Where (s.evaluation < 3 or s.description like '%insufficient%')

and t.supervisor=pe.VAT

and pe.VAT=d.VAT

and d.VAT=e.VAT))

Desc s.evaluation

--we decide to use the intersect operator because we need from employee two different names from two different doctors.

SQL queries for given information [3]

Select c.name,c.city,c.VAT

From consultation co, client c, appointment a

Where c.VAT=a.VAT_client

and co.date_timestamp=a.date_timestamp

and (co.SOAP_O like '%gingivitis%' or co.SOAP_O like '%periodontitis%')

and co.date_timestamp = (select max(date_timestamp) from
consultation)

--Here we use the max() operator in order to find the most recent date

SQL queries for given information [4]

Select c.name, c.VAT, c.street, c.city, c.zip

From client c, consultation co, appointment a

Where c.VAT=a.VAT_client

and co.SOAP_O is null

and co.SOAP_S is null

and co.SOAP_A is null

and co.SOAP_P is null

--To know which client had never been in a consultation despite they had have an appointment we suppose that all of SOAP notes are empty.

SQL queries for given information [5]

Select dc.ID, dc.description, count(distinct m.name)

From diagnostic_code dc, medication m, prescription pre, consultation_diagnostic cd

Where cd.ID=dc.ID

and pre.ID=cd.ID

and pre.name=m.name

Group by dc.ID

Asc count(distinct m.name)

SQL queries for given information [6]

(select avg(count ca.VAT_nurse), avg (count cd.ID), avg(count pic.name),avg(count
pres.name)

From consultation_assistant ca, consultation_diagnostic cd, procedure_in_consultation
pic, prescription pres, client c

Where DATEPART(yy,ca.date_timestamp)=2019

and DATEPART(yy,pic.date_timestamp)= 2019

and DATEPART(yy,cd.date_timestamp)=2019

and DATEPART(yy,pres.date_timestamp)=2019

Group by c.age

Having c.age <=18)

Union

(select avg(count ca.VAT_nurse), avg (count cd.ID), avg(count pic.name),avg(count
pres.name)

From consultation_assistant ca, consultation_diagnostic cd, procedure_in_consultation
pic, prescription pres, client c

Where DATEPART(yy,ca.date_timestamp)=2019

and DATEPART(yy,pic.date_timestamp)= 2019

and DATEPART(yy,cd.date_timestamp)=2019

and DATEPART(yy,pres.date_timestamp)=2019

Group by c.age

Having c.age >18)

--We use the union operator to divide in two different groups (depending on the age of client). We also use the function DATEPART to find the year that project two index requires.

SQL queries for given information [7]

Select m.name

From medication m, diagnostic_code dc, prescription pre, consultation_diagnostic cd

Where m.name=pre.name

and dc.ID=cd.ID

and cd.ID=pre.ID

and count (pre.name) >all (select count(name) from prescription group
by name)

Group by pre.ID

SQL queries for given information [8]

((select m.name, m.labs

From prescription pre, diagnostic_code dc, medication m, consultation_diagnostic cd

Where DATEPART(yy,pre.date_timestamp)=2019

and dc.description like '%dental%'

and dc.description like '%cavities%'

and cd.ID=dc.ID

and cd.ID=pre.ID

and m.name=pre.name)

Except

(select m.name, m.labs

From prescription pre, diagnostic_code dc, medication m, consultation_diagnostic cd

Where DATEPART(yy,pre.date_timestamp)=2019

and dc.description like '%infectious%'

and dc.description like '%disease%'

and cd.ID=dc.ID

and cd.ID=pre.ID

and m.name=pre.name))

Order by m.name,m.labs

--We use the Except operator to find all the dental cavities problems but avoiding the infectious disease.

SQL queries for given information [9]

Select c.name,c.street,c.city,c.zip

From client c, appointment a

Appointment natural join consultation

Where DATEPART(yy,a.date_timestamp)=2019

and a.VAT_client=c.VAT

--Natural join give us the information of clients that goes to every consultant after an appointment.

SQL instructions for changes in the database [1]

```
update employee
set street = 'Sierps' and city = 'Sevilha'
where name='Jane Sweettoth'
```

SQL instructions for changes in the database [2]

```
update employee
set salary = salary * 1.05
where count(VAT_doctor)>100
      (select count (VAT_doctor)
       from appointment
       where DATEPART (yy, a.date_timestamp)=2019
       group by VAT_doctor)
```

SQL instructions for changes in the database [3]

```
delete from employee e, appointment a, consultation co, prescription pres,
      procedure_in_consultation pic, consultation_diagnostic cd
where c.name = 'Jane Sweettoth'
      and c.VAT = a.VAT_doctor
      and a.VAT_doctor = co.VAT_doctor
      and co. VAT_doctor = pres.VAT_doctor
      and pres VAT_doctor = pic.VAT_doctor
      and pic.VAT_doctor = cd.VAT_doctor
```

SQL instructions for changes in the database [4]

```
select dc.ID
from diagnostic_code dc
where dc.description like '%gingivitis%'
insert into diagnostic_code values (0013,'periodontitis')
update consultation_diagnostic
from consultation_diagnostic cd , procedure_charting pc
set cd.ID = 0013
        and cd.description = 'periodontitis'
where cd.ID = 0001
        and avg(pc.measure)>4
```

Views

create view dim_date as

```
(select co.date_timestamp, DAY (co.date_timestamp), MONTH  
(co.date_timestamp), YEAR(co.date_timestamp)  
from consultation co)
```

create view dim_client as

```
(select VAT, gender, age  
from client)
```

create view dim_location_client as

```
(select zip, city  
from client)
```

create view facts_consults as

```
(select c.VAT, dd.datee, c.zip, count(distinct pr.name), count(distinct  
m.name), count(distinct dc.ID)  
from client c, dim_date dd, proceduree pr, medication m,  
diagnostic_code dc)
```

SQL instructions for implementing the Indexes [1]

Create index <IDX_consultations_swettoth>

```
ON clients (VAT)
```

Create index <IDX_consultations_swettoth1>

```
ON clients (name)
```

Create index <IDX_consultations_swettoth2>

```
ON clients (phone_number)
```

SQL instructions for implementing the Indexes [2]

Create index <IDX_trainee>

ON trainee_doctor (VAT)

Create index <IDX_trainee1>

ON trainee_doctor (name)

Create index <IDX_trainee2>

ON trainee_doctor (supervisor)

Create index <IDX_trainee3>

ON trainee_doctor (evaluation)

Create full text index <IDX_trainee4>

ON trainee_doctor (description)