



Information Systems and Databases  
**Project Assignment (part II)**

Maria Casimiro, ist178755  
Mariana Martins, ist180856  
Pedro Carmo, ist190989  
Group 46

November 14, 2017

# 1. SQL Database

```
1 DROP TABLE IF EXISTS Region;
2 DROP TABLE IF EXISTS Element;
3 DROP TABLE IF EXISTS Series;
4 DROP TABLE IF EXISTS Study;
5 DROP TABLE IF EXISTS Request;
6 DROP TABLE IF EXISTS Wears;
7 DROP TABLE IF EXISTS Period;
8 DROP TABLE IF EXISTS Reading;
9 DROP TABLE IF EXISTS Sensor;
10 DROP TABLE IF EXISTS Device;
11 DROP TABLE IF EXISTS Doctor;
12 DROP TABLE IF EXISTS Patient;
13
14
15 CREATE TABLE Patient(
16     p_number integer,
17     name varchar(255),
18     birthday date,
19     address varchar(255),
20     PRIMARY key(p_number));
21
22
23 CREATE TABLE Doctor(
24     p_number integer,
25     doctor_id integer,
26     PRIMARY key(doctor_id),
27     FOREIGN key(p_number) REFERENCES Patient(p_number));
28
29
30 CREATE TABLE Device(
31     serialnum varchar(255),
32     manufacturer varchar(255),
33     model varchar(255),
34     PRIMARY key(serialnum, manufacturer));
35
36
37 CREATE TABLE Sensor(
38     snum varchar(255),
39     manuf varchar(255),
40     units varchar(255),
41     PRIMARY key(snum, manuf),
42     FOREIGN key(snum, manuf) REFERENCES Device(serialnum, manufacturer));
43
44
45 CREATE TABLE Reading(
46     snum varchar(255),
47     manuf varchar(255),
48     date_time datetime,
49     value numeric(20,2),
50     PRIMARY key(snum, manuf, date_time),
51     FOREIGN key(snum, manuf) REFERENCES Sensor(snum, manuf));
```

```

52
53
54 CREATE TABLE Period(
55     p_start date,
56     p_end date,
57     PRIMARY key(p_start, p_end));
58
59
60 CREATE TABLE Wears(
61     p_start date,
62     p_end date,
63     patient integer,
64     snum varchar(255),
65     manuf varchar(255),
66     PRIMARY key(p_start, p_end, patient),
67     FOREIGN key(p_start, p_end) REFERENCES Period(p_start, p_end),
68     FOREIGN key(patient) REFERENCES Patient(p_number),
69     FOREIGN key(snum, manuf) REFERENCES Device(serialnum, manufacturer));
70
71
72 CREATE TABLE Request(
73     r_number integer,
74     patient_id integer,
75     doctor_id integer,
76     r_date date,
77     PRIMARY key(r_number),
78     FOREIGN key(patient_id) REFERENCES Patient(p_number),
79     FOREIGN key(doctor_id) REFERENCES Doctor(doctor_id));
80
81
82 CREATE TABLE Study(
83     request_number integer,
84     description varchar(255),
85     s_date date,
86     doctor_id integer,
87     serial_number varchar(255),
88     manufacturer varchar(255),
89
90     PRIMARY key(request_number, description),
91     FOREIGN key(request_number) REFERENCES Request(r_number),
92     FOREIGN key(doctor_id) REFERENCES Doctor(doctor_id),
93     FOREIGN key(serial_number, manufacturer) REFERENCES Device(serialnum,
94     manufacturer));
95
96
97 CREATE TABLE Series(
98     series_id integer,
99     name varchar(255),
100     base_url varchar(255),
101     request_number integer, description varchar(255),
102     PRIMARY key(series_id),
103     FOREIGN key(request_number, description) REFERENCES Study(
104     request_number, description));

```

```

101
102
103 CREATE TABLE Element(
104     series_id integer,
105     elem_index integer,
106     PRIMARY key(series_id, elem_index),
107     FOREIGN key(series_id) REFERENCES Series(series_id));
108
109
110 CREATE TABLE Region(
111     series_id integer,
112     elem_index integer,
113     x1 numeric(20,2),
114     y1 numeric(20,2),
115     x2 numeric(20,2),
116     y2 numeric(20,2),
117     PRIMARY key(series_id, elem_index, x1, y1, x2, y2),
118     FOREIGN key(series_id, elem_index) REFERENCES Element(series_id,
        elem_index));

```

## 2. Table Population

```
1  --PATIENT-----
2  INSERT INTO Patient
3      VALUES(001, 'Adams', '1978-03-30', 'Sunset Av. ');      --Doctor
4  INSERT INTO Patient
5      VALUES(023, 'Brooks', '1997-04-25', 'Sesame Street');
6  INSERT INTO Patient
7      VALUES(147, 'Curry', '1996-06-16', 'Areeiro Street');
8  INSERT INTO Patient
9      VALUES(055, 'Glenn', '1995-10-07', 'Madison Av. ');
10 INSERT INTO Patient
11     VALUES(012, 'Green', '1943-02-13', 'Baker Street');      --Doctor
12 INSERT INTO Patient
13     VALUES(108, 'Jackson', '1969-12-02', 'Fifth Av. ');      --Doctor
14 INSERT INTO Patient
15     VALUES(097, 'Hayes', '1968-11-21', 'First Street');
16 INSERT INTO Patient
17     VALUES(015, 'Johnson', '2016-03-24', 'Second Street');
18 INSERT INTO Patient
19     VALUES(130, 'Jones', '2005-09-12', 'Madison Av. ');
20 INSERT INTO Patient
21     VALUES(086, 'Lindsay', '1981-05-17', 'Second Street');    --Doctor
22 INSERT INTO Patient
23     VALUES(159, 'Smith', '1956-01-05', 'Fifth Av. ');
24 INSERT INTO Patient
25     VALUES(123, 'Turner', '1974-07-26', 'Madison Av. ');      --Doctor
26
27  --DOCTOR-----
28  INSERT INTO Doctor VALUES(001, 201);
29  INSERT INTO Doctor VALUES(108, 202);
30  INSERT INTO Doctor VALUES(012, 203);
31  INSERT INTO Doctor VALUES(086, 204);
32  INSERT INTO Doctor VALUES(123, 205);
33
34  --DEVICE-----
35  INSERT INTO Device VALUES('A230E5', 'Proteus', 'M3');      --cholesterol
36  INSERT INTO Device VALUES('A10U7F', 'Proteus', 'M3');      --cholesterol
37  INSERT INTO Device VALUES('A47B8M', 'Proteus', 'M3');      --cholesterol
38  INSERT INTO Device VALUES('B5TR46', 'Samsung', 'M6');      --thermometer
39  INSERT INTO Device VALUES('CF2YH9', 'HP', 'HP7');          --voltmeter
40  INSERT INTO Device VALUES('S57BT2', 'Medtronic', 'M4');    --X-ray machine
41  INSERT INTO Device VALUES('S76BT3', 'Medtronic', 'M4');    --X-ray machine
42  INSERT INTO Device VALUES('S89RE4', 'Medtronic', 'M5');    --ECG machine
43  INSERT INTO Device VALUES('S35G7U', 'Medtronic', 'M5');    --ECG machine
44  INSERT INTO Device VALUES('S640I1', 'Medtronic', 'M2');    --echo machine
45
46  --SENSOR-----
47  INSERT INTO Sensor VALUES('A230E5', 'Proteus', 'mg/dL');
48  INSERT INTO Sensor VALUES('A10U7F', 'Proteus', 'mg/dL');
49  INSERT INTO Sensor VALUES('A47B8M', 'Proteus', 'mg/dL');
50  INSERT INTO Sensor VALUES('B5TR46', 'Samsung', 'Â°C');
51  INSERT INTO Sensor VALUES('CF2YH9', 'HP', 'mV');
```

```

52
53 --READING-----
54 INSERT INTO Reading
55     VALUES('A230E5', 'Proteus', '2017-06-15 14:35:06', 235); --Smith
56 INSERT INTO Reading
57     VALUES('A230E5', 'Proteus', '2017-08-23 14:35:06', 235); --Smith
58 INSERT INTO Reading
59     VALUES('A230E5', 'Proteus', '2017-08-30 14:34:30', 220); --Smith
60 INSERT INTO Reading
61     VALUES('A230E5', 'Proteus', '2017-09-07 14:36:00', 205); --Smith
62 INSERT INTO Reading
63     VALUES('A230E5', 'Proteus', '2017-09-14 14:27:53', 180); --Smith
64 INSERT INTO Reading
65     VALUES('A230E5', 'Proteus', '2017-09-21 14:30:26', 190); --Green
66 INSERT INTO Reading
67     VALUES('A230E5', 'Proteus', '2017-09-30 14:32:15', 200); --Green
68 INSERT INTO Reading
69     VALUES('A230E5', 'Proteus', '2017-10-01 14:33:52', 190); --Green
70 INSERT INTO Reading
71     VALUES('A230E5', 'Proteus', '2017-09-17 14:35:26', 214); --Green
72
73 INSERT INTO Reading
74     VALUES('A10U7F', 'Proteus', '2017-09-02 15:24:14', 185); --Hayes
75 INSERT INTO Reading
76     VALUES('A10U7F', 'Proteus', '2017-09-15 13:15:36', 190); --Hayes
77 INSERT INTO Reading
78     VALUES('A10U7F', 'Proteus', '2017-09-21 16:35:54', 200); --Hayes
79 INSERT INTO Reading
80     VALUES('A10U7F', 'Proteus', '2017-09-30 15:24:14', 185); --Hayes
81 INSERT INTO Reading
82     VALUES('A10U7F', 'Proteus', '2017-10-02 12:30:25', 210); --Jackson
83 INSERT INTO Reading
84     VALUES('A10U7F', 'Proteus', '2017-10-14 14:53:05', 195); --Jackson
85
86 INSERT INTO Reading
87     VALUES('A47B8M', 'Proteus', '2017-09-13 09:25:34', 195); --Glenn
88 INSERT INTO Reading
89     VALUES('A47B8M', 'Proteus', '2017-09-20 10:30:45', 200); --Glenn
90 INSERT INTO Reading
91     VALUES('A47B8M', 'Proteus', '2017-09-27 11:14:51', 205); --Glenn
92 INSERT INTO Reading
93     VALUES('A47B8M', 'Proteus', '2017-10-05 10:45:02', 205); --Glenn
94 INSERT INTO Reading
95     VALUES('A47B8M', 'Proteus', '2017-10-15 11:20:10', 210); --Glenn
96 INSERT INTO Reading
97     VALUES('A47B8M', 'Proteus', '2017-10-20 10:53:40', 205); --Glenn
98
99 INSERT INTO Reading
100     VALUES('B5TR46', 'Samsung', '2017-10-07 16:37:24', 38); --Johnson
101 INSERT INTO Reading
102     VALUES('B5TR46', 'Samsung', '2017-10-08 16:40:00', 38.5); --Johnson
103 INSERT INTO Reading
104     VALUES('B5TR46', 'Samsung', '2017-10-09 16:35:16', 38.5); --Johnson

```

```

105 INSERT INTO Reading
106     VALUES('B5TR46', 'Samsung', '2017-10-10 16:30:10', 38);    --Johnson
107 INSERT INTO Reading
108     VALUES('B5TR46', 'Samsung', '2017-10-11 16:37:45', 37);    --Johnson
109 INSERT INTO Reading
110     VALUES('B5TR46', 'Samsung', '2017-10-12 16:35:02', 36.5); --Johnson
111
112 INSERT INTO Reading
113     VALUES('CF2YH9', 'HP', '2017-10-28 17:23:05', 35); --Curry
114 INSERT INTO Reading
115     VALUES('CF2YH9', 'HP', '2017-10-30 17:20:23', 30); --Curry
116 INSERT INTO Reading
117     VALUES('CF2YH9', 'HP', '2017-11-01 17:25:42', 32); --Curry
118 INSERT INTO Reading
119     VALUES('CF2YH9', 'HP', '2017-11-03 17:24:36', 37); --Curry
120
121 --PERIOD-----
122 INSERT INTO Period VALUES('2017-06-10', '2017-09-15');
123 INSERT INTO Period VALUES('2017-09-16', '2017-10-02');
124 INSERT INTO Period VALUES('2017-09-02', '2017-09-30');
125 INSERT INTO Period VALUES('2017-10-01', '2017-10-15');
126 INSERT INTO Period VALUES('2017-09-12', '2017-10-21');
127 INSERT INTO Period VALUES('2017-10-07', '2017-10-12');
128 INSERT INTO Period VALUES('2017-10-27', '2017-11-04');
129
130 --WEARS-----
131 INSERT INTO Wears
132     VALUES('2017-06-10', '2017-09-15', 159, 'A230E5', 'Proteus'); --Smith
133 INSERT INTO Wears
134     VALUES('2017-09-16', '2017-10-02', 012, 'A230E5', 'Proteus'); --Green
135 INSERT INTO Wears
136     VALUES('2017-09-02', '2017-09-30', 097, 'A10U7F', 'Proteus'); --Hayes
137 INSERT INTO Wears
138     VALUES('2017-10-01', '2017-10-15', 108, 'A10U7F', 'Proteus'); --
139     Jackson
140 INSERT INTO Wears
141     VALUES('2017-09-12', '2017-10-21', 055, 'A47B8M', 'Proteus'); --Glenn
142 INSERT INTO Wears
143     VALUES('2017-10-07', '2017-10-12', 015, 'B5TR46', 'Samsung'); --
144     Johnson
145 INSERT INTO Wears
146     VALUES('2017-10-27', '2017-11-04', 147, 'CF2YH9', 'HP');    --Curry
147
148 --REQUEST-----
149 INSERT INTO Request VALUES(01, 159, 201, '2016-08-18'); --Smith
150 INSERT INTO Request VALUES(02, 012, 202, '2016-09-10'); --Green
151 INSERT INTO Request VALUES(03, 097, 203, '2016-08-28'); --Hayes
152 INSERT INTO Request VALUES(04, 108, 204, '2016-09-25'); --Jackson
153 INSERT INTO Request VALUES(05, 023, 201, '2016-09-09'); --Brooks
154 INSERT INTO Request VALUES(06, 015, 205, '2016-10-07'); --Johnson
155 INSERT INTO Request VALUES(07, 159, 203, '2016-10-27'); --Smith
156 INSERT INTO Request VALUES(08, 159, 204, '2016-09-15'); --Smith
157 INSERT INTO Request VALUES(09, 159, 201, '2016-10-16'); --Smith

```

```

156 INSERT INTO Request VALUES(10, 159, 202, '2016-11-04'); --Smith
157 INSERT INTO Request VALUES(11, 108, 203, '2017-03-24'); --Jackson
158 INSERT INTO Request VALUES(12, 108, 203, '2017-05-12'); --Jackson
159 INSERT INTO Request VALUES(13, 108, 203, '2017-06-04'); --Jackson
160 INSERT INTO Request VALUES(14, 108, 203, '2017-09-15'); --Jackson
161 INSERT INTO Request VALUES(15, 108, 203, '2017-09-20'); --Jackson
162
163 --STUDY-----
164 INSERT INTO Study
165     VALUES(01, 'X-ray left knee',
166     '2016-08-23', 202, 'S57BT2', 'Medtronic'); --Smith
167 INSERT INTO Study
168     VALUES(02, 'X-ray both hands',
169     '2016-09-16', 203, 'S76BT3', 'Medtronic'); --Green
170 INSERT INTO Study
171     VALUES(03, 'ECG',
172     '2016-09-02', 204, 'S89RE4', 'Medtronic'); --Hayes
173 INSERT INTO Study
174     VALUES(04, 'ECG',
175     '2016-10-01', 205, 'S35G7U', 'Medtronic'); --Jackson
176 INSERT INTO Study
177     VALUES(05, 'pregnancy echography',
178     '2016-09-12', 202, 'S640I1', 'Medtronic'); --Brooks
179 INSERT INTO Study
180     VALUES(06, 'X-ray right arm',
181     '2016-10-07', 201, 'S57BT2', 'Medtronic'); --Johnson
182 INSERT INTO Study
183     VALUES(07, 'X-ray left knee',
184     '2016-10-27', 204, 'S76BT3', 'Medtronic'); --Smith
185 INSERT INTO Study
186     VALUES(08, 'ECG',
187     '2016-09-15', 205, 'S89RE4', 'Medtronic'); --Smith
188 INSERT INTO Study
189     VALUES(09, 'ECG',
190     '2016-10-16', 202, 'S35G7U', 'Medtronic'); --Smith
191 INSERT INTO Study
192     VALUES(10, 'echography left knee',
193     '2016-11-07', 203, 'S640I1', 'Medtronic'); --Smith
194 INSERT INTO Study
195     VALUES(11, 'X-ray right hand',
196     '2017-04-02', 202, 'S57BT2', 'Medtronic'); --Jackson
197 INSERT INTO Study
198     VALUES(12, 'X-ray both knees',
199     '2017-05-20', 204, 'S76BT3', 'Medtronic'); --Jackson
200 INSERT INTO Study
201     VALUES(13, 'echography right knee',
202     '2017-06-12', 201, 'S640I1', 'Medtronic'); --Jackson
203 INSERT INTO Study
204     VALUES(14, 'ECG',
205     '2017-09-20', 205, 'S89RE4', 'Medtronic'); --Jackson
206 INSERT INTO Study
207     VALUES(15, 'ECG',
208     '2017-10-01', 205, 'S35G7U', 'Medtronic'); --Jackson

```



```

209
210 --SERIES-----
211 INSERT INTO Series
212     VALUES(01, 'S0101', 'series01.01', 01, 'X-ray left knee');
213 INSERT INTO Series
214     VALUES(02, 'S0202', 'series02.02', 02, 'X-ray both hands');
215 INSERT INTO Series
216     VALUES(03, 'S0303', 'series03.03', 03, 'ECG');
217 INSERT INTO Series
218     VALUES(04, 'S0404', 'series04.04', 04, 'ECG');
219 INSERT INTO Series
220     VALUES(05, 'S0505', 'series05.05', 05, 'pregnancy echography');
221 INSERT INTO Series
222     VALUES(06, 'S0606', 'series06.06', 06, 'X-ray right arm');
223 INSERT INTO Series
224     VALUES(07, 'S0707', 'series07.07', 07, 'X-ray left knee');
225 INSERT INTO Series
226     VALUES(08, 'S0808', 'series08.08', 08, 'ECG');
227 INSERT INTO Series
228     VALUES(09, 'S0909', 'series09.09', 09, 'ECG');
229 INSERT INTO Series
230     VALUES(10, 'S1010', 'series10.10', 10, 'echography left knee');
231 INSERT INTO Series
232     VALUES(11, 'S57BT2', 'series11.11', 11, 'X-ray right hand');
233 INSERT INTO Series
234     VALUES(12, 'S76BT3', 'series12.12', 12, 'X-ray both knees');
235 INSERT INTO Series
236     VALUES(13, 'S640I1', 'series13.13', 13, 'echography right knee');
237 INSERT INTO Series
238     VALUES(14, 'S89RE4', 'series14.14', 14, 'ECG');
239
240 --ELEMENT-----
241 INSERT INTO Element VALUES(01, 01);
242 INSERT INTO Element VALUES(02, 01);
243 INSERT INTO Element VALUES(02, 02);
244 INSERT INTO Element VALUES(03, 01);
245 INSERT INTO Element VALUES(03, 02);
246 INSERT INTO Element VALUES(03, 03);
247 INSERT INTO Element VALUES(03, 04);
248 INSERT INTO Element VALUES(04, 01);
249 INSERT INTO Element VALUES(04, 02);
250 INSERT INTO Element VALUES(04, 03);
251 INSERT INTO Element VALUES(05, 01);
252 INSERT INTO Element VALUES(05, 02);
253 INSERT INTO Element VALUES(06, 01);
254 INSERT INTO Element VALUES(07, 01);
255 INSERT INTO Element VALUES(08, 01);
256 INSERT INTO Element VALUES(08, 02);
257 INSERT INTO Element VALUES(09, 01);
258 INSERT INTO Element VALUES(09, 02);
259 INSERT INTO Element VALUES(09, 03);
260 INSERT INTO Element VALUES(10, 01);
261 INSERT INTO Element VALUES(11, 01);

```

```

262 INSERT INTO Element VALUES (12, 01);
263 INSERT INTO Element VALUES (12, 02);
264 INSERT INTO Element VALUES (13, 01);
265 INSERT INTO Element VALUES (14, 01);
266
267 --REGION-----
268 INSERT INTO Region VALUES (01, 01, 0.7, 0.3, 0.75, 0.35);
269 INSERT INTO Region VALUES (05, 01, 0.5, 0.4, 0.6, 0.6);
270 INSERT INTO Region VALUES (05, 02, 0.6, 0.3, 0.8, 0.45);
271 INSERT INTO Region VALUES (06, 01, 0.3, 0.4, 0.35, 0.5);
272 INSERT INTO Region VALUES (07, 01, 0.7, 0.3, 0.75, 0.35);
273 INSERT INTO Region VALUES (10, 01, 0.2, 0.5, 0.35, 0.6);
274 INSERT INTO Region VALUES (12, 01, 0.3, 0.5, 0.35, 0.55);
275 INSERT INTO Region VALUES (13, 01, 0.5, 0.8, 0.6, 0.85);

```

### 3. Query

Name(s) of the patient(s) with the highest number of readings of units of “LDL cholesterol in mg/dL” above 200 in the past 90 days.

```
1  SELECT Patient.name
2  FROM Patient,
3       Wears,
4       Reading,
5       Sensor
6  WHERE Patient.p_number = Wears.patient
7        AND Wears.snum = Reading.snum
8        AND Wears.manuf = Reading.manuf
9        AND Sensor.snum = Reading.snum
10       AND Sensor.manuf = Reading.manuf
11       AND Sensor.units = "mg/dL"
12       AND Reading.value > 200
13       AND Wears.p_start <= DATE(Reading.date_time)
14       AND Wears.p_end >= DATE(Reading.date_time)
15       AND TIMESTAMPDIFF(DAY, CURRENT_TIMESTAMP, Reading.date_time) <= 90
16  GROUP BY Patient.name
17  HAVING COUNT(Reading.value) >= ALL
18       (SELECT COUNT(Reading.value)
19        FROM Patient, Wears, Reading, Sensor
20        WHERE Patient.p_number = Wears.patient
21              AND Wears.snum = Reading.snum
22              AND Wears.manuf = Reading.manuf
23              AND Sensor.snum = Reading.snum
24              AND Sensor.manuf = Reading.manuf
25              AND Sensor.units = "mg/dL"
26              AND Reading.value > 200
27              AND Wears.p_start <= DATE(Reading.date_time)
28              AND Wears.p_end >= DATE(Reading.date_time)
29              AND TIMESTAMPDIFF(DAY, CURRENT_TIMESTAMP, Reading.date_time) <= 90
30       GROUP BY Patient.name);
```

In the database, 4 of the 7 existant patients have readings of “LDL cholesterol in mg/dL” above 200 in the past 90 days - Jackson, Smith, Green and Glenn.

During this time, both Smith and Green have used the device Proteus A230E5 in different periods. For this reason, the lines 13-14 and 27-28 are necessary to guarantee that the readings belong to a certain patient:

```
1  AND Wears.p_start <= DATE(Reading.date_time)
2  AND Wears.p_end >= DATE(Reading.date_time)
```

Jackson and Green have 1 reading over 200 and Glenn and Smith have 4, therefore the result is Glenn and Smith.

## 4. Query

Name(s) of the patient(s) who have been subject of studies with all devices of manufacturer "Medtronic" in the past calendar year.

```
1  SELECT DISTINCT name
2  FROM Patient AS p
3  WHERE NOT EXISTS
4      (SELECT serialnum
5       FROM Device
6       WHERE manufacturer = "Medtronic"
7       AND serialnum NOT IN
8          (SELECT Study.serial_number
9           FROM Patient AS p2, Study, Request
10          WHERE Request.r_number = Study.request_number
11            AND p2.p_number = Request.patient_id
12            AND Request.patient_id = p.p_number
13            AND p2.p_number = p.p_number
14            AND YEAR(Study.s_date) = YEAR(CURRENT_DATE) - 1));
```

The innermost SELECT gets a list all the devices used by a certain patient in the last year (in this case 2016). The middle SELECT gets all the devices from "Medtronic" that are not in the previous list. The outermost SELECT gets patients depending on whether the NOT EXISTS output is TRUE OR FALSE, in other words, if there are some "Medtronic" devices not used by a certain patient or if there are none.

In the database, Jackson has used all "Medtronic" devices in 2017, this ensures that the query is correct if he does not appear as a result. As well as, that many patients have used some but not all "Medtronic" devices in 2016, reassuring that all must be used. Running the query, the only result is Smith.

## 5.i Trigger

Ensure that a doctor who prescribes an exam may not perform that same exam.

```
1 DELIMITER $$
2
3 CREATE TRIGGER check_doctor_insert BEFORE INSERT ON Study
4 FOR EACH ROW
5 BEGIN
6     IF EXISTS (SELECT Request.doctor_id
7                FROM Request
8                WHERE Request.r_number = new.request_number
9                      AND new.doctor_id = Request.doctor_id)
10    THEN
11        CALL invalid_doctor_assignment();
12    END IF;
13 END$$
14
15 CREATE TRIGGER check_doctor_update BEFORE UPDATE ON Study
16 FOR EACH ROW
17 BEGIN
18     IF EXISTS (SELECT Request.doctor_id
19                FROM Request
20                WHERE Request.r_number = old.request_number
21                      AND new.doctor_id = Request.doctor_id)
22    THEN
23        CALL invalid_doctor_assignment();
24    END IF;
25 END$$
26
27 DELIMITER ;
```

To ensure the above, a trigger must be made both for the insert and the update. The trigger will select the doctor\_id from the request that is being inserted/updated in the table Study. If the doctor\_id is the same that is going to be used for the study, then an error is raised.

To prove that the trigger is working, the following instructions are executed

```
1 INSERT INTO Request values(16, 023, 201, '2017-11-13');
2 INSERT INTO Study values(16, 'teste', '2017-11-14', 201, 'S89RE4', '
  Medtronic');
3 UPDATE Study SET doctor_id = 202 WHERE request_number = 10 AND
  description = 'echography left knee';
```

And the final output is

```
1 ERROR 1305 (42000): PROCEDURE ist1xxxxx.invalid_doctor_assignment does
  not exist
```

## 5.ii Trigger

Prevent someone from trying to associate a device to a patient in overlapping periods.

```
1 DELIMITER $$
2
3 CREATE TRIGGER overlapping_periods_insert BEFORE INSERT ON Wears
4 FOR EACH ROW
5 BEGIN
6     IF EXISTS (SELECT snum, manuf, p_start, p_end
7                 FROM Wears
8                 WHERE new.snum = snum
9                       AND new.manuf = manuf
10                      AND NOT (new.p_end <= p_start OR new.p_start >= p_end))
11     THEN
12         SIGNAL SQLSTATE '45000' SET message_text = 'Overlapping periods';
13     END IF;
14 END$$
15
16 CREATE TRIGGER overlapping_periods_update BEFORE UPDATE ON Wears
17 FOR EACH ROW
18 BEGIN
19     IF old.snum = new.snum
20         AND old.manuf = new.manuf
21         AND NOT (new.p_end <= old.p_start OR new.p_start >= old.p_end)
22     THEN
23         SIGNAL SQLSTATE '45000' SET message_text = 'Overlapping periods';
24     END IF;
25 END$$
26
27 DELIMITER ;
```

When updating or inserting a row, a SELECT is ran on the table Wears to get the entries with the same snum and manufacturer in which the wearing start catches the end period of the new entry or the wearing end catches the start period of the new entry. If such entry exists, an error message is fired.

In the database, the patient 159 is wearing the device Proteus A230E5 from 2017-06-10 to 2017-09-15. Doing the following instructions will overlap the periods on insert.

```
1 INSERT INTO Period values('2017-06-01', '2017-06-12');
2 INSERT INTO Wears values('2017-06-01', '2017-06-12', 055, 'A230E5', '
  Proteus');
```

The patient 159 is wearing the device Proteus A230E5 from 2017-06-10 to 2017-09-15. Updating patient 12 with p\_start = 2017-09-14 will create an overlap.

```
1 UPDATE Wears SET p_start = '2017-09-14' WHERE p_start = '2017-09-16' AND
  p_end='2017-10-02' and patient = 012;
```

Both instructions produce the output:

```
1 ERROR 1644 (45000): Overlapping periods
```

## 6. Function

Given the (series\_id, index) of an Element A, and the coordinates (x1, y1, x2, y2) of a Region B, returns true if any region of the element A overlaps with Region B, and false otherwise.

```
1 DELIMITER $$
2
3 CREATE FUNCTION region_overlaps_element(
4     s_id integer,
5     s_index integer,
6     x1_B numeric(20,2),
7     y1_B numeric(20,2),
8     x2_B numeric(20,2),
9     y2_B numeric(20,2))
10 RETURNS integer
11 BEGIN
12     IF EXISTS (SELECT      x1, y1, x2, y2
13                FROM        Region
14                WHERE        series_id = s_id
15                            AND elem_index = s_index
16                            AND (x2_B <= x1
17                                OR x1_B >= x2
18                                OR y2_B <= y1
19                                OR y1_B >= y2))
20     THEN
21         RETURN 0;
22     ELSE
23         RETURN 1;
24     END IF;
25 END$$
26
27 DELIMITER ;
```

The SELECT statement returns the region A if it doesn't overlap B, in that case it returns FALSE.

In the database, the region with the series\_id 1, has the following coordinates (0.70, 0.30, 0.75, 0.35). Executing the following instruction produces a FALSE output.

```
1 SELECT region_overlaps_element(01, 01, 0.5, 0.5, 0.6, 0.6);
```

Whereas the next one returns TRUE:

```
1 SELECT region_overlaps_element(13, 01, 0.50, 0.80, 0.60, 0.85);
```