Instituto Superior Técnico

Information Systems and Databases

Project Assignment - Part 2

Assignment 2 - Implementing the Database

Alunos	*Número de aluno*
Bernardo Bastos	84012
Ivo Simões	87024
João Chumbinho	81761

1 Create tables

The code used to create the tables in the database was the following.

```
CREATE TABLE employee
    (VAT INT NOT NULL UNIQUE,
    name_ VARCHAR(255),
    birth_date DATE,
    street VARCHAR(255),
    city VARCHAR(255),
    zip VARCHAR(255),
    IBAN VARCHAR(100) NOT NULL UNIQUE,
    salary NUMERIC(20,2) NOT NULL CHECK (salary > 0),
    PRIMARY KEY(VAT));


CREATE TABLE phone_number_employee
    (VAT INT NOT NULL,
    phone VARCHAR(15),
    PRIMARY KEY(VAT, phone),
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE);


CREATE TABLE receptionist
    (VAT INT NOT NULL UNIQUE,
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE);


CREATE TABLE doctor
    (VAT INT NOT NULL UNIQUE,
    specialization VARCHAR(255),
    biography VARCHAR(255),
    email VARCHAR(255) NOT NULL UNIQUE,
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE);


CREATE TABLE nurse
    (VAT INT NOT NULL UNIQUE,
    PRIMARY KEY(VAT),
    FOREIGN KEY(VAT) REFERENCES employee(VAT) ON DELETE CASCADE);
```

```
39
40
41 CREATE TABLE client
42     (VAT INT NOT NULL UNIQUE,
43     name_ VARCHAR(255),
44     birth_date DATE,
45     street VARCHAR(255),
46     city VARCHAR(255),
47     zip VARCHAR(255),
48     gender VARCHAR(20),
49     age INT NOT NULL CHECK (age > 0),
50     PRIMARY KEY(VAT));
51
52
53 CREATE TABLE phone_number_client
54     (VAT INT NOT NULL UNIQUE,
55     phone VARCHAR(15),
56     PRIMARY KEY(VAT, phone),
57     FOREIGN KEY(VAT) REFERENCES client(VAT) ON DELETE CASCADE);
58
59
60 CREATE TABLE permanent_doctor
61     (VAT INT NOT NULL UNIQUE,
62     years INT,
63     PRIMARY KEY(VAT),
64     FOREIGN KEY(VAT) REFERENCES doctor(VAT) ON DELETE CASCADE);
65
66
67 CREATE TABLE trainee_doctor
68     (VAT INT NOT NULL UNIQUE,
69     supervisor INT,
70     PRIMARY KEY(VAT),
71     FOREIGN KEY(VAT) REFERENCES doctor(VAT) ON DELETE CASCADE,
72     FOREIGN KEY(supervisor) REFERENCES permanent_doctor(VAT) ON DELETE
    CASCADE);
73
74
75 CREATE TABLE appointment
76     (VAT_doctor INT NOT NULL,
77     date_timestamp DATETIME NOT NULL,
78     description_ VARCHAR(255),
79     VAT_client INT NOT NULL,
```

```
80      PRIMARY KEY(VAT_doctor, date_timestamp),
81      FOREIGN KEY(VAT_doctor) REFERENCES doctor(VAT) ON DELETE CASCADE,
82      FOREIGN KEY(VAT_client) REFERENCES client(VAT) ON DELETE CASCADE);
83
84
85  CREATE TABLE supervision_report
86      (VAT INT NOT NULL,
87      date_timestamp DATETIME,
88      description_ VARCHAR(255),
89      evaluation INT CHECK(evaluation>=1 AND evaluation <= 5),
90      PRIMARY KEY(VAT, date_timestamp),
91      FOREIGN KEY(VAT) REFERENCES trainee_doctor(VAT) ON DELETE CASCADE);
92
93
94  CREATE TABLE consultation
95      (VAT_doctor INT NOT NULL,
96      date_timestamp DATETIME NOT NULL,
97      SOAP_S VARCHAR(255),
98      SOAP_O VARCHAR(255),
99      SOAP_A VARCHAR(255),
100     SOAP_P VARCHAR(255),
101     PRIMARY KEY(VAT_doctor, date_timestamp),
102     FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES appointment(
        VAT_doctor, date_timestamp) ON DELETE CASCADE);
103
104
105 CREATE TABLE consultation_assistant
106     (VAT_doctor INT NOT NULL,
107     date_timestamp DATETIME NOT NULL,
108     VAT_nurse INT NOT NULL,
109     PRIMARY KEY(VAT_doctor, date_timestamp, VAT_nurse),
110     FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES consultation(
        VAT_doctor, date_timestamp) ON DELETE CASCADE,
111     FOREIGN KEY(VAT_nurse) REFERENCES nurse(VAT) ON DELETE CASCADE);
112
113
114 CREATE TABLE diagnostic_code
115     (ID INT NOT NULL UNIQUE,
116     description VARCHAR(255),
117     PRIMARY KEY(ID));
118
119
```

```
120 CREATE TABLE diagnostic_code_relation
121     (ID1 INT NOT NULL,
122     ID2 INT NOT NULL,
123     type_ VARCHAR(255),
124     PRIMARY KEY(ID1, ID2),
125     FOREIGN KEY(ID1) REFERENCES diagnostic_code(ID) ON DELETE CASCADE,
126     FOREIGN KEY(ID2) REFERENCES diagnostic_code(ID) ON DELETE CASCADE);
127
128
129 CREATE TABLE consultation_diagnostic
130     (VAT_doctor INT NOT NULL,
131     date_timestamp DATETIME NOT NULL,
132     ID INT NOT NULL,
133     PRIMARY KEY(VAT_doctor, date_timestamp, ID),
134     FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES consultation(
    VAT_doctor, date_timestamp) ON DELETE CASCADE,
135     FOREIGN KEY(ID) REFERENCES diagnostic_code(ID) ON DELETE CASCADE ON
    UPDATE CASCADE);
136
137
138 CREATE TABLE medication
139     (name_ VARCHAR(255) NOT NULL UNIQUE,
140     lab VARCHAR(255) NOT NULL,
141     PRIMARY KEY(name_, lab));
142
143
144 CREATE TABLE prescription
145     (name_ VARCHAR(255) NOT NULL,
146     lab VARCHAR(255) NOT NULL,
147     VAT_doctor INT NOT NULL,
148     date_timestamp DATETIME NOT NULL,
149     ID INT NOT NULL,
150     dosage VARCHAR(255),
151     description_ VARCHAR(255),
152     PRIMARY KEY(name_, lab, VAT_doctor, date_timestamp, ID),
153     FOREIGN KEY(VAT_doctor, date_timestamp, ID) REFERENCES
    consultation_diagnostic(VAT_doctor, date_timestamp, ID) ON DELETE
    CASCADE,
154     FOREIGN KEY(name_, lab) REFERENCES medication(name_, lab) ON DELETE
    CASCADE);
155
156
```

```
157 CREATE TABLE procedure_
158     (name_ VARCHAR(255) NOT NULL UNIQUE,
159     type_ VARCHAR(255),
160     PRIMARY KEY(name_));
161
162
163 CREATE TABLE procedure_in_consultation
164     (name_ VARCHAR(255) NOT NULL,
165     VAT_doctor INT NOT NULL,
166     date_timestamp DATETIME NOT NULL,
167     description_ VARCHAR(255),
168     PRIMARY KEY(name_, VAT_doctor, date_timestamp),
169     FOREIGN KEY(name_) REFERENCES procedure_(name_) ON DELETE CASCADE,
170     FOREIGN KEY(VAT_doctor, date_timestamp) REFERENCES consultation(
    VAT_doctor, date_timestamp) ON DELETE CASCADE);
171
172
173 CREATE TABLE procedure_radiology
174     (name_ VARCHAR(255) NOT NULL,
175     file_ VARCHAR(255),
176     VAT_doctor INT NOT NULL,
177     date_timestamp DATETIME NOT NULL,
178     PRIMARY KEY(name_, file_, VAT_doctor, date_timestamp),
179     FOREIGN KEY(name_, VAT_doctor, date_timestamp) REFERENCES
    procedure_in_consultation(name_, VAT_doctor, date_timestamp) ON
    DELETE CASCADE);
180
181
182 CREATE TABLE teeth
183     (quadrant VARCHAR(255),
184     number_ INT,
185     name_ VARCHAR(255),
186     PRIMARY KEY(quadrant, number_));
187
188
189 CREATE TABLE procedure_charting
190     (name_ VARCHAR(255) NOT NULL,
191     VAT INT NOT NULL,
192     date_timestamp DATETIME NOT NULL,
193     quadrant VARCHAR(255),
194     number_ INT,
195     desc_ VARCHAR(255),
```

```
196     measure INT,
197     PRIMARY KEY(name_, VAT, date_timestamp, quadrant, number_),
198     FOREIGN KEY(name_, VAT, date_timestamp) REFERENCES
    procedure_in_consultation(name_, VAT_doctor, date_timestamp) ON
    DELETE CASCADE,
199     FOREIGN KEY(quadrant, number_) REFERENCES teeth(quadrant, number_)
    ON DELETE CASCADE);
```

2 Inserting Records

The code used to inserting records was the following.

```
1  Insert into client values (12345678, 'Joao', '1996-08-08', 'Praceta da
       Luz', 'Lisboa', '2610-062', 'Male', 23);
2  Insert into client values (23456789, 'Rui', '1946-08-08', 'Avenida da
       Liberdade ', 'Lisboa', '2610-062', 'Male', 40);
3  Insert into client values (89123456, 'Ze', '1967-03-01', 'Rossio', '
       Lisboa', '2610-062', 'Male', 89);
4  Insert into client values (87654321, 'Maria', '2003-01-14', 'Rua do
       Campo', 'Rio de Janeiro', '2610-062', 'Female', 40);
5  Insert into client values (45678123, 'Tiago', '1965-01-01', 'Estrada da
       Luz', 'Lisboa', '2690-052', 'Male', 30);
6  Insert into client values (43215678, 'Rita', '1916-03-03', 'Ra[U+FFFD]da
       Alegria', 'Lisboa', '1610-035', 'Female', 103);
7  Insert into client values (18273645, 'Luis Carlos', '1997-03-28', 'Rua
       do Marqu[U+FFFD]s', 'Lisboa', '2610-350', 'Male', 22);
8  Insert into client values (87333321, 'H[U+FFFD]', '2005-08-17', 'Praceta do
      Com[U+FFFD]rcio', 'Gaia', '2610-062', 'Female', 14);
9
10
11 Insert into phone_number_client values(12345678,  351917654321);
12 Insert into phone_number_client values(23456789,  351917754321);
13 Insert into phone_number_client values(89123456,  351937754321);
14 Insert into phone_number_client values(87654321,  351937754421);
15 Insert into phone_number_client values(45678123,  351966654321);
16 Insert into phone_number_client values(43215678,  351922254321);
17 Insert into phone_number_client values(18273645,  351922224321);
18 Insert into phone_number_client values(87333321,  351937666421);
19
20
21 Insert into employee values(12345876, 'Joao', '1996-12-08', 'Principe
       Real', 'Lisboa', '2610-062', '339237212', 1300);
```

```
22 Insert into employee values(12456876, 'Rui', '1960-12-30', 'Ra[U+FFFD]de
   Londres', 'Lisboa', '2610-062', '44423721', 1300);
23 Insert into employee values(88888888, 'Jane Sweettooth', '1963-11-30', '
   Rua das Conchas', 'Lisboa', '2612-063', '55577788', 2500);
24 Insert into employee values(11111222, 'Jack Sweettath', '1964-11-15', '
   Rua da Falagueira', 'Lisboa', '2612-373', '44577788', 3000);
25 Insert into employee values(44440000, 'Joseph', '1965-03-26', 'Norwich',
    'London', '2610-062', '66677788', 1300);
26 Insert into employee values(14444876, 'Richard', '1995-01-26', 'Gatewick
   ', 'London', '2610-355', '11677788', 1300);
27 Insert into employee values(55555555, 'Mara', '1965-03-26', 'Central
   Park L', 'London', '2610-072', '65657788', 1300);
28 Insert into employee values(222000333, 'Johnson', '1965-03-26', 'Route
   69', 'London', '2610-062', '96677788', 1300);
29 Insert into employee values(14445876, 'Luis', '1978-01-08', 'AmadoraCity
   ', 'Lisboa', '2610-062', '66637212', 1300);
30 Insert into employee values(14345876, 'Ana', '1978-02-18', 'Ra[U+FFFD]das
   Flores', 'Lisboa', '2615-252', '66677212', 1000);
31 Insert into employee values(54445875, 'Lucas', '1979-01-18', 'Cova da
   Mora', 'Lisboa', '2610-333', '88823212', 1200);
32 Insert into employee values(54445867, 'Hugo', '1945-01-18', 'Rua Castelo
    Branco', 'Castelo Branco', '2610-332', '88822212', 1200);
33 Insert into employee values(54445869, 'Ana', '2000-01-18', 'Avenida dos
   d', 'Faro', '8610-332', '88823292', 1200);
34 Insert into employee values(54445870, 'Joana', '2002-01-18', 'Avenida',
   'Tavira', '5231-344', '888232232', 1200);
35
36
37 Insert into phone_number_employee values(12345876,  351911111111);
38 Insert into phone_number_employee values(12456876,  351911111333);
39 Insert into phone_number_employee values(88888888,  351911122333);
40 Insert into phone_number_employee values(11111222,  351922232333);
41 Insert into phone_number_employee values(44440000,  351914441333);
42 Insert into phone_number_employee values(14444876,  351914441333);
43 Insert into phone_number_employee values(55555555,  351914441333);
44 Insert into phone_number_employee values(222000333, 351916641333);
45 Insert into phone_number_employee values(14445876,  351914455533);
46 Insert into phone_number_employee values(14345876,  351914455443);
47 Insert into phone_number_employee values(54445875,  351964455533);
48 Insert into phone_number_employee values(54445867,  351964455588);
49 Insert into phone_number_employee values(54445869,  351964455576);
50 Insert into phone_number_employee values(54445870,  351964455571);
```

```
51
52
53 Insert into doctor values(12345876, 'Feet',[U+FFFD]timdoutor estudou no
   IST', 'doctor1@gmail.com');
54 Insert into doctor values(12456876, 'Hands', 'P[U+FFFDsimo doutor estudou no
    ISEL', 'doctor2@gmail.com');
55 Insert into doctor values(88888888, 'Teeth', 'Excelente doutor estudou
   no IST', 'doctor3@gmail.com');
56 Insert into doctor values(11111222, 'Ears', 'Razo[U+FFFDvel e experiente
   doutor estudou na nova', 'doctor4@gmail.com');
57 Insert into doctor values(44440000, 'Fingers', 'Studied in Harvard', '
   doctor200@gmail.com');
58 Insert into doctor values(14444876, 'Stomach', 'Good trainee', '
   doctor208@gmail.com');
59 Insert into doctor values(55555555, 'Nails', 'Studied in Portugal FCT',
    'doctor209@gmail.com');
60 Insert into doctor values(222000333, 'Dead People', 'The worst I have
   ever seen', 'doctor210@gmail.com');
61
62
63 Insert into permanent_doctor values(12345876, 10);
64 Insert into permanent_doctor values(12456876, 6);
65 Insert into permanent_doctor values(88888888, 8);
66 Insert into permanent_doctor values(11111222, 15);
67
68
69 Insert into trainee_doctor values(44440000, 12456876);
70 Insert into trainee_doctor values(14444876, 12456876);
71 Insert into trainee_doctor values(55555555, 11111222);
72 Insert into trainee_doctor values(222000333, 12456876);
73
74
75 Insert into receptionist values(14445876);
76 Insert into receptionist values(14345876);
77
78
79 Insert into nurse values(54445875);
80 Insert into nurse values(54445867);
81 Insert into nurse values(54445869);
82 Insert into nurse values(54445870);
83
84
```

```
85 Insert into supervision_report values(44440000, '2010-10-10 10:00:00', '
     O trainee anda a brincar com isto', 4);
86 Insert into supervision_report values(14444876, '2018-10-12 11:00:00', '
     Very good doctor', 5);
87 Insert into supervision_report values(55555555, '2019-10-10 19:00:00', '
     Mara is terrible', 1);
88 Insert into supervision_report values(222000333, '2019-10-15 16:00:00',
     'Johnson is not terrible', 1);
89 Insert into supervision_report values(55555555, '2016-10-15 16:00:00', '
     Mara is really bad', 2);
90 Insert into supervision_report values(14444876, '2017-10-12 9:00:00', '
     It was good overall, however in feet part it was insufficient', 4);
91
92
93 insert into appointment values(12456876, '2008-12-3 12:00:00', 'O
     cliente esteve presente no consult[U+FFFD]rio', 12345678);
94 insert into appointment values(12345876, '2008-12-4 10:30:00', 'O
     cliente esteve impaciente', 23456789);
95 insert into appointment values(88888888, '2008-12-6 16:35:00', '
     Confirmado para as 18', 18273645);
96 insert into appointment values(11111222, '2008-12-5 17:00:00', 'O
     cliente era chato que doia', 18273645);
97 insert into appointment values(88888888, '2008-12-8 8:00:00', 'O cliente
      era impossivel', 23456789);
98 insert into appointment values(88888888, '2008-12-6 17:00:00', 'A Ines
     era impossivel', 87333321);
99 insert into appointment values(88888888, '2010-12-6 17:00:00', 'She is
     very sick', 87333321);
100 insert into appointment values(88888888, '2012-12-6 17:00:00', 'She is
     very sick', 87654321);
101 insert into appointment values(12345876, '2019-07-6 17:00:00', 'The
     client was angry', 87333321);
102 insert into appointment values(88888888, '2019-08-6 17:00:00', '
     Confirmed', 43215678);
103 insert into appointment values(12345876, '2019-09-6 17:00:00', 'The
     client attended the appointment', 18273645);
104 insert into appointment values(88888888, '2019-12-6 17:00:00', 'The
     appointment changed date', 87333321);
105 insert into appointment values(12456876, '2019-10-6 17:00:00', 'The
     client was older than expected', 12345678);
106 insert into appointment values(12456876, '2017-10-6 17:00:00', 'The was
     extremely tall', 12345678);
```

```
107 insert into appointment values(12456876, '2019-12-25 17:00:00', 'He was
    close to dying', 18273645);
108 insert into appointment values(88888888, '2019-01-31 17:00:00', N[U+FFFD]
    veio', 87333321);
109 insert into appointment values(11111222, '2019-01-21 17:00:00', 'Nao
    veio outra vez', 87333321);
110 insert into appointment values(88888888, '2019-02-01 17:00:00', N[U+FFFD]
    veio', 87333321);
111 insert into appointment values(88888888, '2019-02-02 17:00:00', N[U+FFFD]
    veio', 87333321);
112 insert into appointment values(88888888, '2019-02-03 17:00:00', N[U+FFFD]
    veio', 87333321);
113 insert into appointment values(88888888, '2019-02-04 17:00:00', N[U+FFFD]
    veio', 87333321);
114 insert into appointment values(88888888, '2019-02-05 17:00:00', N[U+FFFD]
    veio', 87333321);
115 insert into appointment values(88888888, '2019-02-06 17:00:00', N[U+FFFD]
    veio', 87333321);
116 insert into appointment values(88888888, '2019-02-07 17:00:00', N[U+FFFD]
    veio', 87333321);
117 insert into appointment values(11111222, '2007-12-6 17:00:00', 'The
    patient is not nice', 87333321);
118
119
120 insert into consultation values(12456876, '2008-12-3 12:00:00', 'a', 'b'
    , 'c', 'd');
121 insert into consultation values(12345876, '2008-12-4 10:30:00', 'S', 'O'
    , 'A', 'P');
122 insert into consultation values(88888888, '2008-12-6 16:35:00', 'e', '
    gingivitis', 'g', 'h');
123 insert into consultation values(11111222, '2008-12-5 17:00:00', 'l', '
    periodontitis ', 'j', 'i');
124 insert into consultation values(88888888, '2008-12-8 8:00:00', 'm', 'k',
    [U+FFFD] 'p');
125 insert into consultation values(88888888, '2008-12-6 17:00:00', 'z', 'w'
    , 'x', 'y');
126 insert into consultation values(12345876, '2019-07-6 17:00:00', 'z', 'w'
    , 'x', 'y');
127 insert into consultation values(88888888, '2019-08-6 17:00:00', 'z', 'w'
    , 'x', 'y');
128 insert into consultation values(12345876, '2019-09-6 17:00:00', 'z', 'w'
    , 'x', 'y');
```

```
129 insert into consultation values(88888888, '2019-12-6 17:00:00', 'z', 'He
        has gingivitis', 'x', 'y');
130 insert into consultation values(12456876, '2019-10-6 17:00:00', 'z', 'w'
       , 'x', 'y');
131 insert into consultation values(12456876, '2017-10-6 17:00:00', 'z', 'w'
       , 'x', 'y');
132 insert into consultation values(12456876, '2019-12-25 17:00:00', 'g', '
       periodontitis', 'x', 'i');
133 insert into consultation values(11111222, '2007-12-6 17:00:00', 'z', 'w'
       , 'x', 'y');
134
135
136 insert into consultation_assistant values(12456876, '2008-12-3 12:00:00'
       , 54445875);
137 insert into consultation_assistant values(12456876, '2008-12-3 12:00:00'
       , 54445867);
138 insert into consultation_assistant values(12345876, '2008-12-4 10:30:00'
       , 54445867);
139 insert into consultation_assistant values(12345876, '2008-12-4 10:30:00'
       , 54445870);
140 insert into consultation_assistant values(12345876, '2008-12-4 10:30:00'
       , 54445869);
141 insert into consultation_assistant values(88888888, '2008-12-6 16:35:00'
       , 54445867);
142 insert into consultation_assistant values(88888888, '2008-12-6 16:35:00'
       , 54445869);
143 insert into consultation_assistant values(11111222, '2008-12-5 17:00:00'
       , 54445867);
144 insert into consultation_assistant values(11111222, '2008-12-5 17:00:00'
       , 54445869);
145 insert into consultation_assistant values(88888888, '2008-12-8 8:00:00'
       , 54445867);
146 insert into consultation_assistant values(88888888, '2008-12-8 8:00:00'
       , 54445870);
147 insert into consultation_assistant values(12345876, '2019-07-6 17:00:00'
       , 54445867);
148 insert into consultation_assistant values(12345876, '2019-07-6 17:00:00'
       , 54445869);
149 insert into consultation_assistant values(88888888, '2019-08-6 17:00:00'
       , 54445867);
150 insert into consultation_assistant values(88888888, '2019-08-6 17:00:00'
       , 54445875);
```

```
151 insert into consultation_assistant values(12345876, '2019-09-6 17:00:00', 54445867);
152 insert into consultation_assistant values(88888888, '2019-12-6 17:00:00', 54445867);
153 insert into consultation_assistant values(12456876, '2019-10-6 17:00:00', 54445867);
154 insert into consultation_assistant values(88888888, '2008-12-6 17:00:00', 54445867);
155 insert into consultation_assistant values(12456876, '2017-10-6 17:00:00', 54445867);
156 insert into consultation_assistant values(11111222, '2007-12-6 17:00:00', 54445867);
157
158
159 insert into diagnostic_code values(123456000, 'bleeding or swollen gums after brushing or flossing');
160 insert into diagnostic_code values(987654321, 'chronic bad breath');
161 insert into diagnostic_code values(123789456, 'sudden sensitivity to hot and cold temperatures or beverages');
162 insert into diagnostic_code values(987321654, 'pain or toothache');
163 insert into diagnostic_code values(980000000, 'loose teeth');
164 insert into diagnostic_code values(980000001, 'receding gums');
165 insert into diagnostic_code values(980000002, 'pain with chewing or biting');
166 insert into diagnostic_code values(980000003, 'swelling of the face and cheek');
167 insert into diagnostic_code values(980000004, 'cracked or broken teeth');
168 insert into diagnostic_code values(980000005, 'frequent dry mouth');
169 insert into diagnostic_code values(980000006, 'dental cavities, swelling of the face and cheek');
170 insert into diagnostic_code values(980000007, 'dental cavities, cracked or broken teeth');
171 insert into diagnostic_code values(980000008, 'infectious disease, frequent dry mouth');
172 insert into diagnostic_code values(980000009, 'gingivitis');
173
174
175 insert into consultation_diagnostic values(12456876, '2008-12-3 12:00:00', 123456000);
176 insert into consultation_diagnostic values(12456876, '2008-12-3 12:00:00', 980000009);
```

```
177 insert into consultation_diagnostic values(12345876, '2008-12-4 10:30:00
    ', 123456000);
178 insert into consultation_diagnostic values(88888888, '2008-12-6 16:35:00
    ', 123789456);
179 insert into consultation_diagnostic values(11111222, '2008-12-5 17:00:00
    ', 987321654);
180 insert into consultation_diagnostic values(88888888, '2008-12-6 17:00:00
    ', 123456000);
181 insert into consultation_diagnostic values(12345876, '2019-07-6 17:00:00
    ', 980000006);
182 insert into consultation_diagnostic values(88888888, '2019-08-6 17:00:00
    ', 980000007);
183 insert into consultation_diagnostic values(12345876, '2019-09-6 17:00:00
    ', 980000008);
184 insert into consultation_diagnostic values(88888888, '2019-12-6 17:00:00
    ', 980000006);
185 insert into consultation_diagnostic values(12456876, '2019-10-6 17:00:00
    ', 123456000);
186 insert into consultation_diagnostic values(12456876, '2017-10-6 17:00:00
    ', 980000007);
187 insert into consultation_diagnostic values(11111222, '2007-12-6 17:00:00
    ', 980000009);
188
189
190 insert into medication values('Advil', 'Bayer Schering Pharma AG');
191 insert into medication values('Benzamycin', 'Biogen');
192 insert into medication values('Peridex', 'Axcan Pharma');
193 insert into medication values('Atridox', 'Aspen Pharmacare');
194 insert into medication values('Mycostatin', 'Mitsubishi Pharma');
195 insert into medication values('Acidul', 'Novartis');
196 insert into medication values('Valium', 'Servier Laboratories');
197
198
199 insert into prescription values('Benzamycin','Biogen', 12456876, '
    2008-12-3 12:00:00', 123456000, 'Two doses', 'Every twelve hours, for
     1 week');
200 insert into prescription values('Peridex','Axcan Pharma', 12345876, '
    2008-12-4 10:30:00', 123456000, 'Three doses', 'Every five hours, for
     7 week');
201 insert into prescription values('Mycostatin','Mitsubishi Pharma',
    88888888, '2008-12-6 16:35:00', 123789456, 'Two doses', 'Every eight
    hours, for 1 week' );
```

```
202 insert into prescription values('Mycostatin','Mitsubishi Pharma',
    11111222, '2008-12-5 17:00:00', 987321654,'Three doses/Day', 'Every
    five hours, for 7 week') ;
203 insert into prescription values('Benzamycin','Biogen', 88888888,  '
    2008-12-6 17:00:00', 123456000,'Three doses', 'Every five hours, for
    7 week');
204 insert into prescription values('Acidul','Novartis',12345876, '2019-07-6
     17:00:00', 980000006,'Three doses', 'Every eight hours, for 1 week')
    ;
205 insert into prescription values('Atridox','Aspen Pharmacare', 88888888,
    '2019-08-6 17:00:00', 980000007,'Three doses', 'Every five hours, for
     7 week');
206 insert into prescription values('Benzamycin','Biogen', 12345876, '
    2019-09-6 17:00:00', 980000008,'Four doses/Day', 'Every five hours,
    for 7 week');
207 insert into prescription values('Valium','Servier Laboratories',
    88888888, '2019-12-6 17:00:00', 980000006,'Three doses', 'Every five
    hours, for 7 week');
208 insert into prescription values('Peridex','Axcan Pharma', 12456876, '
    2019-10-6 17:00:00', 123456000,'Five doses', 'Every twelve hours, for
     1 week');
209 insert into prescription values('Peridex','Axcan Pharma', 12456876, '
    2017-10-6 17:00:00', 980000007, 'Four doses', 'Every twelve hours,
    for 1 week');
210
211
212 insert into procedure_ values('maxillary molar periapical radiograph', '
    radiography exam');
213 insert into procedure_ values('maxillary incisivus radiograph', '
    radiography exam');
214 insert into procedure_ values('maxillary caninus radiograph', '
    radiography exam');
215 insert into procedure_ values('caninus extraction', 'tooth extractions')
    ;
216 insert into procedure_ values('molar extraction', 'tooth extractions');
217 insert into procedure_ values('upper side map chart', 'dental charting')
    ;
218 insert into procedure_ values('down side map chart', 'dental charting');
219
220
221 insert into procedure_in_consultation values('maxillary molar periapical
     radiograph', 88888888, '2008-12-6 17:00:00', 'Went well');
```

```
222 insert into procedure_in_consultation values('molar extraction',
    88888888, '2008-12-6 17:00:00', 'Went more less');
223 insert into procedure_in_consultation values('down side map chart',
    11111222, '2008-12-5 17:00:00', 'Went very badly');
224 insert into procedure_in_consultation values('molar extraction',
    12345876, '2019-07-6 17:00:00', 'Went very badly');
225 insert into procedure_in_consultation values('down side map chart',
    88888888, '2019-08-6 17:00:00', 'Went very badly');
226 insert into procedure_in_consultation values('down side map chart ',
    12456876, '2008-12-03 12:00:00', 'All the teeth must be removed');
227 insert into procedure_in_consultation values('down side map chart ',
    11111222, '2007-12-6 17:00:00', 'Front teeth must be removed');
228
229
230 insert into teeth values('Right Superior', 1, 'Central Incisor');
231 insert into teeth values('Left Superior', 2, 'Left Incisor');
232 insert into teeth values('Right Inferior', 3, 'Canine');
233 insert into teeth values('Left Inferior', 4, 'First Molar');
234 insert into teeth values('Right Superior', 5, 'Second Molar');
235
236
237 insert into procedure_charting values('down side map chart ', 12456876,
    '2008-12-03 12:00:00', 'Left Inferior', 4, 'The teeth is ok', 5);
238 insert into procedure_charting values('down side map chart ', 12456876,
    '2008-12-03 12:00:00', 'Left Superior', 2, 'The teeth is ok', 6);
239 insert into procedure_charting values('down side map chart ',12456876, '
    2008-12-03 12:00:00', 'Right Inferior', 3, 'The teeth is ok', 4);
240 insert into procedure_charting values('down side map chart ',12456876, '
    2008-12-03 12:00:00', 'Right Superior', 1, 'The teeth is ok', 7);
241 insert into procedure_charting values('down side map chart ', 12456876,
    '2008-12-03 12:00:00', 'Right Superior', 5, 'The teeth is ok', 8);
242 insert into procedure_charting values('down side map chart ', 11111222,
    '2007-12-6 17:00:00', 'Left Inferior', 4, 'The teeth is ok', 2);
243 insert into procedure_charting values('down side map chart ', 11111222,
    '2007-12-6 17:00:00', 'Left Superior', 2, 'The teeth is ok', 3);
244 insert into procedure_charting values('down side map chart ', 11111222,
    '2007-12-6 17:00:00', 'Right Inferior', 3, 'The teeth is ok', 4);
245 insert into procedure_charting values('down side map chart ', 11111222,
    '2007-12-6 17:00:00', 'Right Superior', 1, 'The teeth is ok', 1);
246 insert into procedure_charting values('down side map chart ', 11111222,
    '2007-12-6 17:00:00', 'Right Superior', 5, 'The teeth is ok', 2);
```

3 SQL Queries

3.1 1)

In this query we only needed to connect the client to the appointments and the appointments to the employees. This way we can search the doctor that have the name Jane Sweettooth and the related clients by the appointments he had. Lastly we it is only needed to show its name together with the VAT and the phone from the table phone_number_client.

```
1  SELECT client.VAT, client.name_, phone_number_client.phone
2  FROM client, phone_number_client, appointment, consultation, employee,
       doctor
3  WHERE employee.name_ LIKE 'Jane Sweettooth'
4  AND doctor.VAT = employee.VAT
5  AND client.VAT = appointment.VAT_client
6  AND appointment.VAT_doctor = consultation.VAT_doctor
7  AND appointment.VAT_doctor = doctor.VAT
8  AND client.VAT = phone_number_client.VAT
9  AND appointment.date_timestamp = consultation. date_timestamp
10 ORDER BY client.name_;
```

Output:

VAT	name_	phone
87333321	Inês	351937666421
87333321	Inês	351937666421
18273645	Luis Carlos	351922224321
43215678	Rita	351922254321
23456789	Rui	351917754321

3.2 2)

In this query we needed to get information from the permanent doctors and the trainee doctors, so we needed to access the employee table. We also needed to get the description and score from the supervision report, so that we can select the only trainee doctors with the score below 3 or the description that has a word 'insufficient'.

```
1  SELECT distinct e.name_, t.VAT as VAT_trainee, sr.evaluation, sr.
       description_, e2.name_ as Supervisor_Name, d2.VAT VAT_Supervisor
2  FROM trainee_doctor as t, permanent_doctor as p, supervision_report as
       sr, employee as e, doctor as d, employee as e2, doctor as d2
```

```
WHERE (sr.evaluation < 3 OR sr.description_ LIKE '%insufficient%')
AND sr.VAT = t.VAT
AND t.VAT = d.VAT
AND d.VAT = e.VAT
AND t.supervisor = d2.VAT
AND d2.VAT = e2.VAT
ORDER BY sr.evaluation asc;
```

Output:

name_	VAT_trainee	evaluation	description_	Supervisor_Name	VAT_Supervisor
Mara	55555555	1	Mara is terrible	Jack Sweettath	11111222
Johnson	222000333	1	Johnson is not terrible	Rui	12456876
Mara	55555555	2	Mara is really bad	Jack Sweettath	11111222
Richard	14444876	4	It was good overall, however in feet part it was insufficient	Rui	12456876

3.3 3)

In this query we select the most recent consultation of each client and check if the objective SOAP (SOAP_O) has the description gengivitis or periodontitis in it.

```
Select distinct name_, city, VAT
FROM client as cl, consultation as c, appointment as a
WHERE (SOAP_O LIKE '%gingivitis%' OR SOAP_O LIKE '%periodontitis%')
AND c. VAT_doctor = a.VAT_doctor
AND c.date_timestamp = a. date_timestamp
AND a.VAT_client = cl.VAT
AND  c.date_timestamp IN
(SELECT max(consultation.date_timestamp)
FROM appointment natural join consultation
GROUP BY appointment.VAT_client);
```

Output:

name_	city	VAT
Luis Carlos	Lisboa	18273645
Inês	Gaia	87333321

3.4 4)

We want to present the information Name, VAT and Address of the clients whose appointments were never consultations. We start by selecting the clients that already had at least one consultation by joining the appointment and consultation tables. Then, we selected all the clients that were not listed in the previous table.

```sql
SELECT name_, VAT, street, city, zip
FROM client, appointment
WHERE client.VAT = appointment.VAT_client
AND VAT NOT IN
(SELECT appointment.VAT_client
FROM appointment
NATURAL JOIN consultation);
```

Output:

name_	VAT	street	city	zip
Maria	87654321	Rua do Campo	Rio de Janeiro	2610-062

3.5 5)

In this query we want the number of different medications used to treat a particular condition. This is obtained by joining the prescription table with the diagnostic_code table. Finally we do a Count(Distinct name_) and a Group By id, to count every distinct medication name associated with a particular diagnostic code.

```sql
SELECT  ID, d.description, COUNT( DISTINCT name_) as number
FROM diagnostic_code as d
NATURAL JOIN prescription
GROUP BY id
ORDER BY number asc;
```

Output:

ID	description	number
123789456	sudden sensitivity to hot and cold temperatures or beverages	1
980000008	infectious disease, frequent dry mouth	1
987321654	pain or toothache	1
123456000	bleeding or swollen gums after brushing or flossing	2
980000006	dental cavities, swelling of the face and cheek	2
980000007	dental cavities, cracked or broken teeth	2

3.6 6)

In this query we divide into 8 sub queries where they correspond to count of the nurses, procedures, diagnostic codes and procedures for all consultations in 2019 and for clients under and over 18 years, grouped by the respective table. In the end we did the average for each of them and presented the result.

```sql
select avg(count_nurse_u_18) as avg_nurse_under_18, avg(count_nurse_o_18) as avg_nurse_over_18,
avg(count_proc_u_18) as avg_proc_under_18, avg(count_proc_o_18) as avg_proc_over_18,
avg(count_diag_u_18) as avg_diag_under_18, avg(count_diag_o_18) as avg_diag_over_18,
avg(count_med_u_18) as avg_med_under_18, avg(count_med_o_18) as avg_med_over_18

from
(select count(ca.VAT_nurse) as count_nurse_u_18
from consultation_assistant as ca, appointment as a, client as cl
where ca.VAT_doctor = a.VAT_doctor
and ca.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age <= 18 and extract(year from a.date_timestamp) = 2019
group by ca.VAT_doctor, ca.date_timestamp) as num_n_under_18,

(select count(ca.VAT_nurse) as count_nurse_o_18
from consultation_assistant as ca, appointment as a, client as cl
where ca.VAT_doctor = a.VAT_doctor
and ca.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age > 18 and extract(year from a.date_timestamp) = 2019
group by ca.VAT_doctor, ca.date_timestamp) as num_n_over_18,

(select count(pc.name_) as count_proc_u_18
from procedure_in_consultation as pc, appointment as a, client as cl
where pc.VAT_doctor = a.VAT_doctor
and pc.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age <= 18 and extract(year from a.date_timestamp) = 2019
group by pc.VAT_doctor, pc.date_timestamp) as num_p_u_18,

(select count(pc.name_) as count_proc_o_18
from procedure_in_consultation as pc, appointment as a, client as cl
where pc.VAT_doctor = a.VAT_doctor
and pc.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age > 18 and extract(year from a.date_timestamp) = 2019
group by pc.VAT_doctor, pc.date_timestamp) as num_p_o_18,

```

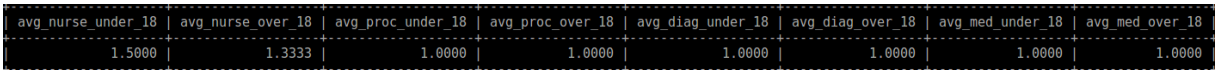
```
(select count(cd.ID) as count_diag_u_18
from consultation_diagnostic as cd, appointment as a, client as cl
where cd.VAT_doctor = a.VAT_doctor
and cd.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age <= 18 and extract(year from a.date_timestamp) = 2019
group by cd.VAT_doctor, cd.date_timestamp) as num_dia_u_18,

(select count(cd.ID) as count_diag_o_18
from consultation_diagnostic as cd, appointment as a, client as cl
where cd.VAT_doctor = a.VAT_doctor
and cd.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age > 18 and extract(year from a.date_timestamp) = 2019
group by cd.VAT_doctor, cd.date_timestamp) as num_dia_o_18,

(select count(p.name_) as count_med_u_18
from prescription as p, appointment as a, client as cl
where p.VAT_doctor = a.VAT_doctor
and p.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age <= 18 and extract(year from a.date_timestamp) = 2019
group by p.VAT_doctor, p.date_timestamp) as num_med_u_18,

(select count(p.name_) as count_med_o_18
from prescription as p, appointment as a, client as cl
where p.VAT_doctor = a.VAT_doctor
and p.date_timestamp = a.date_timestamp
and a.VAT_client = cl.VAT
and cl.age > 18 and extract(year from a.date_timestamp) = 2019
group by p.VAT_doctor, p.date_timestamp) as num_med_o_18;
```

Output:



avg_nurse_under_18	avg_nurse_over_18	avg_proc_under_18	avg_proc_over_18	avg_diag_under_18	avg_diag_over_18	avg_med_under_18	avg_med_over_18
1.5000	1.3333	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

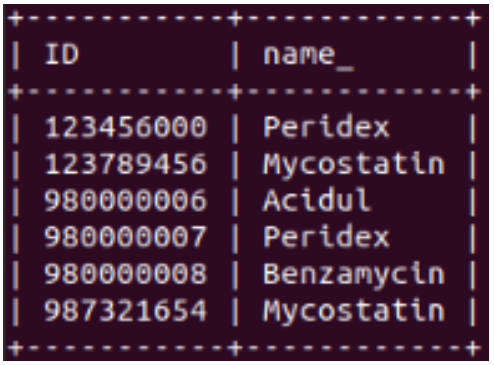
3.7 7)

In this query, we needed to count the most common medication for each dignostic code. We start by selecting the ID of the dagnostic and the name of the medication that was prescribed. Then using count and group by, the number of occurrences of each pair

(ID, name_) is obtained. Since we want to select the maximum count, the values are ordered in a descending way to ensure that the final (group by name_, lab) selects always the maximum.

```
SELECT ID, name_
FROM (SELECT ID , name_ , COUNT(*) as Count
FROM prescription
GROUP BY ID, name_
ORDER BY Count DESC) AS y
GROUP BY ID;
```

Output:



ID	name_
123456000	Peridex
123789456	Mycostatin
980000006	Acidul
980000007	Peridex
980000008	Benzamycin
987321654	Mycostatin

3.8 8)

In this query we used the name "dental cavities"and "infectious diseases"to search for the needed diagnostic codes in order to identify the names and labs of the medication using the prescription table. Once identified, the names and labs are ordered alphabetically.

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

Output:

name_	lab
Acidul	Novartis
Atridox	Aspen Pharmacare
Valium	Servier Laboratories

3.9 9)

In this query, we started by creating a sub query with all the clients that have missed at least one appointment in 2019. Then using the client table, we selected all the needed information (name and adress), from the clients that were not in the subquery created before.

```
SELECT DISTINCT client.name_, client.street, client.city, client.zip
    FROM appointment, client
    WHERE year(date_timestamp) = 2019
    AND client.VAT = appointment.VAT_client
    AND VAT_client NOT IN(
    SELECT  appointment.VAT_client
            FROM appointment
            LEFT JOIN consultation
            ON appointment.date_timestamp=consultation.date_timestamp
            AND appointment.VAT_doctor = consultation.VAT_doctor
            WHERE year(appointment.date_timestamp) = 2019
            AND SOAP_S IS NULL);
```

Output:

name_	street	city	zip
Joao	Praceta da Luz	Lisboa	2610-062
Luis Carlos	Rua do Marquês	Lisboa	2610-350
Rita	Praça da Alegria	Lisboa	1610-035

4 SQL Indexes

Indexing in SQL is used to speed up searching in the database. An index can be used to efficiently find all rows matching some column in your query and then walk through only that subset of the table to find exact matches. If you don't have indexes on any

column in the WHERE clause, the SQL server has to walk through the whole table and check every row to see if it matches, which may be a slow operation on big tables.

In the cases of the first query, to create the index it is needed the tables and columns used on that query. In the first one it is used the both the tables client and phone_number_client and the columns used were VAT, name_ from client and phone from phone_number_client. It is only needed to create for the column name_ in client, as the other columns are primary keys which are already indexes.

In the second query, to create the index it is only needed the column name_ from the table employee and the columns evaluation and description_ from the the supervision report because the VAT from employee and the VAT from the supervision report are primary keys.

The SQL instructions for creating the indexes are the following.

```
/*Index for the first querry*/
create index querry_1_client on client(name_);

/*Index for the second querry*/
create index querry_2_client on employee(name_);
create index querry_2_client on supervision_report(evaluation,
   description_)
```

5 SQL Instructions to Change the Database

5.1 1)

```
update employee
set city = 'Los Angeles', street = 'Hollywood Boulevard'
where name_='Jane Sweettooth';
```

Output:

```
MySQL [ist425312]> select * from employee;
+-----------+----------------+------------+--------------------+---------------+---------+-----------+---------+
| VAT       | name_          | birth_date | street             | city          | zip     | IBAN      | salary  |
+-----------+----------------+------------+--------------------+---------------+---------+-----------+---------+
|  11111222 | Jack Sweettath | 1964-11-15 | Rua da Falagueira  | Lisboa        | 2612-373 | 44577788  | 3000.00 |
|  12345876 | Joao           | 1996-12-08 | Principe Real      | Lisboa        | 2610-062 | 339237212 | 1300.00 |
|  12456876 | Rui            | 1960-12-30 | Praça de Londres   | Lisboa        | 2610-062 | 44423721  | 1300.00 |
|  14345876 | Ana            | 1978-02-18 | Praça das Flores   | Lisboa        | 2615-252 | 66677212  | 1000.00 |
|  14444876 | Richard        | 1995-01-26 | Gatewick           | London        | 2610-355 | 11677788  | 1300.00 |
|  14445876 | Luis           | 1978-01-08 | AmadoraCity        | Lisboa        | 2610-062 | 66637212  | 1300.00 |
|  44440000 | Joseph         | 1965-03-26 | Norwich            | London        | 2610-062 | 66677788  | 1300.00 |
|  54445867 | Hugo           | 1945-01-18 | Rua Castelo Branco | Castelo Branco | 2610-332 | 88822212  | 1200.00 |
|  54445869 | Ana            | 2000-01-18 | Avenida dos d      | Faro          | 8610-332 | 88823292  | 1200.00 |
|  54445870 | Joana          | 2002-01-18 | Avenida            | Tavira        | 5231-344 | 888232232 | 1200.00 |
|  54445875 | Lucas          | 1979-01-18 | Cova da Mora       | Lisboa        | 2610-333 | 88823212  | 1200.00 |
|  55555555 | Mara           | 1965-03-26 | Central Park L     | London        | 2610-072 | 65657788  | 1300.00 |
|  88888888 | Jane Sweettooth | 1963-11-30 | Rua das Conchas    | Lisboa        | 2612-063 | 55577788  | 2500.00 |
| 222000333 | Johnson        | 1965-03-26 | Route 69           | London        | 2610-062 | 96677788  | 1300.00 |
+-----------+----------------+------------+--------------------+---------------+---------+-----------+---------+
MySQL [ist425312]> update employee
    -> set city = 'Los Angeles', street = 'Hollywood Boulevard'
    -> where name_='Jane Sweettooth';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
MySQL [ist425312]> select * from employee;
+-----------+----------------+------------+--------------------+---------------+---------+-----------+---------+
| VAT       | name_          | birth_date | street             | city          | zip     | IBAN      | salary  |
+-----------+----------------+------------+--------------------+---------------+---------+-----------+---------+
|  11111222 | Jack Sweettath | 1964-11-15 | Rua da Falagueira  | Lisboa        | 2612-373 | 44577788  | 3000.00 |
|  12345876 | Joao           | 1996-12-08 | Principe Real      | Lisboa        | 2610-062 | 339237212 | 1300.00 |
|  12456876 | Rui            | 1960-12-30 | Praça de Londres   | Lisboa        | 2610-062 | 44423721  | 1300.00 |
|  14345876 | Ana            | 1978-02-18 | Praça das Flores   | Lisboa        | 2615-252 | 66677212  | 1000.00 |
|  14444876 | Richard        | 1995-01-26 | Gatewick           | London        | 2610-355 | 11677788  | 1300.00 |
|  14445876 | Luis           | 1978-01-08 | AmadoraCity        | Lisboa        | 2610-062 | 66637212  | 1300.00 |
|  44440000 | Joseph         | 1965-03-26 | Norwich            | London        | 2610-062 | 66677788  | 1300.00 |
|  54445867 | Hugo           | 1945-01-18 | Rua Castelo Branco | Castelo Branco | 2610-332 | 88822212  | 1200.00 |
|  54445869 | Ana            | 2000-01-18 | Avenida dos d      | Faro          | 8610-332 | 88823292  | 1200.00 |
|  54445870 | Joana          | 2002-01-18 | Avenida            | Tavira        | 5231-344 | 888232232 | 1200.00 |
|  54445875 | Lucas          | 1979-01-18 | Cova da Mora       | Lisboa        | 2610-333 | 88823212  | 1200.00 |
|  55555555 | Mara           | 1965-03-26 | Central Park L     | London        | 2610-072 | 65657788  | 1300.00 |
|  88888888 | Jane Sweettooth | 1963-11-30 | Hollywood Boulevard | Los Angeles   | 2612-063 | 55577788  | 2500.00 |
| 222000333 | Johnson        | 1965-03-26 | Route 69           | London        | 2610-062 | 96677788  | 1300.00 |
+-----------+----------------+------------+--------------------+---------------+---------+-----------+---------+
```

In this instruction it is assumed that there is only one Jane Sweettooth in the clinic, that is why we used the name to search for the doctor Jane Sweettooth.

5.2 2)

In this instruction, since we did not have 100 appointments for each doctor the number of appointments used is 5, because if the query works for 5 it will also work for 100. We start by selecting the doctors that have more than 5 appointments for 2019 and after that, we update the table employee raising the salary of those doctors.

```
update employee
set salary = 1.05*salary
where VAT in
(select VAT_doctor
from appointment
where extract(year from appointment.date_timestamp) = 2019
group by VAT_doctor
having count(VAT_doctor) > 5);
```

Output:

```
MySQL [ist425312]> select * from appointment where year(date_timestamp)=2019;
+------------+---------------------+-----------------------------------+------------+
| VAT_doctor | date_timestamp      | description_                      | VAT_client |
+------------+---------------------+-----------------------------------+------------+
|   11111222 | 2019-01-21 17:00:00 | Nao veio outra vez                |   87333321 |
|   12345876 | 2019-07-06 17:00:00 | The client was angry              |   87333321 |
|   12345876 | 2019-09-06 17:00:00 | The client attended the appointment |   18273645 |
|   12456876 | 2019-10-06 17:00:00 | The client was older than expected |   12345678 |
|   12456876 | 2019-12-25 17:00:00 | He was close to dying             |   18273645 |
|   88888888 | 2019-01-31 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-02-01 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-02-02 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-02-03 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-02-04 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-02-05 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-02-06 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-02-07 17:00:00 | Não veio                          |   87333321 |
|   88888888 | 2019-08-06 17:00:00 | Confirmed                         |   43215678 |
|   88888888 | 2019-12-06 17:00:00 | The appointment changed date      |   87333321 |
+------------+---------------------+-----------------------------------+------------+
MySQL [ist425312]> select * from employee where VAT=88888888;
+----------+----------------+------------+----------------+--------+----------+----------+---------+
| VAT      | name_          | birth_date | street         | city   | zip      | IBAN     | salary  |
+----------+----------------+------------+----------------+--------+----------+----------+---------+
| 88888888 | Jane Sweettooth | 1963-11-30 | Rua das Conchas | Lisboa | 2612-063 | 55577788 | 2500.00 |
+----------+----------------+------------+----------------+--------+----------+----------+---------+
MySQL [ist425312]> update employee
    -> set salary = 1.05*salary
    -> where VAT in
    -> (select VAT_doctor
    -> from appointment
    -> where extract(year from appointment.date_timestamp) = 2019
    -> group by VAT_doctor
    -> having count(VAT_doctor) > 5);
Query OK, 1 row affected (54.80 sec)
MySQL [ist425312]> select * from employee where VAT=88888888;
+----------+----------------+------------+----------------+--------+----------+----------+---------+
| VAT      | name_          | birth_date | street         | city   | zip      | IBAN     | salary  |
+----------+----------------+------------+----------------+--------+----------+----------+---------+
| 88888888 | Jane Sweettooth | 1963-11-30 | Rua das Conchas | Lisboa | 2612-063 | 55577788 | 2625.00 |
+----------+----------------+------------+----------------+--------+----------+----------+---------+
```

The doctor whose id is 88888888 has more than 5 appointments for 2019 and his salary is 2500.00 euro. After running the query it is possible to see that his salary raised 5% to 2625.00 euro, confirming that the query is correct.

5.3 3)

```
1 delete from employee where name_='Jane Sweettooth';
2
3 delete from procedure_
4 where name_ not in
5 (select distinct name_ from procedure_in_consultation);
6
7 delete from diagnostic_code
8 where ID not in
9 (select distinct ID from consultation_diagnostic);
```

Output:

```
MySQL [ist425312]> select * from employee;
+-----------+----------------+------------+--------------------+---------------+---------+----------+--------+
| VAT       | name_          | birth_date | street             | city          | zip     | IBAN     | salary |
+-----------+----------------+------------+--------------------+---------------+---------+----------+--------+
|  11111222 | Jack Sweettath | 1964-11-15 | Rua da Falagueira  | Lisboa        | 2612-373 | 44577788 | 3000.00 |
|  12345876 | Joao           | 1996-12-08 | Principe Real      | Lisboa        | 2610-062 | 339237212 | 1300.00 |
|  12456876 | Rui            | 1960-12-30 | Praça de Londres   | Lisboa        | 2610-062 | 44423721 | 1300.00 |
|  14345876 | Ana            | 1978-02-18 | Praça das Flores   | Lisboa        | 2615-252 | 66677212 | 1000.00 |
|  14444876 | Richard        | 1995-01-26 | Gatewick           | London        | 2610-355 | 11677788 | 1300.00 |
|  14445876 | Luis           | 1978-01-08 | AmadoraCity        | Lisboa        | 2610-062 | 66637212 | 1300.00 |
|  44440000 | Joseph         | 1965-03-26 | Norwich            | London        | 2610-062 | 66677788 | 1300.00 |
|  54445867 | Hugo           | 1945-01-18 | Rua Castelo Branco | Castelo Branco | 2610-332 | 88822212 | 1200.00 |
|  54445869 | Ana            | 2000-01-18 | Avenida dos d      | Faro          | 8610-332 | 88823292 | 1200.00 |
|  54445870 | Joana          | 2002-01-18 | Avenida            | Tavira        | 5231-344 | 888232232 | 1200.00 |
|  54445875 | Lucas          | 1979-01-18 | Cova da Mora       | Lisboa        | 2610-333 | 88823212 | 1200.00 |
|  55555555 | Mara           | 1965-03-26 | Central Park L     | London        | 2610-072 | 65657788 | 1300.00 |
|  88888888 | Jane Sweettooth | 1963-11-30 | Rua das Conchas    | Lisboa        | 2612-063 | 55577788 | 2500.00 |
| 222000333 | Johnson        | 1965-03-26 | Route 69           | London        | 2610-062 | 96677788 | 1300.00 |
+-----------+----------------+------------+--------------------+---------------+---------+----------+--------+
14 rows in set (0.00 sec)

MySQL [ist425312]> delete from employee where name_='Jane Sweettooth';
Query OK, 1 row affected (0.00 sec)

MySQL [ist425312]>
MySQL [ist425312]> delete from procedure_
    -> where name_ not in
    -> (select distinct name_ from procedure_in_consultation);
Query OK, 5 rows affected (0.00 sec)

MySQL [ist425312]>
MySQL [ist425312]> delete from diagnostic_code
    -> where ID not in
    -> (select distinct ID from consultation_diagnostic);
Query OK, 8 rows affected (0.00 sec)

MySQL [ist425312]> select * from employee;
+-----------+----------------+------------+--------------------+---------------+---------+----------+--------+
| VAT       | name_          | birth_date | street             | city          | zip     | IBAN     | salary |
+-----------+----------------+------------+--------------------+---------------+---------+----------+--------+
|  11111222 | Jack Sweettath | 1964-11-15 | Rua da Falagueira  | Lisboa        | 2612-373 | 44577788 | 3000.00 |
|  12345876 | Joao           | 1996-12-08 | Principe Real      | Lisboa        | 2610-062 | 339237212 | 1300.00 |
|  12456876 | Rui            | 1960-12-30 | Praça de Londres   | Lisboa        | 2610-062 | 44423721 | 1300.00 |
|  14345876 | Ana            | 1978-02-18 | Praça das Flores   | Lisboa        | 2615-252 | 66677212 | 1000.00 |
|  14444876 | Richard        | 1995-01-26 | Gatewick           | London        | 2610-355 | 11677788 | 1300.00 |
|  14445876 | Luis           | 1978-01-08 | AmadoraCity        | Lisboa        | 2610-062 | 66637212 | 1300.00 |
|  44440000 | Joseph         | 1965-03-26 | Norwich            | London        | 2610-062 | 66677788 | 1300.00 |
|  54445867 | Hugo           | 1945-01-18 | Rua Castelo Branco | Castelo Branco | 2610-332 | 88822212 | 1200.00 |
|  54445869 | Ana            | 2000-01-18 | Avenida dos d      | Faro          | 8610-332 | 88823292 | 1200.00 |
|  54445870 | Joana          | 2002-01-18 | Avenida            | Tavira        | 5231-344 | 888232232 | 1200.00 |
|  54445875 | Lucas          | 1979-01-18 | Cova da Mora       | Lisboa        | 2610-333 | 88823212 | 1200.00 |
|  55555555 | Mara           | 1965-03-26 | Central Park L     | London        | 2610-072 | 65657788 | 1300.00 |
| 222000333 | Johnson        | 1965-03-26 | Route 69           | London        | 2610-062 | 96677788 | 1300.00 |
+-----------+----------------+------------+--------------------+---------------+---------+----------+--------+
13 rows in set (0.00 sec)
```

To delete all records related to the doctor Jane Sweettooth, it is only necessary to give one command to delete the employee that has a name Jane Sweettooth (we considered that there is only one employee named Jane Swseettooth in the clinic) because all foreign keys related with employee have the 'ON DELETE CASCADE' which deletes all records recorded to the one deleted.

After that, it is also deleted the procedures and diagnostic codes that were only related with that doctor, as they are strong entities they are not related with the employee, therefore have to be deleted separately.

5.4 4)

In this query, we start by inserting a new diagnostic corresponding to the new diagnosis "Periodentis"and we find the ID corresponding to the "ginvivitis"diagnostic. Then we select every dental charting that has an average gap above 4 and replace the diagnostic id's that correspond to "gengivitis"by the id related the "Periodentis"diagnostic code.

```
1 insert into diagnostic_code values(980000010, 'Periodontitis');
2 update consultation_diagnostic as d
3 set ID = (select ID from diagnostic_code where description = '
    Periodontitis')
```

```
where d.ID IN (select ID from diagnostic_code where description ='
    gingivitis')
and exists(select d.VAT_doctor, d.ID from
procedure_charting as c
where d.VAT_doctor = c.VAT and d.date_timestamp = c.date_timestamp
group by d.VAT_doctor, d.date_timestamp
having avg(measure) > 4);
```

Output:

```
MySQL [ist425312]> select * from consultation_diagnostic where ID=980000009;
+------------+---------------------+-----------+
| VAT_doctor | date_timestamp      | ID        |
+------------+---------------------+-----------+
|   11111222 | 2007-12-06 17:00:00 | 980000009 |
|   12456876 | 2008-12-03 12:00:00 | 980000009 |
+------------+---------------------+-----------+
2 rows in set (0.00 sec)
MySQL [ist425312]> update consultation_diagnostic as d
    -> set ID = (select ID from diagnostic_code where description = 'Periodontitis')
    -> where d.ID IN (select ID from diagnostic_code where description ='gingivitis')
    -> and exists(select d.VAT_doctor, d.ID from
    -> procedure_charting as c
    -> where d.VAT_doctor = c.VAT and d.date_timestamp = c.date_timestamp
    -> group by d.VAT_doctor, d.date_timestamp
    -> having avg(measure) > 4);
Query OK, 1 row affected (0.00 sec)
MySQL [ist425312]> select * from consultation_diagnostic;
+------------+---------------------+-----------+
| VAT_doctor | date_timestamp      | ID        |
+------------+---------------------+-----------+
|   12345876 | 2008-12-04 10:30:00 | 123456000 |
|   12456876 | 2008-12-03 12:00:00 | 123456000 |
|   12456876 | 2019-10-06 17:00:00 | 123456000 |
|   88888888 | 2008-12-06 17:00:00 | 123456000 |
|   88888888 | 2008-12-06 16:35:00 | 123789456 |
|   12345876 | 2019-07-06 17:00:00 | 980000006 |
|   88888888 | 2019-12-06 17:00:00 | 980000006 |
|   12456876 | 2017-10-06 17:00:00 | 980000007 |
|   88888888 | 2019-08-06 17:00:00 | 980000007 |
|   12345876 | 2019-09-06 17:00:00 | 980000008 |
|   11111222 | 2007-12-06 17:00:00 | 980000009 |
|   12456876 | 2008-12-03 12:00:00 | 980000010 |
|   11111222 | 2008-12-05 17:00:00 | 987321654 |
+------------+---------------------+-----------+
```

To update all consultation diagnostics to all clients that had an average of the dental gap above 4 millimeters, it is only needed to update. We can see that the consultation diagnostic ID selected in the above image has changed to 980000010.

6 Views

6.1 1)

```
create view dim_date
as
(select c.date_timestamp,
extract(year from c.date_timestamp) as year,
extract(month from c.date_timestamp) as month,
extract(day from c.date_timestamp) as day
```

```
from consultation as c);

/*To show the view*/

select * from dim_date
```

Output:

date_timestamp	year	month	day
2008-12-05 17:00:00	2008	12	5
2008-12-04 10:30:00	2008	12	4
2019-07-06 17:00:00	2019	7	6
2019-09-06 17:00:00	2019	9	6
2008-12-03 12:00:00	2008	12	3
2017-10-06 17:00:00	2017	10	6
2019-10-06 17:00:00	2019	10	6
2019-12-25 17:00:00	2019	12	25
2008-12-06 16:35:00	2008	12	6
2008-12-06 17:00:00	2008	12	6
2008-12-08 08:00:00	2008	12	8
2019-08-06 17:00:00	2019	8	6
2019-12-06 17:00:00	2019	12	6

6.2 2)

```
create view dim_date
as
(select c.date_timestamp,
extract(year from c.date_timestamp) as year,
extract(month from c.date_timestamp) as month,
extract(day from c.date_timestamp) as day
from consultation as c);

/*To show the view*/

select * from dim_date
```

Output:

VAT	gender	age
12345678	Male	23
18273645	Male	22
23456789	Male	40
43215678	Female	103
45678123	Male	30
87333321	Female	14
87654321	Female	40
89123456	Male	89

6.3 3)

```
create view dim_date
as
(select c.date_timestamp,
extract(year from c.date_timestamp) as year,
extract(month from c.date_timestamp) as month,
extract(day from c.date_timestamp) as day
from consultation as c);

/*To show the view*/

select * from dim_date
```

Output:

zip	city
2610-062	Lisboa
2610-350	Lisboa
2610-062	Lisboa
1610-035	Lisboa
2690-052	Lisboa
2610-062	Gaia
2610-062	Rio de Janeiro
2610-062	Lisboa

6.4 4)

Since the MySQL doesn't allow subqueries inside a view we used Left Outer Joins to get all the needed information. The tables used were procedure_in_consultation, appointment, consultation_diagnostic, prescription, client. Besides, the views previosly defined, dim_date, dim_client and dim_location_client were also used to obtain the VAt, zip from the client and date for each consultation. For counting the Procedures, Diagnostics and Medications we used Count(Distinct) since we only want to count the different elements for each consultation.

```
create view facts_consults as
SELECT dim_client.VAT, dim_location_client.zip, dim_date.date_timestamp
    as date, Count(DISTINCT procedure_in_consultation.name_)
as Number_Procedures, Count(DISTINCT consultation_diagnostic.ID) as
    Number_Diagnostics, Count(DISTINCT prescription.name_) as
    Number_Medications
FROM dim_client, dim_location_client, dim_date
LEFT OUTER JOIN procedure_in_consultation ON dim_date.date_timestamp =
    procedure_in_consultation.date_timestamp
LEFT OUTER JOIN consultation_diagnostic ON consultation_diagnostic.
    date_timestamp=dim_date.date_timestamp
LEFT OUTER JOIN prescription ON prescription.date_timestamp = dim_date.
    date_timestamp
JOIN client AS c
JOIN appointment AS a
WHERE dim_client.VAT=c.VAT
AND dim_location_client.zip = c.zip
AND dim_client.VAT=a.VAT_client
AND dim_date.date_timestamp=a.date_timestamp
GROUP BY dim_client.VAT,dim_date.date_timestamp;

/*To show the view*/

select * from facts_consults;
```

Output:

VAT	zip	date	Number_Procedures	Number_Diagnostics	Number_Medications
12345678	2610-062	2008-12-03 12:00:00	0	2	1
12345678	2610-062	2017-10-06 17:00:00	0	1	1
12345678	2610-062	2019-10-06 17:00:00	0	1	1
18273645	2610-350	2008-12-05 17:00:00	1	1	1
18273645	2610-350	2008-12-06 16:35:00	0	1	1
18273645	2610-350	2019-09-06 17:00:00	0	1	1
18273645	2610-350	2019-12-25 17:00:00	0	0	0
23456789	2610-062	2008-12-04 10:30:00	0	1	1
23456789	2610-062	2008-12-08 08:00:00	0	0	0
43215678	1610-035	2019-08-06 17:00:00	1	1	1
87333321	2610-062	2008-12-06 17:00:00	2	1	1
87333321	2610-062	2019-07-06 17:00:00	1	1	1
87333321	2610-062	2019-12-06 17:00:00	0	1	1