

# Sistemas de Informação e Bases de Dados

## Assignment 2 – Implementing the Database

Bernardo Gonilho nº 84016  
Diogo Alves nº 86980  
Diogo Moura nº 86976  
Grupo nº 11

```
1  drop table if exists procedure_radiology;
2  drop table if exists procedure_charting;
3  drop table if exists teeth;
4  drop table if exists procedure_in_consultation;
5  drop table if exists procedure_;
6  drop table if exists prescription;
7  drop table if exists medication;
8  drop table if exists consultation_diagnostic;
9  drop table if exists diagnostic_code_relation;
10 drop table if exists diagnostic_code;
11 drop table if exists consultation_assistant;
12 drop table if exists consultation;
13 drop table if exists appointment;
14 drop table if exists supervision_report;
15 drop table if exists trainee_doctor;
16 drop table if exists permanent_doctor;
17 drop table if exists phone_number_client;
18 drop table if exists client;
19 drop table if exists nurse;
20 drop table if exists doctor;
21 drop table if exists doctor;
22 drop table if exists receptionist;
23 drop table if exists phone_number_employee;
24 drop table if exists employee;
25
26 create table employee(
27     VAT varchar(20) not null,
28     name varchar(255) not null,
29     birth_date date,
30     street varchar(255),
31     city varchar(255),
32     zip varchar(8),
33     IBAN char(25) not null,
34     salary numeric(20,2),
35     unique(IBAN),
36     check(salary >= 0),
37     primary key(VAT)
38 );
39 --IC: All employees are either receptionists, nurses or doctors
40
41 create table phone_number_employee(
42     VAT varchar(20) not null,
43     phone varchar(20),
44     primary key(VAT, phone),
45     foreign key(VAT)
46         references employee(VAT)
47         on delete cascade
48         on update cascade
49 );
50
51 create table receptionist(
52     VAT varchar(20) not null,
53     primary key(VAT),
54     foreign key(VAT)
55         references employee(VAT)
56         on delete cascade
57         on update cascade
58 );
59
60
61 create table doctor(
62     VAT varchar(20) not null,
63     specialization varchar(255),
64     biography varchar(8000),
65     email varchar(255) not null,
66     unique(email),
67     primary key(VAT),
68     foreign key(VAT)
69         references employee(VAT)
70         on delete cascade
71         on update cascade
72 );
```

```
73  --IC: All doctors are either trainees or permanent
74
75
76  create table nurse(
77      VAT varchar(20) not null,
78      primary key(VAT),
79      foreign key(VAT)
80          references employee(VAT)
81          on delete cascade
82          on update cascade
83
84  );
85
86  create table client(
87      VAT varchar(20) not null,
88      name varchar(255) not null,
89      birth_date date not null,
90      street varchar(255),
91      city varchar(255),
92      zip varchar(8),
93      gender varchar(20),
94      age integer not null,
95      primary key(VAT),
96      check(age>0),
97      check(gender in ('Male', 'Female'))
98  );
99  --IC: Age is derived from the birth date
100
101  create table phone_number_client(
102      VAT varchar(20) not null,
103      phone varchar(20),
104      primary key(VAT, phone),
105      foreign key(VAT)
106          references client(VAT)
107          on delete cascade
108          on update cascade
109  );
110
111  create table permanent_doctor(
112      VAT varchar(20) not null,
113      years integer,
114      primary key(VAT),
115      foreign key(VAT)
116          references doctor(VAT)
117          on delete cascade
118          on update cascade
119  );
120
121
122  create table trainee_doctor(
123      VAT varchar(20) not null,
124      supervisor varchar(20) not null,
125      primary key(VAT),
126      foreign key(VAT)
127          references doctor(VAT)
128          on delete cascade
129          on update cascade,
130      foreign key(supervisor)
131          references permanent_doctor(VAT)
132          on delete cascade
133          on update cascade
134  );
135
136  create table supervision_report(
137      VAT varchar(20) not null,
138      date_timestamp datetime not null,
139      description varchar(8000),
140      evaluation integer not null,
141      check(evaluation >= 1 and evaluation <= 5),
142      primary key(VAT, date_timestamp),
143      foreign key(VAT)
144          references trainee_doctor(VAT)
```

```

145         on delete cascade
146         on update cascade
147     ) ENGINE=MyISAM;
148
149     create table appointment(
150         VAT_doctor varchar(20) not null,
151         date_timestamp datetime not null,
152         description varchar(8000),
153         VAT_client varchar(20) not null,
154         primary key(VAT_doctor, date_timestamp),
155         foreign key(VAT_doctor)
156             references doctor(VAT)
157             on delete cascade
158             on update cascade,
159         foreign key(VAT_client)
160             references client(VAT)
161             on delete cascade
162             on update cascade
163     );
164
165     create table consultation(
166         VAT_doctor varchar(20) not null,
167         date_timestamp datetime not null,
168         SOAP_S varchar(8000),
169         SOAP_O varchar(8000),
170         SOAP_A varchar(8000),
171         SOAP_P varchar(8000),
172         primary key(VAT_doctor, date_timestamp),
173         foreign key(VAT_doctor, date_timestamp)
174             references appointment(VAT_doctor, date_timestamp)
175             on delete cascade
176             on update cascade
177     );
178     --IC: Consultations are always assigned to at least one assistant nurse
179
180     create table consultation_assistant(
181         VAT_doctor varchar(20) not null,
182         date_timestamp datetime not null,
183         VAT_nurse varchar(20) not null,
184         primary key(VAT_doctor, date_timestamp, VAT_nurse),
185         foreign key(VAT_doctor, date_timestamp)
186             references consultation(VAT_doctor, date_timestamp)
187             on delete cascade
188             on update cascade,
189         foreign key(VAT_nurse)
190             references nurse(VAT)
191             on delete cascade
192             on update cascade
193     );
194
195     create table diagnostic_code(
196         ID varchar(20) not null,
197         description varchar(255),
198         primary key(ID)
199     );
200
201
202     create table diagnostic_code_relation(
203         ID1 varchar(20) not null,
204         ID2 varchar(20) not null,
205         type varchar(255),
206         primary key(ID1, ID2),
207         foreign key(ID1)
208             references diagnostic_code(ID)
209             on delete cascade
210             on update cascade,
211         foreign key(ID2)
212             references diagnostic_code(ID)
213             on delete cascade
214             on update cascade
215     );
216

```

```
217 create table consultation_diagnostic(  
218     VAT_doctor varchar(20) not null,  
219     date_timestamp datetime not null,  
220     ID varchar(20) not null,  
221     primary key(VAT_doctor, date_timestamp, ID),  
222     foreign key(VAT_doctor, date_timestamp)  
223         references consultation(VAT_doctor, date_timestamp)  
224         on delete cascade  
225         on update cascade,  
226     foreign key(ID)  
227         references diagnostic_code(ID)  
228         on delete cascade  
229         on update cascade  
230 );  
231  
232 create table medication(  
233     name varchar(255) not null,  
234     lab varchar(255) not null,  
235     primary key(name, lab)  
236 );  
237  
238  
239 create table prescription(  
240     name varchar(255) not null,  
241     lab varchar(255) not null,  
242     VAT_doctor varchar(20) not null,  
243     date_timestamp datetime not null,  
244     ID varchar(20) not null,  
245     dosage varchar(20),  
246     description varchar(255),  
247     primary key(name, lab, VAT_doctor, date_timestamp, ID),  
248     foreign key(VAT_doctor, date_timestamp, ID)  
249         references consultation_diagnostic(VAT_doctor, date_timestamp, ID)  
250         on delete cascade  
251         on update cascade,  
252     foreign key(name, lab)  
253         references medication(name, lab)  
254         on delete cascade  
255         on update cascade  
256 );  
257  
258  
259 create table procedure_(  
260     name varchar(255) not null,  
261     type varchar(255),  
262     primary key(name)  
263 );  
264  
265  
266 create table procedure_in_consultation(  
267     name varchar(255) not null,  
268     VAT_doctor varchar(20) not null,  
269     date_timestamp datetime not null,  
270     description varchar(255),  
271     primary key(name, VAT_doctor, date_timestamp),  
272     foreign key(name)  
273         references procedure_(name)  
274         on delete cascade  
275         on update cascade,  
276     foreign key(VAT_doctor, date_timestamp)  
277         references consultation(VAT_doctor, date_timestamp)  
278         on delete cascade  
279         on update cascade  
280 );  
281  
282 create table procedure_radiology(  
283     name varchar(255) not null,  
284     file varchar(255) not null,  
285     VAT_doctor varchar(20) not null,  
286     date_timestamp datetime not null,  
287     primary key(name, file, VAT_doctor, date_timestamp),  
288     foreign key(name, VAT_doctor, date_timestamp)
```

```

289         references procedure_in_consultation(name, VAT_doctor,
290         date_timestamp)
291         on delete cascade
292         on update cascade
293     );
294     create table teeth(
295         quadrant integer not null,
296         number_ integer not null,
297         name char(255),
298         primary key(quadrant, number_)
299     );
300
301     create table procedure_charting(
302         name varchar(255) not null,
303         VAT varchar(20) not null,
304         date_timestamp datetime not null,
305         quadrant integer not null,
306         number_ integer not null,
307         description char(255),
308         measure float(24) not null,
309         primary key(name, VAT, date_timestamp, quadrant, number_),
310         foreign key(name, VAT, date_timestamp)
311         references procedure_in_consultation(name, VAT_doctor,
312         date_timestamp)
313         on delete cascade
314         on update cascade,
315         foreign key(quadrant, number_)
316         references teeth(quadrant, number_)
317         on delete cascade
318         on update cascade
319     );
320
321
322     insert into employee values ('111111111', 'Jane Sweettooth', '1980-10-01',
323     'Wessex Street', 'Centregoldlon', '8283-128',
324     'PT5091039402930192019464817', 8000.00);
325     insert into employee values ('222222222', 'Wilbur Wonka', '1964-01-17',
326     'Devon Street', 'Dergrandfney', '3829-192', 'PT5002830291029302910293029',
327     9000.00);
328     insert into employee values ('333333333', 'Sally Fernandez', '1987-03-29',
329     'Hanson Street', 'Saint Marys City', '1023-134',
330     'PT5015465135549871563214563', 5000.00);
331     insert into employee values ('444444444', 'Joseph Mcdonald', '1982-11-19',
332     'Whitley Street', 'Orem', '3483-461', 'PT5065478213654785236145963',
333     4500.00);
334     insert into employee values ('555555555', 'Samuel Stevenson', '1975-12-12',
335     'Charlotte Street', 'Compton', '1345-422', 'PT5023014520369874520145698',
336     3000.00);
337     insert into employee values ('666666666', 'Otis Burns', '1985-05-11', 'Eden
338     Street', 'Ashtabula', '1345-134', 'PT5012023655410023654789632', 3000.00);
339     insert into employee values ('777777777', 'Roxanne Davidson', '1990-02-28',
340     'Tweed Street', 'Americus', '5432-523', 'PT5074532102301489602365041',
341     1500.00);
342
343     insert into phone_number_employee values ('111111111', '968192849');
344     insert into phone_number_employee values ('111111111', '927123781');
345     insert into phone_number_employee values ('222222222', '961029301');
346     insert into phone_number_employee values ('333333333', '912039139');
347     insert into phone_number_employee values ('444444444', '936541256');
348     insert into phone_number_employee values ('555555555', '968846514');
349     insert into phone_number_employee values ('666666666', '915455145');
350     insert into phone_number_employee values ('777777777', '965521545');
351
352     insert into receptionist values ('777777777');
353
354     insert into doctor values ('111111111', 'Pediatric dentistry',
355     'xxxxxxxxxxxxxxxx', 'jane.sweettooth@hotmail.com');
356     insert into doctor values ('222222222', 'Endodontics', 'xxxxxxxxxxxxxxxx',
357     'wilbur.wonka@yahoo.com');
358     insert into doctor values ('333333333', 'Oral and maxillofacial surgery',

```

```

'xxxxxxxxxxxxxxxx', 'sallyfernandez87@gmail.com');
344 insert into doctor values ('444444444', 'Anesthesiology', 'xxxxxxxxxxxxxxxx',
'thejoemcdonald@gmail.com');
345
346 insert into nurse values('555555555');
347 insert into nurse values('666666666');
348
349 insert into client values('888888888','George Michael',
'1963-06-25','Brooklyn Street','Winslow','9123-123','Male', 56);
350 insert into client values('999999999','David Bowie',
'1947-01-08','Broomhill Street','Hastings','2245-442','Male', 72);
351 insert into client values('101010101','Paul McCartney', '1942-06-18','Lower
Street','McAllen','5346-134','Male', 77);
352 insert into client values('110110110','Bryan Adams', '1959-11-05','Edmund
Street','Rolla','3542-133','Male', 60);
353 insert into client values('121212121','Celine Dion', '1968-03-30','Third
Street','Joplin','3455-245','Female', 51);
354 insert into client values('131313131','Marvin Gaye', '1939-04-02','St
Peters Street','Fillmore','5743-532','Male', 80);
355 insert into client values('141414141','Bonnie Tyler',
'1951-06-08','Avondale Street','Bloomington','3456-245','Female', 68);
356 insert into client values('151515151','Stevie Wonder', '1950-05-13','Old
Mill Street','Lynchburg','4562-623','Male', 69);
357 insert into client values('161616161','John Lennon',
'1940-10-09','Homefield Street','Northampton','4653-645','Male', 79);
358 insert into client values('171717171','Freddie Mercury',
'1946-09-05','Rosewood Street','Towson','3243-453','Male', 73);
359 insert into client values('181818181','Elton John', '1947-03-25','Smith
Street','Tallahassee','1246-654','Male', 72);
360 insert into client values('191919191','Mick Jagger', '1943-05-26','Warren
Street','San Marino','5674-765','Male', 76);
361 insert into client values('202020202','Morgan Bush', '2001-05-26','Moorside
Street','Northampton','3442-765','Male', 18);
362 insert into client values('212121212','Abraham Walters', '2001-05-26','Teal
Street','Monett','3432-123','Male', 18);
363 insert into client values('220220220','Sara Wolf', '2001-05-26','Bouverie
Street','Baton Rouge','4328-134','Female', 18);
364 insert into client values('232323232','Felicity Morgan',
'2005-05-26','Newport Street','Thaxted','5234-765','Female', 14);
365 insert into client values('242424242','Sabrina Adams',
'2004-05-26','Ashleigh Street','Beckenham','1344-765','Female', 15);
366 insert into client values('252525252','Bruce Reese', '2003-05-26','Addison
Street','Westfield','4328-531','Male', 16);
367 insert into client values('262626262','Harriet Daniels',
'2002-05-26','Ingles Street','Needles','4134-765','Female', 17);
368 insert into client values('272727272','May Torres', '2002-05-26','Moseley
Street','Norris','5325-765','Female', 17);
369 insert into client values('282828282','Shane Reyes', '2010-05-26','Belward
Street','Uxbridge','4134-123','Male', 9);
370 insert into client values('292929292','Evangeline Chapman',
'2015-05-26','Moorside Street','Silssaltquay','1343-132','Female', 4);
371
372 insert into phone_number_client values ('888888888','964549625');
373 insert into phone_number_client values ('888888888','918739182');
374 insert into phone_number_client values ('999999999','918294719');
375 insert into phone_number_client values ('101010101','930192843');
376 insert into phone_number_client values ('110110110','961381324');
377 insert into phone_number_client values ('110110110','961238100');
378 insert into phone_number_client values ('121212121','962341569');
379 insert into phone_number_client values ('131313131','915432599');
380 insert into phone_number_client values ('141414141','935432156');
381 insert into phone_number_client values ('151515151','914565431');
382 insert into phone_number_client values ('161616161','961249828');
383 insert into phone_number_client values ('171717171','931345657');
384 insert into phone_number_client values ('181818181','911324551');
385 insert into phone_number_client values ('181818181','981239791');
386 insert into phone_number_client values ('191919191','921435344');
387 insert into phone_number_client values ('212121212','901238192');
388 insert into phone_number_client values ('220220220','971237814');
389 insert into phone_number_client values ('232323232','941273882');
390 insert into phone_number_client values ('262626262','931828315');
391 insert into phone_number_client values ('272727272','961238128');

```

```

392 insert into phone_number_client values ('282828282','928123812');
393
394 insert into permanent_doctor values('111111111', 12);
395 insert into permanent_doctor values('222222222', 20);
396
397 insert into trainee_doctor values('333333333','222222222');
398 insert into trainee_doctor values('444444444','111111111');
399
400 insert into supervision_report values('333333333', '2017-01-01 18:00:00',
'xxxxxxxx', 2);
401 insert into supervision_report values('333333333', '2018-01-01 18:00:00',
'xxxxxxxx', 3);
402 insert into supervision_report values('333333333', '2019-01-01 18:00:00',
'xxxxxxx insufficient xxxxx', 4);
403 insert into supervision_report values('444444444', '2017-02-02 18:00:00',
'xxxxxxxx', 4);
404 insert into supervision_report values('444444444', '2018-02-02 18:00:00',
'xxxxxx insufficient xxxxx', 3);
405 insert into supervision_report values('444444444', '2019-02-1-02 18:00:00',
'xxxxxxxxxxxx', 2);
406
407 insert into appointment values('333333333', '2019-01-01 10:00:00',
'xxxxxxxx', '888888888');
408 insert into appointment values('222222222', '2019-01-02 10:00:00',
'xxxxxxxx', '888888888');
409 insert into appointment values('111111111', '2019-01-01 10:00:00',
'xxxxxxxx', '999999999');
410 insert into appointment values('444444444', '2019-01-02 10:00:00',
'xxxxxxxx', '999999999');
411 insert into appointment values('333333333', '2019-01-01 14:00:00',
'xxxxxxxx', '101010101');
412 insert into appointment values('333333333', '2019-01-02 10:00:00',
'xxxxxxxx', '101010101');
413 insert into appointment values('111111111', '2019-01-01 13:00:00',
'xxxxxxxx', '110110110');
414 insert into appointment values('111111111', '2019-01-01 17:00:00',
'xxxxxxxx', '110110110');
415 insert into appointment values('444444444', '2019-01-01 13:00:00',
'xxxxxxxx', '121212121');
416 insert into appointment values('111111111', '2019-01-02 09:00:00',
'xxxxxxxx', '121212121');
417 insert into appointment values('444444444', '2019-01-02 11:00:00',
'xxxxxxxx', '131313131');
418 insert into appointment values('111111111', '2019-01-02 12:00:00',
'xxxxxxxx', '131313131');
419 insert into appointment values('222222222', '2019-01-01 13:00:00',
'xxxxxxxx', '141414141');
420 insert into appointment values('111111111', '2019-01-02 16:00:00',
'xxxxxxxx', '141414141');
421 insert into appointment values('222222222', '2019-01-01 15:00:00',
'xxxxxxxx', '151515151');
422 insert into appointment values('333333333', '2019-01-02 14:00:00',
'xxxxxxxx', '151515151');
423 insert into appointment values('111111111', '2019-01-03 10:00:00',
'xxxxxxxx', '161616161');
424 insert into appointment values('444444444', '2019-01-02 12:00:00',
'xxxxxxxx', '161616161');
425 insert into appointment values('111111111', '2019-01-03 13:00:00',
'xxxxxxxx', '171717171');
426 insert into appointment values('111111111', '2019-01-03 17:00:00',
'xxxxxxxx', '171717171');
427 insert into appointment values('333333333', '2019-01-03 10:00:00',
'xxxxxxxx', '181818181');
428 insert into appointment values('222222222', '2019-01-02 11:00:00',
'xxxxxxxx', '181818181');
429 insert into appointment values('444444444', '2019-01-02 17:00:00',
'xxxxxxxx', '191919191');
430 insert into appointment values('222222222', '2019-01-02 13:00:00',
'xxxxxxxx', '191919191');
431 insert into appointment values('111111111', '2018-02-01 10:00:00',
'xxxxxxxx', '202020202');
432 insert into appointment values('333333333', '2019-02-02 10:00:00',

```



```

'xxxxxxx', '202020202');
433 insert into appointment values('222222222', '2019-02-01 10:00:00',
'xxxxxxx', '202020202');
434 insert into appointment values('444444444', '2019-02-02 10:00:00',
'xxxxxxx', '212121212');
435 insert into appointment values('222222222', '2019-02-01 14:00:00',
'xxxxxxx', '212121212');
436 insert into appointment values('222222222', '2019-02-01 13:00:00',
'xxxxxxx', '212121212');
437 insert into appointment values('333333333', '2019-02-01 17:00:00',
'xxxxxxx', '232323232');
438 insert into appointment values('111111111', '2019-02-01 13:00:00',
'xxxxxxx', '232323232');
439 insert into appointment values('444444444', '2019-02-02 09:00:00',
'xxxxxxx', '242424242');
440 insert into appointment values('111111111', '2019-02-12 13:00:00',
'xxxxxxx', '242424242');
441 insert into appointment values('333333333', '2019-02-02 16:00:00',
'xxxxxxx', '252525252');
442 insert into appointment values('444444444', '2018-02-01 15:00:00',
'xxxxxxx', '252525252');
443 insert into appointment values('111111111', '2019-02-02 14:00:00',
'xxxxxxx', '262626262');
444 insert into appointment values('333333333', '2019-02-03 10:00:00',
'xxxxxxx', '262626262');
445 insert into appointment values('444444444', '2019-02-03 13:00:00',
'xxxxxxx', '272727272');
446 insert into appointment values('111111111', '2019-02-03 17:00:00',
'xxxxxxx', '272727272');
447 insert into appointment values('222222222', '2018-02-02 11:00:00',
'xxxxxxx', '282828282');
448 insert into appointment values('444444444', '2019-02-02 17:00:00',
'xxxxxxx', '292929292');
449 insert into appointment values('111111111', '2019-02-02 13:00:00',
'xxxxxxx', '292929292');
450
451
452 insert into consultation values('333333333', '2019-01-01 10:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
453 insert into consultation values('222222222', '2019-01-02 10:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
454 insert into consultation values('111111111', '2019-01-01 10:00:00',
'xxxxxxx','xxxx gingivitis xxx','xxxxxxx','xxxxxxx');
455 insert into consultation values('444444444', '2019-01-02 10:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
456 insert into consultation values('333333333', '2019-01-01 14:00:00',
'xxxxxxx','xxx gingivitis xxx','xxxxxxx','xxxxxxx');
457 insert into consultation values('111111111', '2019-01-01 13:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
458 insert into consultation values('444444444', '2019-01-01 13:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
459 insert into consultation values('111111111', '2019-01-02 09:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
460 insert into consultation values('444444444', '2019-01-02 11:00:00',
'xxxxxxx','xx gingivitis xxx','xxxxxxx','xxxxxxx');
461 insert into consultation values('111111111', '2019-01-02 12:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
462 insert into consultation values('111111111', '2019-01-02 16:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
463 insert into consultation values('222222222', '2019-01-01 15:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
464 insert into consultation values('333333333', '2019-01-02 14:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
465 insert into consultation values('111111111', '2019-01-03 10:00:00',
'xxxxxxx','xxx gingivitis xxx','xxxxxxx','xxxxxxx');
466 insert into consultation values('444444444', '2019-01-02 12:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
467 insert into consultation values('111111111', '2019-01-03 17:00:00',
'xxxxxxx','xxxx periodontitis xxx','xxxxxxx','xxxxxxx');
468 insert into consultation values('333333333', '2019-01-03 10:00:00',
'xxxxxxx','xxxx gingivitis xxx','xxxxxxx','xxxxxxx');
469 insert into consultation values('222222222', '2019-01-02 11:00:00',

```

```
'xxxxxxx','xxxx gingivitis xxx','xxxxxxx','xxxxxxx');
470 insert into consultation values('22222222', '2019-01-02 13:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
471 insert into consultation values('11111111', '2018-02-01 10:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
472 insert into consultation values('33333333', '2019-02-02 10:00:00',
'xxxxxxx','xx periodontitis xxxxx','xxxxxxx','xxxxxxx');
473 insert into consultation values('22222222', '2019-02-01 10:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
474 insert into consultation values('44444444', '2019-02-02 10:00:00',
'xxxxxxx','xxx gingivitis periodontitis xxx','xxxxxxx','xxxxxxx');
475 insert into consultation values('22222222', '2019-02-01 14:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
476 insert into consultation values('22222222', '2019-02-01 13:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
477 insert into consultation values('33333333', '2019-02-01 17:00:00',
'xxxxxxx','xxxxxx periodontitis x','xxxxxxx','xxxxxxx');
478 insert into consultation values('11111111', '2019-02-01 13:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
479 insert into consultation values('44444444', '2019-02-02 09:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
480 insert into consultation values('11111111', '2019-02-12 13:00:00',
'xxxxxxx','xxx gingivitis xxx','xxxxxxx','xxxxxxx');
481 insert into consultation values('44444444', '2018-02-01 15:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
482 insert into consultation values('44444444', '2019-02-03 13:00:00',
'xxxxxxx','xxxxxx gingivitis xx','xxxxxxx','xxxxxxx');
483 insert into consultation values('11111111', '2019-02-03 17:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
484 insert into consultation values('11111111', '2019-02-02 13:00:00',
'xxxxxxx','xxxxxxx','xxxxxxx','xxxxxxx');
485
486 insert into consultation_assistant values('33333333', '2019-01-01
10:00:00', '55555555');
487 insert into consultation_assistant values('33333333', '2019-01-01
10:00:00', '66666666');
488 insert into consultation_assistant values('22222222', '2019-01-02
10:00:00', '66666666');
489 insert into consultation_assistant values('11111111', '2019-01-01
10:00:00', '55555555');
490 insert into consultation_assistant values('44444444', '2019-01-02
10:00:00', '66666666');
491 insert into consultation_assistant values('33333333', '2019-01-01
14:00:00', '55555555');
492 insert into consultation_assistant values('11111111', '2019-01-01
13:00:00', '66666666');
493 insert into consultation_assistant values('44444444', '2019-01-01
13:00:00', '55555555');
494 insert into consultation_assistant values('11111111', '2019-01-02
09:00:00', '66666666');
495 insert into consultation_assistant values('44444444', '2019-01-02
11:00:00', '55555555');
496 insert into consultation_assistant values('11111111', '2019-01-02
12:00:00', '66666666');
497 insert into consultation_assistant values('11111111', '2019-01-02
16:00:00', '55555555');
498 insert into consultation_assistant values('22222222', '2019-01-01
15:00:00', '66666666');
499 insert into consultation_assistant values('33333333', '2019-01-02
14:00:00', '55555555');
500 insert into consultation_assistant values('11111111', '2019-01-03
10:00:00', '66666666');
501 insert into consultation_assistant values('44444444', '2019-01-02
12:00:00', '55555555');
502 insert into consultation_assistant values('11111111', '2019-01-03
17:00:00', '66666666');
503 insert into consultation_assistant values('33333333', '2019-01-03
10:00:00', '55555555');
504 insert into consultation_assistant values('22222222', '2019-01-02
11:00:00', '66666666');
505 insert into consultation_assistant values('22222222', '2019-01-02
13:00:00', '55555555');
```

```

506 insert into consultation_assistant values('111111111', '2018-02-01
10:00:00', '666666666');
507 insert into consultation_assistant values('333333333', '2019-02-02
10:00:00', '666666666');
508 insert into consultation_assistant values('222222222', '2019-02-01
10:00:00', '555555555');
509 insert into consultation_assistant values('444444444', '2019-02-02
10:00:00', '666666666');
510 insert into consultation_assistant values('222222222', '2019-02-01
14:00:00', '555555555');
511 insert into consultation_assistant values('222222222', '2019-02-01
13:00:00', '555555555');
512 insert into consultation_assistant values('333333333', '2019-02-01
17:00:00', '666666666');
513 insert into consultation_assistant values('333333333', '2019-02-01
17:00:00', '555555555');
514 insert into consultation_assistant values('111111111', '2019-02-01
13:00:00', '555555555');
515 insert into consultation_assistant values('444444444', '2019-02-02
09:00:00', '555555555');
516 insert into consultation_assistant values('111111111', '2019-02-12
13:00:00', '666666666');
517 insert into consultation_assistant values('444444444', '2018-02-01
15:00:00', '555555555');
518 insert into consultation_assistant values('444444444', '2019-02-03
13:00:00', '666666666');
519 insert into consultation_assistant values('444444444', '2019-02-03
13:00:00', '555555555');
520 insert into consultation_assistant values('111111111', '2019-02-03
17:00:00', '666666666');
521 insert into consultation_assistant values('111111111', '2019-02-02
13:00:00', '666666666');
522
523
524
525 insert into diagnostic_code values('118065D', 'Dental Cavities');
526 insert into diagnostic_code values('101050D', 'Bacterial Oral Infection');
527 insert into diagnostic_code values('111975D', 'Disorder of Hard Tissues of
Tooth');
528 insert into diagnostic_code values('140880D', 'Infectious Disease');
529 insert into diagnostic_code values('135048D', 'Sensitive Dentin');
530 insert into diagnostic_code values('138192D', 'Gingivitis');
531
532 insert into diagnostic_code_relation values('140880D', '101050D', 'Is a');
533 insert into diagnostic_code_relation values('118065D', '101050D', 'Is a');
534 insert into diagnostic_code_relation values('118065D', '111975D', 'Is a');
535 insert into diagnostic_code_relation values('135048D', '111975D', 'Is a');
536
537 insert into consultation_diagnostic values('333333333', '2019-01-01
10:00:00', '140880D');
538 insert into consultation_diagnostic values('222222222', '2019-01-02
10:00:00', '101050D');
539 insert into consultation_diagnostic values('444444444', '2019-01-02
10:00:00', '140880D');
540 insert into consultation_diagnostic values('111111111', '2019-01-01
13:00:00', '140880D');
541 insert into consultation_diagnostic values('444444444', '2019-01-01
13:00:00', '118065D');
542 insert into consultation_diagnostic values('111111111', '2019-01-02
09:00:00', '118065D');
543 insert into consultation_diagnostic values('111111111', '2019-01-02
12:00:00', '111975D');
544 insert into consultation_diagnostic values('333333333', '2019-01-02
14:00:00', '135048D');
545 insert into consultation_diagnostic values('111111111', '2019-01-03
17:00:00', '135048D');
546 insert into consultation_diagnostic values('333333333', '2019-01-03
10:00:00', '135048D');
547 insert into consultation_diagnostic values('222222222', '2019-01-02
13:00:00', '135048D');
548 insert into consultation_diagnostic values('222222222', '2019-02-01
14:00:00', '138192D');

```

```
549 insert into consultation_diagnostic values('3333333333', '2019-02-01
17:00:00', '138192D');
550 insert into consultation_diagnostic values('1111111111', '2019-02-12
13:00:00', '138192D');
551
552 insert into medication values('Tylenol','McNeil');
553 insert into medication values('Ibuprofen','Boots Group');
554 insert into medication values('Orajel','Norwich Warner Pharmaceuticals');
555
556 insert into prescription values('Ibuprofen','Boots Group', '4444444444',
'2019-01-02 10:00:00', '140880D', '100mg', 'Every night');
557 insert into prescription values('Ibuprofen','Boots Group', '1111111111',
'2019-01-01 13:00:00', '140880D', '100mg', 'At breakfast');
558 insert into prescription values('Ibuprofen','Boots Group', '4444444444',
'2019-01-01 13:00:00', '118065D', '50mg', 'Every night');
559 insert into prescription values('Tylenol','McNeil', '1111111111',
'2019-01-02 09:00:00', '118065D', '100mg', 'After dinner');
560 insert into prescription values('Tylenol','McNeil', '3333333333',
'2019-01-03 10:00:00', '135048D', '100mg', 'At breakfast');
561 insert into prescription values('Orajel','Norwich Warner Pharmaceuticals',
'3333333333', '2019-01-02 14:00:00', '135048D', '100mg', 'At breakfast');
562 insert into prescription values('Orajel','Norwich Warner Pharmaceuticals',
'1111111111', '2019-01-03 17:00:00', '135048D', '50mg', 'Every night');
563 insert into prescription values('Orajel','Norwich Warner Pharmaceuticals',
'2222222222', '2019-01-02 13:00:00', '135048D', '100mg', 'Every night');
564 insert into prescription values('Orajel','Norwich Warner Pharmaceuticals',
'1111111111', '2019-02-12 13:00:00', '138192D', '100mg', 'Every night');
565
566 insert into procedure_ values('Extraction', 'Surgery');
567 insert into procedure_ values('Braces', 'Repair');
568 insert into procedure_ values('Gum Surgery', 'Surgery');
569 insert into procedure_ values('Teeth Whitening', 'Repair');
570
571 insert into procedure_in_consultation values('Extraction', '3333333333',
'2019-01-01 10:00:00', 'xxxxxxx');
572 insert into procedure_in_consultation values('Teeth Whitening',
'3333333333', '2019-01-01 10:00:00', 'xxxxxxx');
573 insert into procedure_in_consultation values('Braces', '1111111111',
'2019-01-01 13:00:00', 'xxxxxxx');
574 insert into procedure_in_consultation values('Extraction', '4444444444',
'2019-01-01 13:00:00', 'xxxxxxx');
575 insert into procedure_in_consultation values('Teeth Whitening',
'4444444444', '2019-01-02 10:00:00', 'xxxxxxx');
576 insert into procedure_in_consultation values('Extraction', '1111111111',
'2019-01-02 12:00:00', 'xxxxxxx');
577 insert into procedure_in_consultation values('Teeth Whitening',
'2222222222', '2019-01-02 13:00:00', 'xxxxxxx');
578 insert into procedure_in_consultation values('Teeth Whitening',
'2222222222', '2019-02-01 14:00:00', 'xxxxxxx');
579 insert into procedure_in_consultation values('Teeth Whitening',
'3333333333', '2019-02-01 17:00:00', 'xxxxxxx');
580 insert into procedure_in_consultation values('Teeth Whitening',
'1111111111', '2019-02-12 13:00:00', 'xxxxxxx');
581
582 insert into procedure_radiology values('Braces', 'images1.png',
'1111111111', '2019-01-01 13:00:00');
583 insert into procedure_radiology values('Extraction', 'images2.png',
'3333333333', '2019-01-01 10:00:00');
584
585 insert into teeth values(1,1,'Molar');
586 insert into teeth values(1,2,'Pre-Molar');
587 insert into teeth values(2,1,'Incisor');
588 insert into teeth values(2,2,'Molar');
589
590 insert into procedure_charting values('Extraction', '3333333333',
'2019-01-01 10:00:00', 1,1, 'xxxxxxx', 10.0);
591 insert into procedure_charting values('Extraction', '3333333333',
'2019-01-01 10:00:00', 1,2, 'xxxxxxx', 5.0);
592 insert into procedure_charting values('Extraction', '3333333333',
'2019-01-01 10:00:00', 2,1, 'xxxxxxx', 10.0);
593 insert into procedure_charting values('Extraction', '3333333333',
'2019-01-01 10:00:00', 2,2, 'xxxxxxx', 5.0);
```

```
594 insert into procedure_charting values('Braces', '111111111', '2019-01-01
13:00:00', 1,1,'xxxxxxx', 10.0);
595 insert into procedure_charting values('Braces', '111111111', '2019-01-01
13:00:00', 1,2,'xxxxxxx', 10.0);
596 insert into procedure_charting values('Braces', '111111111', '2019-01-01
13:00:00', 2,1,'xxxxxxx', 10.0);
597 insert into procedure_charting values('Teeth Whitening', '222222222',
'2019-02-01 14:00:00', 1,1, 'xxxxxxx', 5.0);
598 insert into procedure_charting values('Teeth Whitening', '222222222',
'2019-02-01 14:00:00', 1,2, 'xxxxxxx', 5.0);
599 insert into procedure_charting values('Teeth Whitening', '222222222',
'2019-02-01 14:00:00', 2,1, 'xxxxxxx', 10.0);
600 insert into procedure_charting values('Teeth Whitening', '222222222',
'2019-02-01 14:00:00', 2,2, 'xxxxxxx', 10.0);
601 insert into procedure_charting values('Teeth Whitening', '333333333',
'2019-02-01 17:00:00', 1,1, 'xxxxxxx', 2.0);
602 insert into procedure_charting values('Teeth Whitening', '333333333',
'2019-02-01 17:00:00', 2,1, 'xxxxxxx', 1.0);
603 insert into procedure_charting values('Teeth Whitening', '111111111',
'2019-02-12 13:00:00', 1,1, 'xxxxxxx', 10.0);
604 insert into procedure_charting values('Teeth Whitening', '111111111',
'2019-02-12 13:00:00', 1,2, 'xxxxxxx', 4.0);
605 insert into procedure_charting values('Teeth Whitening', '111111111',
'2019-02-12 13:00:00', 2,1, 'xxxxxxx', 5.0);
606
```

```

1
2  /*Query 1 - List the VAT, name, and phone number(s) for all clients that had
3  consultations with the doctor named Jane Sweettooth. The list should
4  be presented according to the alphabetical order for the names.*/
5  select client.VAT, client.name, phone
6  from consultation, appointment, client natural left join
7  phone_number_client, doctor, employee
8  where appointment.VAT_doctor = consultation.VAT_doctor and
9  appointment.date_timestamp = consultation.date_timestamp and
10  appointment.VAT_client = client.VAT
11  and consultation.VAT_doctor = doctor.VAT and doctor.VAT = employee.VAT
12  and employee.name='Jane Sweettooth'
13  order by client.name asc;
14
15 /*Query 2 - List the name of all trainee doctors with reports associated to
16 an
17 evaluation score below the value of three, or with a description that
18 contains the term insufficient. The name should be presented together with
19 the VAT of the trainee, the name for the doctor that
20 made the evaluation, the evaluation score, and the textual description for
21 the evaluation report. Results should be sorted according to
22 the evaluation score, in descending order.*/
23 select t.name as trainee_name, t.VAT as trainee_VAT, p.name as
24 supervisor_name, evaluation, description
25 from employee t, trainee_doctor, supervision_report, employee p
26 where t.VAT = trainee_doctor.VAT and supervision_report.VAT =
27 trainee_doctor.VAT and trainee_doctor.supervisor = p.VAT and (evaluation <
28 3 or description like '%insufficient%')
29 order by evaluation desc;
30
31 /*Query 3 - List the name, city, and VAT for all clients where the most
32 recent
33 consultation has the objective part of the SOAP note mentioning the
34 terms gingivitis or periodontitis*/
35 select name, city, VAT
36 from(
37 /*select most recent consultation of each client*/
38 select * from consultation natural join appointment as c
39 group by VAT_client, date_timestamp
40 having( date_timestamp >=all(select date_timestamp from appointment natural
41 join consultation where c.VAT_client = appointment.VAT_client))
42 ) as most_recent_consultation, client
43 where most_recent_consultation.VAT_client = client.VAT and (SOAP_O like
44 '%gingivitis%' or SOAP_O like '%periodontitis%');
45
46 /*Query 4 - List the name, VAT and address (i.e., street, city and zip) of
47 all
48 clients of the clinic that have had appointments but that never had
49 a consultation (i.e., clients that never showed to an appointment).*/
50 select name, VAT, street, city, zip
51 from client c
52 where exists( /*client has at least one appointment*/
53 select * from appointment
54 where c.VAT = appointment.VAT_client
55 )
56 and not exists( /*client doesn't have any consultations*/
57 select * from consultation natural join appointment
58 where c.VAT = appointment.VAT_client
59 );
60
61 /*Query 5 - For each possible diagnosis, presenting the code together with
62 the
63 description, list the number of distinct medication names that have
64 been prescribed to treat that condition. Sort the results according
65 to the number of distinct medication names, in ascending order.*/
66 select diagnostic_code.ID, diagnostic_code.description, count(distinct
67 prescription.name)
68 from prescription, diagnostic_code
69 where prescription.ID = diagnostic_code.ID
70 group by diagnostic_code.ID

```

```

57 order by count(distinct prescription.name) asc;
58
59 /*Query 6 - Present the average number of nurses/assistants, procedures,
60 diagnostic codes, and prescriptions involved in consultations from the
61 year 2019, respectively for clients belonging to two age groups: less
62 or equal to 18 years old, and more than 18 years old*/
63 (select '>18' as age, avg(table_nurse.count) as avg_nurses,
64 avg(table_procedure.count) as avg_procedures, avg(table_diagnostic.count)
65 as avg_diagnostic_codes, avg(table_prescription.count) as avg_prescriptions
66 from( /*column containing the number of nurses for each consultation, for
67 clients with >18 years and in the year 2019*/
68 select count(consultation_assistant.VAT_nurse) as count
69 from consultation natural left outer join consultation_assistant,
70 appointment, client
71 where appointment.VAT_doctor = consultation.VAT_doctor and
72 appointment.date_timestamp = consultation.date_timestamp and
73 client.VAT = appointment.VAT_client and
74 year(appointment.date_timestamp) = '2019' and client.age > 18
75 group by consultation.VAT_doctor, consultation.date_timestamp
76 ) as table_nurse,
77 ( /*column containing the number of procedures for each consultation,
78 for clients with >18 years and in the year 2019*/
79 select count(procedure_in_consultation.name) as count
80 from consultation natural left outer join procedure_in_consultation,
81 appointment, client
82 where appointment.VAT_doctor = consultation.VAT_doctor and
83 appointment.date_timestamp = consultation.date_timestamp and
84 client.VAT = appointment.VAT_client and
85 year(appointment.date_timestamp) = '2019' and client.age > 18
86 group by consultation.VAT_doctor, consultation.date_timestamp
87 ) as table_procedure,
88 ( /*column containing the number of diagnostics for each consultation,
89 for clients with >18 years and in the year 2019*/
90 select count(consultation_diagnostic.ID) as count
91 from consultation natural left outer join consultation_diagnostic,
92 appointment, client
93 where appointment.VAT_doctor = consultation.VAT_doctor and
94 appointment.date_timestamp = consultation.date_timestamp and
95 client.VAT = appointment.VAT_client and
96 year(appointment.date_timestamp) = '2019' and client.age > 18
97 group by consultation.VAT_doctor, consultation.date_timestamp
98 ) as table_diagnostic,
99 ( /*column containing the prescription of diagnostics for each
100 consultation, for clients with >18 years and in the year 2019*/
101 select count(prescription.name) as count
102 from consultation natural left outer join prescription, appointment,
103 client
104 where appointment.VAT_doctor = consultation.VAT_doctor and
105 appointment.date_timestamp = consultation.date_timestamp and
106 client.VAT = appointment.VAT_client and
107 year(appointment.date_timestamp) = '2019' and client.age > 18
108 group by consultation.VAT_doctor, consultation.date_timestamp
109 ) as table_prescription
110 )
111 union
112 (select '<=18' as age, avg(table_nurse.count), avg(table_procedure.count),
113 avg(table_diagnostic.count), avg(table_prescription.count)
114 from( /*column containing the number of nurses for each consultation, for
115 clients with <=18 years and in the year 2019*/
116 select count(consultation_assistant.VAT_nurse) as count
117 from consultation natural left outer join consultation_assistant,
118 appointment, client
119 where appointment.VAT_doctor = consultation.VAT_doctor and
120 appointment.date_timestamp = consultation.date_timestamp and
121 client.VAT = appointment.VAT_client and
122 year(appointment.date_timestamp) = '2019' and client.age <= 18
123 group by consultation.VAT_doctor, consultation.date_timestamp
124 ) as table_nurse,
125 ( /*column containing the number of procedures for each consultation,
126 for clients with <=18 years and in the year 2019*/
127 select count(procedure_in_consultation.name) as count
128 from consultation natural left outer join procedure_in_consultation,

```



```

appointment, client
104  where appointment.VAT_doctor = consultation.VAT_doctor and
appointment.date_timestamp = consultation.date_timestamp and
105  client.VAT = appointment.VAT_client and
year(appointment.date_timestamp) = '2019' and client.age <= 18
106  group by consultation.VAT_doctor, consultation.date_timestamp
107  ) as table_procedure,
108  ( /*column containing the number of diagnostics for each consultation,
for clients with <=18 years and in the year 2019*/
109  select count(consultation_diagnostic.ID) as count
110  from consultation natural left outer join consultation_diagnostic,
appointment, client
111  where appointment.VAT_doctor = consultation.VAT_doctor and
appointment.date_timestamp = consultation.date_timestamp and
112  client.VAT = appointment.VAT_client and
year(appointment.date_timestamp) = '2019' and client.age <= 18
113  group by consultation.VAT_doctor, consultation.date_timestamp
114  ) as table_diagnostic,
115  ( /*column containing the prescription of diagnostics for each
consultation, for clients with <=18 years and in the year 2019*/
116  select count(prescription.name) as count
117  from consultation natural left outer join prescription, appointment,
client
118  where appointment.VAT_doctor = consultation.VAT_doctor and
appointment.date_timestamp = consultation.date_timestamp and
119  client.VAT = appointment.VAT_client and
year(appointment.date_timestamp) = '2019' and client.age <= 18
120  group by consultation.VAT_doctor, consultation.date_timestamp
121  ) as table_prescription
122  );
123
124
125  /*Query 7 - For each diagnostic code, present the name of the most common
medication used to treat that condition (i.e., the medication name that
126  more often appears associated to prescriptions for that diagnosis).*/
127  select ID, name
128  from prescription p
129  group by ID, name
130  having (count(*) >= all(select count(*) from prescription where ID = p.ID
group by name));
131
132  /*Query 8 - List, alphabetically, the names and labs for the medications
that, in
133  the year 2019, have been used to treat "dental cavities", but have
134  not been used to treat any "infectious disease". You can use the
135  aforementioned names for searching diagnostic codes in the dataset,
136  without considering relations (e.g., part-of relations) between diagnostic
codes.*/
137  select name, lab
138  from prescription p, diagnostic_code
139  where diagnostic_code.ID = p.ID and diagnostic_code.description = 'Dental
Cavities' and year(p.date_timestamp) = '2019'
140  and not exists(
141      select * from prescription, diagnostic_code where prescription.name =
p.name
142      and prescription.lab = p.lab and prescription.ID = diagnostic_code.ID
143      and diagnostic_code.description = 'Infectious Disease' and
year(prescription.date_timestamp) = '2019')
144  order by name asc;
145
146  /*Query 9 - List the names and addresses of clients that have never missed an
147  appointment in 2019 (i.e., the clients that, in the year 2019, have
148  always appeared in all the consultations scheduled for them).*/
149  select name, street, city, zip
150  from client
151  where VAT not in ( /*select all clients that have missed at least one
appointment in 2019*/
152      select VAT_client
153      from appointment natural left outer join consultation
154      where consultation.VAT_doctor is null and
year(appointment.date_timestamp) = '2019'
155  );

```





```
1  --INDEXES FOR QUERY 1
2  --We don't need to create indexes on columns that aren't primary keys but
3  --they are foreign keys, so there is already an index used to compare the
4  --values to the column, and creating another
5  --index would be only a waste of memory space
6  --
7  -- We use a BTree index instead of a Hash index, since we want to retrieve
8  -- many values
9  -- and sort them by order.
10 --BTree is more efficient when ones wants to retrieve a large set of
11 --ordered values, instead a single value, since it isn't required
12 --to retrieve those values from many buckets and then ordering them
13 -- The used index is NonClustered since the clustered index is already used
14 -- for the primary key
15 -- The index is created on column that needs to be sorted. This makes it
16 -- much easier for the query to retrieve the names sorted from the binary tree,
17 -- using binary search
18 create index index1_1 on employee (name);
19
20 --query that uses this index:
21 select client.VAT, client.name, phone
22 from consultation, appointment, client natural left join
23 phone_number_client, doctor, employee
24 where appointment.VAT_doctor = consultation.VAT_doctor and
25 appointment.date_timestamp = consultation.date_timestamp
26 and appointment.VAT_client = client.VAT
27 and consultation.VAT_doctor = doctor.VAT and doctor.VAT = employee.VAT
28 and employee.name='Jane Sweettooth'
29 order by client.name asc;
30
31 --INDEXES FOR QUERY 2
32 --We don't need to create indexes on columns that aren't primary keys but
33 --they are foreign keys, so there is already an index used to compare the
34 --values to the column, and creating another
35 --index would be only a waste of memory space
36 --
37 -- a BTree index is created on the column evaluation of supervision_report,
38 -- instead of a hash index,
39 -- since we want to retrieve many values using the less than or equal
40 -- operator
41 --Since the Btree index orders the values, it is more suited for this case
42 --the index is NonClustered because the primary_key of supervision_report
43 --already uses the clustered_index
44 create index index2_1 on supervision_report(evaluation);
45
46 --A fulltext index is created on the description column of
47 supervision_report since we want to find the
48 --frequency of occurrence of a word in the text
49 create fulltext index index2_2 on supervision_report(description);
50
51 --query that uses this index:
52 --We need to change the query to use the FullText Index, since the like
53 --operator
54 --won't use the full text index, but will instead use a buffer
55 --Therefore, we need to use the MATCH(column_name) AGAINST('query') operator
56 select t.name as trainee_name, t.VAT as trainee_VAT, p.name as
57 supervisor_name, evaluation, description
58 from employee t, trainee_doctor, supervision_report, employee p
59 where t.VAT = trainee_doctor.VAT and supervision_report.VAT =
60 trainee_doctor.VAT and trainee_doctor.supervisor = p.VAT
61 and (evaluation < 3 or match(supervision_report.description)
62 against('insufficient'))
63 order by evaluation desc;
```

```

1  /*1. Change the address of the doctor named Jane Sweettooth, to a different
   city and street of your choice.*/
2  update employee inner join doctor on doctor.VAT = employee.VAT
3  set street = 'Alameda D. Afonso Henriques', city = 'Lisboa' where name =
   'Jane Sweettooth';
4
5  /*2. Change the salary of all doctors that had more than 100 appointments
   in 2019. The new salaries should correspond to an increase in
6  5% from the old values.*/
7  update employee set salary =1.05*salary
8  where employee.VAT in( /*select all doctors that had more than 100
   appointments in 2019*/
9      select d.VAT
10     from doctor as d inner join appointment as a on d.VAT=a.VAT_doctor
11     where year(a.date_timestamp) = '2019'
12     group by a.VAT_doctor
13     having(count(distinct date_timestamp)>100)
14     );
15
16
17  /*3. Delete the doctor named Jane Sweettooth from the database, removing
   also all the appointments and all the consultations (including
18  the associated procedures, diagnosis and prescriptions) in which she
19  was involved. Notice that if there are procedures/diagnosis that were
20  only performed/assigned by this doctor, you should remove them also
21  from the database.*/
22
23  /*consultation and appointment, as well as procedure_in_consultation,
   consultation_diagnostic and prescription
24  with this doctor are automatically deleted because of the "on delete
   cascade" command used in "create table" in all these tables*/
25  /*So, we only have to manually delete procedure_ and diagnostic code that
   are only used by the doctor that is going to be deleted*/
26
27  /*delete all procedures performed by Jane Sweettooth such that the
   procedure was NOT performed by anyone who was NOT Jane Sweettooth*/
28  delete p
29  from procedure_p
30      join procedure_in_consultation as pic
31      on p.name = pic.name
32      join doctor as d
33      on pic.VAT_doctor = d.VAT
34      join employee
35      on d.VAT = employee.VAT
36  where employee.name = 'Jane Sweettooth' and
37  not exists(select * from (select * from procedure_) as p_ natural join
   procedure_in_consultation
38  where p_.name = p.name and procedure_in_consultation.VAT_doctor <> d.VAT);
39
40  /*delete all diagnostic codes assigned to Jane Sweettooth such that the
   diagnostic code was NOT assigned to anyone who was NOT Jane Sweettooth*/
41  delete dc
42  from diagnostic_code dc
43      join consultation_diagnostic as cd
44      on dc.ID = cd.ID
45      join doctor as d
46      on cd.VAT_doctor = d.VAT
47      join employee
48      on d.VAT = employee.VAT
49  where employee.name = 'Jane Sweettooth' and
50  not exists(select * from (select * from diagnostic_code) as cd_ natural
   join consultation_diagnostic
51  where cd.ID = cd_.ID and consultation_diagnostic.VAT_doctor <> d.VAT);
52
53  /*Finally, delete Jane Sweettooth from the database*/
54  DELETE employee
55  from employee
56  where name = 'Jane Sweettooth';
57
58
59  /*4. Find the diagnosis code corresponding to gingivitis. Create also a new
   diagnosis code corresponding to periodontitis. Change the diagnosis
60

```

```
61  from gingivitis to periodontitis for all clients where, for the same
62  consultation/diagnosis, a dental charting procedure shows a value
63  above 4 in terms of the average gap between the teeth and the gums.*/
64
65  /*diagnostic code corresponding to gingivitis*/
66  select ID from diagnostic_code where description = 'Gengivitis';
67
68  /*Creation of a new diagnosis code corresponding to periodontitis*/
69  insert into diagnostic_code values('231050D', 'Peridontitis');
70
71
72  update consultation_diagnostic cd set ID = (select ID from diagnostic_code
73  where description = 'Peridontitis')
74  where cd.ID = (select ID from diagnostic_code where description =
75  'Gengivitis')
76  and exists(
77    select pc.VAT from
78    procedure_charting as pc
79    where cd.VAT_doctor = pc.VAT and cd.date_timestamp = pc.date_timestamp
80    group by pc.VAT, pc.date_timestamp
81    having avg(measure) > 4);
82
83
84
85
```

```
1 drop view if exists facts_consults;
2 drop view if exists dim_location_client;
3 drop view if exists dim_client;
4 drop view if exists dim_date;
5
6 --View 1
7 create view dim_date as
8 (select date_timestamp, day(date_timestamp) as day, month(date_timestamp)
9 as month, year(date_timestamp) as year
10 from consultation);
11 select * from dim_date;
12
13 --View 2
14 create view dim_client as
15 (select VAT, gender, floor(datediff(now(), birth_date)/365) as age
16 from client);
17 select * from dim_client;
18
19 --View 3
20 create view dim_location_client as
21 (select zip, city
22 from client);
23 select * from dim_location_client;
24
25 --View 4
26 create view facts_consults as
27 (
28 select client.VAT, c.date_timestamp, client.zip, count(pic.name) as
29 num_procedures, count(p.name) as num_medications, count(cd.ID) as
30 num_diagnostic_codes
31 from consultation as c left join procedure_in_consultation as pic on
32 c.date_timestamp = pic.date_timestamp and c.VAT_doctor = pic.VAT_doctor
33 left join prescription as p on c.date_timestamp = p.date_timestamp and
34 c.VAT_doctor = p.VAT_doctor
35 left join consultation_diagnostic as cd on c.date_timestamp =
36 cd.date_timestamp and c.VAT_doctor = cd.VAT_doctor
37 join appointment as a on c.date_timestamp = a.date_timestamp and
38 c.VAT_doctor = a.VAT_doctor
39 join client on client.VAT = a.VAT_client
40 group by c.VAT_doctor, c.date_timestamp
41 );
42 select * from facts_consults;
```