Information Systems and Databases

Report 2nd Project Assignment

Health care center database

15/11/2017

1st Semester – 2017/2018

Professors: Mário Gaspar da Silva
Francisco Regateiro
Miguel Amaral

Group number 49:

81731 – Catarina Aleixo
81138 – João Ramiro
81567 – José Miragaia

I.   Create the database

```
create table Patient
         (patient_number  varchar(255),
          patient_name   varchar(255),
          birthday   date,
          address   varchar(255),
          primary key(patient_number));

create table Doctor
         (patient_number   varchar(255),
          doctor_id   varchar(255),
          primary key(doctor_id),
          foreign key (patient_number) references Patient(patient_number));

create table Device
         (serialnum varchar(255),
          manufacturer varchar(255),
          model varchar(255),
          primary key(serialnum, manufacturer));

create table Sensor
         (serialnum varchar(255),
          manufacturer varchar(255),
          units varchar(255),
          primary key(serialnum, manufacturer),
          foreign key(serialnum, manufacturer) references Device(serialnum, manufacturer));

create table Reading
         (serialnum varchar(255),
          manufacturer varchar(255),
          read_datetime timestamp,
          value float(10,2),
          primary key(serialnum, manufacturer, read_datetime),
          foreign key(serialnum, manufacturer) references Sensor(serialnum, manufacturer));

create table Period
         (start_date timestamp,
          end_date timestamp,
          primary key(start_date, end_date));

create table Wears
         (start_date timestamp,
          end_date timestamp,
          patient_number varchar(255),
          serialnum varchar(255),
          manufacturer varchar(255),
          primary key(start_date, end_date, patient_number),
          foreign key(start_date, end_date) references Period(start_date, end_date),
          foreign key(patient_number) references Patient(patient_number),
          foreign key(serialnum, manufacturer) references Device(serialnum, manufacturer));

create table Request
         (request_number int(10) UNSIGNED,
          patient_number varchar(255),
          doctor_id varchar(255),
          request_date date,
          primary key(request_number),
          foreign key(patient_number) references Patient(patient_number),
          foreign key(doctor_id) references Doctor(doctor_id));

create table Study
         (request_number int(10) UNSIGNED,
          description varchar(255),
          study_date date,
```

```
            doctor_id varchar(255),
            manufacturer varchar(255),
            serialnum varchar(255),
            primary key(request_number, description),
            foreign key(request_number) references Request(request_number),
            foreign key(doctor_id) references Doctor(doctor_id),
            foreign key(serialnum, manufacturer) references Device(serialnum, manufacturer));

create table Series
            (series_id int(10) UNSIGNED,
            series_name varchar(255),
            base_url varchar(255),
            request_number int(10) UNSIGNED,
            description varchar(255),
            primary key(series_id),
            foreign key(request_number, description) references Study(request_number, description));

create table Element
            (series_id int(10) UNSIGNED,
            elem_index int(10) UNSIGNED,
            primary key(series_id, elem_index),
            foreign key(series_id) references Series(series_id));

create table Region
            (series_id int(10) UNSIGNED,
            elem_index int(10) UNSIGNED,
            x1 float(4,3),
            y1 float(4,3),
            x2 float(4,3),
            y2 float(4,3),
            primary key(series_id, elem_index, x1, y1, x2, y2),
            foreign key(series_id, elem_index) Element(series_id, elem_index));
```

II. Querys

1. Query to retrieve the name(s) of patient(s) with the highest number of readings of units of "LDL cholesterol in mg/dL" above 200 in the past 90 days

```
select patient_name
from Patient natural join Reading natural join Sensor natural join Wears
where value > 200 and units like 'LDL cholesterol in mg/dL' and
        TIMESTAMPDIFF(day, read_datetime, CURRENT_TIMESTAMP()) <= 90
group by patient_name
having count(value) >= all (select count(value)
                                from Patient natural join Reading natural join Sensor natural join Wears
                                where value > 200 and units like 'LDL cholesterol in mg/dL' and
                                TIMESTAMPDIFF(day, read_datetime, CURRENT_TIMESTAMP()) <= 90
                        group by patient_name);
```

2. Query to retrieve the name(s) of patient(s) who have been subject of studies with all devices of manufacturer "Medtronic" in the past calendar year

```
select patient_name
from Patient as p
where not exists (select serialnum
                 from Device as d
                 where manufacturer like 'Medtronic'
                 and serialnum not in (select serialnum
                                       from Study as s, Request as r, Patient as p2
                                       where s.request_number = r.request_number and
                                       r.patient_number = p2.patient_number and
                                       YEAR(s.study_date) = YEAR(CURRENT_DATE()) – 1 and
                                       p.patient_name = p2.patient_name));
```

III. Triggers

In this version of *mysql*, it's necessary to create two triggers: one for inserting data into Study table and another one for update it.

1. Trigger to ensure that a doctor who prescribes an exam may not perform that same exam

```
delimiter $$
create trigger prevent_insert before insert on Study
for each row
begin
        if exists(select *
                from Request
                where request_number = new.request_number and doctor_id = new.doctor_id) then
                call doctor_who_prescribes_an_exame_cannot_perform_the_same_exam();
        end if;
end$$
delimiter ;
```

```
delimiter $$
create trigger prevent_update before update on Study
for each row
begin
        if exists(select *
                from Request
                where request_number = new.request_number and doctor_id = new.doctor_id) then
                call doctor_who_prescribes_an_exame_cannot_perform_the_same_exam();
        end if;
end$$
delimiter ;
```

2. Trigger to prevent someone from trying to associate a device to a patient in overlapping periods. Additionally, when this event occurs, a text message "Overlapping periods" will be thrown

```
delimiter $$
create trigger prevent_device_association_insert before insert on Wears
for each row
begin
  if exists(select *
        from Wears
        where serialnum = new.serialnum and manufacturer = new.manufacturer and
    (((TIMESTAMPDIFF(second, start_date, new.end_date) >= 0) and (TIMESTAMPDIFF(second, new.end_date, end_date) >= 0)) or
    ((TIMESTAMPDIFF(second, start_date, new.start_date) >= 0) and (TIMESTAMPDIFF(second, new.start_date, end_date) >= 0)) or
    ((TIMESTAMPDIFF(second, new.start_date, start_date) >= 0) and (TIMESTAMPDIFF(second, end_date, new.end_date) >= 0)))) then

                signal sqlstate '45000' set message_text = 'Overlapping Periods';
        end if;
end$$
delimiter ;
```

```
delimiter $$
```

```
create trigger prevent_device_association_insert before update on Wears
for each row
begin
    if exists(select *
            from Wears
            where serialnum = new.serialnum and manufacturer = new.manufacturer and
        (((TIMESTAMPDIFF(second, start_date, new.end_date) >= 0) and (TIMESTAMPDIFF(second, new.end_date, end_date) >= 0)) or
        ((TIMESTAMPDIFF(second, start_date, new.start_date) >= 0) and (TIMESTAMPDIFF(second, new.start_date, end_date) >= 0)) or
        ((TIMESTAMPDIFF(second, new.start_date, start_date) >= 0) and (TIMESTAMPDIFF(second, end_date, new.end_date) >= 0)))) then

                        signal sqlstate '45000' set message_text = 'Overlapping Periods';
            end if;
end$$
delimiter ;
```

IV. Function

```
delimiter $$
create function region_overlaps_element(series_id_A int(10), elem_index_A int(10), x1_B float(4,3), y1_B float(4,3), x2_B
float(4,3), y2_B float(4,3))
returns int
begin
        declare overlaps int;
        select count(r.elem_index) into overlaps
        from region as r
        where r.series_id = series_id_A and r.elem_index = elem_index_A and
            ((r.x1 < x2_B) and (r.x2 > x1_B) and (r.y1 < y2_B) and (r.y2 > y1_B));
        if overlaps > 0 then
                set overlaps = 1;
        end if;
        return overlaps;
end$$
delimiter ;
```

This function *region_overlaps_element()* checks if a given Region B (x1_B, y1_B, x2_B, y2_B) overlaps with any region of
Element A (series_id_A, elem_index_A). If overlaps return 1 (means TRUE). Otherwise returns 0 (means FALSE).
NOTE: it's assumed that: x1 < x2 and y1< y2.

V. Populate tables

```
insert into Patient values ('P-1', 'John Smith', '1990-07-01', 'London');
insert into Patient values ('P-2', 'Roger Smith', '1987-05-05', 'Liverpool');
insert into Patient values ('P-3', 'James Bond', '1975-08-30', 'Bristol');
insert into Patient values ('P-4', 'Kelly Fernandez', '1985-10-06', 'London');
insert into Patient values ('P-5', 'Liz Wang', '1987-12-25', 'Birmingham');
insert into Patient values ('P-6', 'Richard Gun', '1965-11-15', 'Manchester');
insert into Patient values ('P-7', 'Gisele Joly', '1950-02-07', 'Manchester');
insert into Patient values ('P-8', 'Isaac Newton', '1966-04-09', 'Sheffield');
insert into Patient values ('P-9', 'Lilian Holmes', '1955-06-20', 'Liverpool');
insert into Patient values ('P-10', 'Wellington Ramirez', '1959-09-14', 'London');

insert into Doctor values ('P-1', 'D-101');
insert into Doctor values ('P-2', 'D-102');
insert into Doctor values ('P-3', 'D-103');
insert into Doctor values ('P-4', 'D-104');
insert into Doctor values ('P-5', 'D-105');
insert into Doctor values ('P-6', 'D-106');
insert into Doctor values ('P-7', 'D-107');
insert into Doctor values ('P-8', 'D-108');
insert into Doctor values ('P-9', 'D-109');
insert into Doctor values ('P-10', 'D-110');
```

insert into Device values ('S1', 'Siemens', 's6373');
insert into Device values ('B1', 'Bosch', 'b7566');
insert into Device values ('A1', 'Airsense', 'a9800');
insert into Device values ('A2', 'Airsense', 'a7658');
insert into Device values ('A3', 'Airsense', 'a3333');
insert into Device values ('A4', 'Airsense', 'a3333');
insert into Device values ('A5', 'Airsense', 'a9800');
insert into Device values ('S2', 'Siemens', 's3421');
insert into Device values ('B2', 'Bosch', 'b8765');
insert into Device values ('T1', 'Thermotec', 't2990');
insert into Device values ('T2', 'Thermotec', 't2990');
insert into Device values ('H1', 'Honeywell', 'h6776');
insert into Device values ('H2', 'Honeywell', 'h4444');
insert into Device values ('E1', 'Envitec', 'e5876');
insert into Device values ('E2', 'Envitec', 'e5876');
insert into Device values ('M1', 'Medtronic', 'm7891');
insert into Device values ('M2', 'Medtronic', 'm4536');
insert into Device values ('M3', 'Medtronic', 'm6543');
insert into Device values ('M4', 'Medtronic', 'm9023');
insert into Device values ('M5', 'Medtronic', 'm7123');
insert into Device values ('S3', 'Siemens', 's4040');
insert into Device values ('S4', 'Siemens', 's3131');
insert into Device values ('S5', 'Siemens', 's1111');
insert into Device values ('B3', 'Bosch', 'b4589');
insert into Device values ('B4', 'Bosch', 'b3232');
insert into Device values ('B5', 'Bosch', 'b4455');
insert into Device values ('I1', 'Iberdata', 'i3030');
insert into Device values ('I2', 'Iberdata', 'i9009');
insert into Device values ('I3', 'Iberdata', 'i7050');
insert into Device values ('I4', 'Iberdata', 'i3553');
insert into Device values ('I5', 'Iberdata', 'i7005');
insert into Device values ('I6', 'Iberdata', 'i6531');

insert into Sensor values ('S1', 'Siemens', 'LDL cholesterol in mg/dL');
insert into Sensor values ('B1', 'Bosch', 'Creatinine in mg/L');
insert into Sensor values ('A1', 'Airsense', 'HDL cholesterol in mg/dL');
insert into Sensor values ('A2', 'Airsense', 'Hemoglobin in g/dL');
insert into Sensor values ('A3', 'Airsense', 'Creatinine in mg/L');
insert into Sensor values ('A4', 'Airsense', 'Creatinine in mg/L');
insert into Sensor values ('A5', 'Airsense', 'Hemoglobin in g/dL');
insert into Sensor values ('S2', 'Siemens', 'HDL cholesterol in mg/dL');
insert into Sensor values ('B2', 'Bosch', 'LDL cholesterol in mg/dL');
insert into Sensor values ('T1', 'Thermotec', 'Temperature in ºC');
insert into Sensor values ('T2', 'Thermotec', 'Temperature in ºC');
insert into Sensor values ('H1', 'Honeywell', 'Glucose level in mmol/L');
insert into Sensor values ('H2', 'Honeywell', 'Glucose level in mmol/L');
insert into Sensor values ('E1', 'Envitec', 'Hemoglobin in g/dL');
insert into Sensor values ('E2', 'Envitec', 'Creatinine in mg/L');

insert into Reading values ('S1', 'Siemens', '2017-10-30 18:20:00', 217);
insert into Reading values ('S1', 'Siemens', '2017-10-30 18:21:00', 225);
insert into Reading values ('S1', 'Siemens', '2017-10-15 13:12:11', 220);
insert into Reading values ('S1', 'Siemens', '2017-10-17 20:08:21', 201);
insert into Reading values ('B1', 'Bosch', '2017-10-15 10:17:55', 10.1);
insert into Reading values ('A1', 'Airsense', '2017-07-31 15:25:45', 77);
insert into Reading values ('A1', 'Airsense', '2017-08-7 08:54:07', 100);
insert into Reading values ('A2', 'Airsense', '2017-10-31 09:57:35', 12.8);
insert into Reading values ('A2', 'Airsense', '2017-03-17 11:47:59', 15.7);
insert into Reading values ('A3', 'Airsense', '2016-12-25 20:20:20', 8.2);
insert into Reading values ('A4', 'Airsense', '2017-01-20 23:17:05', 11.8);
insert into Reading values ('A5', 'Airsense', '2017-03-29 21:40:47', 17.2);
insert into Reading values ('S2', 'Siemens', '2017-01-29 19:20:27', 117);
insert into Reading values ('B2', 'Bosch', '2017-10-14 10:39:06', 224);
insert into Reading values ('T1', 'Thermotec', '2017-10-28 07:44:21', 35);
insert into Reading values ('T2', 'Thermotec', '2016-04-06 09:45:17', 36.5);
insert into Reading values ('T2', 'Thermotec', '2016-07-16 11:12:12', 34.7);
insert into Reading values ('H1', 'Honeywell', '2017-10-31 12:55:59', 5.9);

insert into Reading **values** ('H2', 'Honeywell', '2016-04-17 14:32:58', 7.5);
insert into Reading **values** ('E1', 'Envitec', '2016-01-30 16:27:44', 17.9);
insert into Reading **values** ('E2', 'Envitec', '2017-05-29 19:42:28', 12.2);

insert into Period **values** ('2017-04-01 10:00:01', '2017-10-31 17:18:19');
insert into Period **values** ('2017-01-17 08:11:17', '2017-10-25 19:00:91');
insert into Period **values** ('2016-01-02 10:08:00', '2030-12-31 00:00:00');
insert into Period **values** ('2017-03-04 12:34:55', '2030-12-31 00:00:00');
insert into Period **values** ('2015-05-08 22:30:05', '2030-12-31 00:00:00');
insert into Period **values** ('2016-02-26 13:08:45', '2030-12-31 00:00:00');
insert into Period **values** ('2016-07-31 15:03:41', '2017-09-17 13:00:07');
insert into Period **values** ('2016-08-26 21:00:23', '2017-02-23 17:55:42');
insert into Period **values** ('2017-04-06 12:07:44', '2030-12-31 00:00:00');
insert into Period **values** ('2016-01-12 00:00:00', '2017-11-02 07:15:18');
insert into Period **values** ('2017-05-10 11:12:15', '2030-12-31 00:00:00');
insert into Period **values** ('2016-04-2 08:11:17', '2016-12-15 00:00:00');
insert into Period **values** ('2016-03-20 20:17:25', '2017-01-01 10:00:10');
insert into Period **values** ('2016-01-07 12:00:15', '2016-04-28 17:17:17');
insert into Period **values** ('2016-01-02 09:30:27', '2016-01-07 18:20:24');
insert into Period **values** ('2017-02-01 10:10:10', '2020-02-01 00:00:00');
insert into Period **values** ('2017-11-01 00:00:00', '2017-11-05 00:00:00');

insert into Wears **values** ('2017-04-01 10:00:01', '2017-10-31 17:18:19', 'P-1', 'S1', 'Siemens');
insert into Wears **values** ('2017-01-17 08:11:17', '2017-10-25 19:00:21', 'P-2', 'B1', 'Bosch');
insert into Wears **values** ('2016-01-02 10:08:00', '2030-12-31 00:00:00', 'P-3', 'A1', 'AirSense');
insert into Wears **values** ('2017-03-04 12:34:55', '2030-12-31 00:00:00', 'P-4', 'A2', 'AirSense');
insert into Wears **values** ('2015-05-08 22:30:05', '2030-12-31 00:00:00', 'P-5', 'A3', 'AirSense');
insert into Wears **values** ('2016-02-26 13:08:45', '2030-12-31 00:00:00', 'P-6', 'A4', 'AirSense');
insert into Wears **values** ('2016-07-31 15:03:41', '2017-09-17 13:00:07', 'P-7', 'A5', 'AirSense');
insert into Wears **values** ('2016-08-26 21:00:23', '2017-02-23 17:55:42', 'P-8', 'S2', 'Siemens');
insert into Wears **values** ('2017-04-06 12:07:44', '2030-12-31 00:00:00', 'P-9', 'B2', 'Bosch');
insert into Wears **values** ('2016-01-12 00:00:00', '2017-11-02 07:15:18', 'P-10', 'T1', 'Thermotec');
insert into Wears **values** ('2017-05-10 11:12:15', '2030-12-31 00:00:00', 'P-8', 'H1', 'Honeywell');
insert into Wears **values** ('2016-04-02 08:11:10', '2016-12-15 00:01:10', 'P-2', 'T2', 'Thermotec');
insert into Wears **values** ('2016-03-20 20:17:25', '2017-01-01 10:00:10', 'P-1', 'H2', 'Honeywell');
insert into Wears **values** ('2016-01-07 12:00:15', '2016-04-28 17:17:17', 'P-7', 'E1', 'Envitec');
insert into Wears **values** ('2016-01-02 09:30:27', '2016-01-07 18:20:24', 'P-10', 'E2', 'Envitec');

insert into Request **values** (1, 'P-1', 'D-101', '2016-02-07');
insert into Request **values** (2, 'P-2', 'D-102', '2017-07-17');
insert into Request **values** (3, 'P-3', 'D-103', '2016-05-15');
insert into Request **values** (4, 'P-4', 'D-104', '2017-08-31');
insert into Request **values** (5, 'P-5', 'D-105', '2016-10-16');
insert into Request **values** (6, 'P-6', 'D-106', '2016-09-21');
insert into Request **values** (7, 'P-7', 'D-107', '2017-04-19');
insert into Request **values** (8, 'P-8', 'D-108', '2017-01-21');
insert into Request **values** (9, 'P-9', 'D-109', '2016-12-16');
insert into Request **values** (10, 'P-10' 'D-110', '2017-06-25');
insert into Request **values** (11, 'P-5', 'D-105', '2017-05-09');
insert into Request **values** (12, 'P-7', 'D-107', '2017-10-24');
insert into Request **values** (13, 'P-8', 'D-108', '2016-07-05');
insert into Request **values** (14, 'P-2', 'D-102', '2016-04-02');
insert into Request **values** (15, 'P-1', 'D-101', '2016-10-09');
insert into Request **values** (16, 'P-3', 'D-103', '2017-09-06');
insert into Request **values** (17, 'P-4', 'D-104', '2017-03-11');
insert into Request **values** (18, 'P-3', 'D-103', '2016-02-10');
insert into Request **values** (19, 'P-3', 'D-103', '2016-04-03');
insert into Request **values** (20, 'P-3', 'D-103', '2016-07-20');
insert into Request **values** (21, 'P-3', 'D-103', '2016-09-07');

insert into Study **values** (1, 'X-ray both feet', '2016-03-05', 'D-102', 'Medtronic', 'M1');
insert into Study **values** (2, 'Ecography both feet', '2017-08-30', 'D-101', 'Medtronic', 'M2');
insert into Study **values** (3, 'Endoscopy esophagus', '2016-06-02', 'D-105', 'Medtronic', 'M3');
insert into Study **values** (4, 'Endoscopy stomach', '2017-10-07', 'D-103', 'Medtronic', 'M4');
insert into Study **values** (5, 'Echocardiography', '2016-10-31', 'D-104', 'Medtronic', 'M5');
insert into Study **values** (6, 'Mammography', '2016-03-05', 'D-107', 'Iberdata', 'I6');
insert into Study **values** (7, 'Colonoscopy', '2017-05-12', 'D-110', 'Siemens', 'S3');

insert into Study **values** (8, 'Magnetic Resonance both shoulders', '2017-02-25', 'D-101', 'Siemens', 'S4');
insert into Study **values** (9, 'Magnetic Resonance both knees', '2017-01-08', 'D-108', 'Siemens', 'S5');
insert into Study **values** (10, 'Electrocardiogram', '2017-07-27', 'D-109', 'Bosch', 'B3');
insert into Study **values** (11, 'Electrocardiogram', '2017-10-08', 'D-106', 'Bosch', 'B4');
insert into Study **values** (12, 'Prostate Specific Antigen (PSA test)', '2017-10-31', 'D-102', 'Bosch', 'B5');
insert into Study **values** (13, 'Mammography', '2017-07-27', 'D-107', 'Iberdata', 'I1');
insert into Study **values** (14, 'Colonoscopy', '2017-05-09', 'D-105', 'Iberdata', 'I2');
insert into Study **values** (15, 'Echocardiography', '2017-10-29', 'D-108', 'Iberdata', 'I3');
insert into Study **values** (16, 'X-ray chest', '2017-10-01', 'D-102', 'Iberdata', 'I4');
insert into Study **values** (17, 'X-ray both elbows', '2017-05-27', 'D-101', 'Iberdata', 'I5');
insert into Study **values** (18, 'X-ray right foot', '2016-03-25', 'D-104', 'Medtronic', 'M1');
insert into Study **values** (19, 'Ecography left foot', '2016-08-31', 'D-109', 'Medtronic', 'M2');
insert into Study **values** (20, 'Endoscopy stomach', '2016-10-10', 'D-110', 'Medtronic', 'M4');
insert into Study **values** (21, 'Echocardiography', '2016-10-17', 'D-105', 'Medtronic', 'M5');

insert into Series **values** (1, 'X-ray right foot', 'http://www.healthcarecentre/11/', 1, 'X-ray both feet');
insert into Series **values** (2, 'X-ray left foot', 'http://www.healthcarecentre/21/', 1, 'X-ray both feet');
insert into Series **values** (3, 'Ecography right foot', 'http://www.healthcarecentre/32/', 2, 'Ecography both feet');
insert into Series **values** (4, 'Ecography left foot', 'http://www.healthcarecentre/42/', 2, 'Ecography both feet');
insert into Series **values** (5, 'Endoscopy esophagus', 'http://www.healthcarecentre/53/', 3, 'Endoscopy esophagus');
insert into Series **values** (6, 'Endoscopy stomach', 'http://www.healthcarecentre/64/', 4, 'Endoscopy stomach');
insert into Series **values** (7, 'Echocardiography', 'http://www.healthcarecentre/75/', 5, 'Echocardiography');
insert into Series **values** (8, 'Mammography', 'http://www.healthcarecentre/86/', 6, 'Mammography');
insert into Series **values** (9, 'Colonoscopy', 'http://www.healthcarecentre/97/', 7, 'Colonoscopy');
insert into Series **values** (10, 'Magnetic Resonance right shoulder', 'http://www.healthcarecentre/108/', 8, 'Magnetic Resonance both shoulders');
insert into Series **values** (11, 'Magnetic Resonance left shoulder', 'http://www.healthcarecentre/118/', 8, 'Magnetic Resonance both shoulders');
insert into Series **values** (12, 'Magnetic Resonance right knee', 'http://www.healthcarecentre/129/', 9, 'Magnetic Resonance both shoulders');
insert into Series **values** (13, 'Magnetic Resonance left knee', 'http://www.healthcarecentre/139/', 9, 'Magnetic Resonance both shoulders');
insert into Series **values** (14, 'Electrocardiogram', 'http://www.healthcarecentre/1410/', 10, 'Electrocardiogram');
insert into Series **values** (15, 'Electrocardiogram', 'http://www.healthcarecentre/1511/', 11, 'Electrocardiogram');
insert into Series **values** (16, 'Prostate Specific Antigen (PSA test)', 'http://www.healthcarecentre/1612/', 12, 'Prostate Specific Antigen (PSA test)');
insert into Series **values** (17, 'Mammography', 'http://www.healthcarecentre/1713/', 13, 'Mammography');
insert into Series **values** (18, 'Colonoscopy', 'http://www.healthcarecentre/1814/', 14, 'Colonoscopy');
insert into Series **values** (19, 'Echocardiography', 'http://www.healthcarecentre/1915/', 15, 'Echocardiography');
insert into Series **values** (20, 'X-ray chest', 'http://www.healthcarecentre/2016/', 16, 'X-ray chest');
insert into Series **values** (21, 'X-ray right elbow', 'http://www.healthcarecentre/2117/', 17, 'X-ray both elbows');
insert into Series **values** (22, 'X-ray left elbow', 'http://www.healthcarecentre/2217/', 17, 'X-ray both elbows');
insert into Series **values** (23, 'X-ray right foot', 'http://www.healthcarecentre/2318/', 18, 'X-ray right foot');
insert into Series **values** (24, 'Ecography left foot', 'http://www.healthcarecentre/2419/', 19, 'Ecography left foot');
insert into Series **values** (25, 'Endoscopy stomach', 'http://www.healthcarecentre/2520/', 20, 'Endoscopy stomach');
insert into Series **values** (26, 'Echocardiography', 'http://www.healthcarecentre/2621/', 21, 'Echocardiography');

insert into Element **values** (1, 1);
insert into Element **values** (1, 2);
insert into Element **values** (1, 3);
insert into Element **values** (2, 1);
insert into Element **values** (2, 2);
insert into Element **values** (3, 1);
insert into Element **values** (3, 2);
insert into Element **values** (3, 3);
insert into Element **values** (3, 4);
insert into Element **values** (4, 1);
insert into Element **values** (4, 2);
insert into Element **values** (5, 1);
insert into Element **values** (5, 2);
insert into Element **values** (6, 1);
insert into Element **values** (6, 2);
insert into Element **values** (7, 1);
insert into Element **values** (7, 2);
insert into Element **values** (8, 1);
insert into Element **values** (8, 2);
insert into Element **values** (8, 3);

insert into Element **values** (8, 4);
insert into Element **values** (9, 1);
insert into Element **values** (9, 2);
insert into Element **values** (10, 1);
insert into Element **values** (10, 2);
insert into Element **values** (11, 1);
insert into Element **values** (11, 2);
insert into Element **values** (12, 1);
insert into Element **values** (12, 2);
insert into Element **values** (13, 1);
insert into Element **values** (13, 2);
insert into Element **values** (14, 1);
insert into Element **values** (14, 2);
insert into Element **values** (15, 1);
insert into Element **values** (15, 2);
insert into Element **values** (16, 1);
insert into Element **values** (16, 2);
insert into Element **values** (17, 1);
insert into Element **values** (17, 2);
insert into Element **values** (18, 1);
insert into Element **values** (18, 2);
insert into Element **values** (19, 1);
insert into Element **values** (19, 2);
insert into Element **values** (20, 1);
insert into Element **values** (20, 2);
insert into Element **values** (21, 1);
insert into Element **values** (21, 2);
insert into Element **values** (22, 1);
insert into Element **values** (22, 2);
insert into Element **values** (22, 3);
i**nsert into** Element **values** (22, 4);

insert into Region **values** (1, 1, 0.107, 0.638, 0.180, 0.762);
insert into Region **values** (1, 2, 0.230, 0.567, 0.443, 0.899);
insert into Region **values** (1, 3, 0.432, 0.565, 0.118, 0.779);
insert into Region **values** (2, 1, 0.508, 0.234, 0.734, 0.923);
insert into Region **values** (2, 1, 0.100, 0.100, 0.230, 0.230);
insert into Region **values** (2, 2, 0.214, 0.657, 0.978, 0.999);
insert into Region **values** (3, 1, 0.115, 0.376, 0.647, 0.762);
insert into Region **values** (3, 2, 0.534, 0.980, 0.234, 0.762);
insert into Region **values** (3, 3, 0.675, 0.192, 0.567, 0.762);
insert into Region **values** (3, 4, 0.453, 0.876, 0.290, 0.834);
insert into Region **values** (4, 1, 0.489, 0.789, 0.117, 0.878);
insert into Region **values** (4, 2, 0.222, 0.333, 0.444, 0.555);
insert into Region **values** (5, 1, 0.654, 0.218, 0.905, 0.675);
insert into Region **values** (5, 2, 0.232, 0.798, 0.190, 0.458);
insert into Region **values** (6, 1, 0.616, 0.717, 0.818, 0.919);
insert into Region **values** (6, 2, 0.111, 0.999, 0.222, 0.777);
insert into Region **values** (7, 1, 0.668, 0.669, 0.888, 0.889);
insert into Region **values** (7, 2, 0.558, 0.475, 0.222, 0.332);
insert into Region **values** (8, 1, 0.142, 0.173, 0.194, 0.289);
insert into Region **values** (8, 2, 0.765, 0.975, 0.333, 0.777);
insert into Region **values** (8, 3, 0.211, 0.991, 0.224, 0.664);
insert into Region **values** (8, 4, 0.168, 0.845, 0.333, 0.456);
insert into Region **values** (9, 1, 0.121, 0.343, 0.565, 0.787);
insert into Region **values** (9, 2, 0.099, 0.677, 0.345, 0.899);
insert into Region **values** (10, 1, 0.321, 0.654, 0.765, 0.987);
insert into Region **values** (10, 2, 0.545, 0.878, 0.688, 0.727);
insert into Region **values** (11, 1, 0.646, 0.789, 0.889, 0.989);
insert into Region **values** (11, 2, 0.109, 0.234, 0.565, 0.898);
insert into Region **values** (12, 1, 0.212, 0.245, 0.656, 0.787);
insert into Region **values** (12, 2, 0.893, 0.909, 0.345, 0.897);
insert into Region **values** (13, 1, 0.666, 0.999, 0.222, 0.666);
insert into Region **values** (13, 2, 0.878, 0.989, 0.356, 0.676);
insert into Region **values** (14, 1, 0.111, 0.225, 0.348, 0.987);
insert into Region **values** (14, 2, 0.590, 0.334, 0.212, 0.455);
insert into Region **values** (15, 1, 0.690, 0.990, 0.213, 0.676);

insert into Region **values** (15, 2, 0.276, 0.387, 0.432, 0.878);
insert into Region **values** (16, 1, 0.175, 0.345, 0.478, 0.889);
insert into Region **values** (16, 2, 0.289, 0.367, 0.878, 0.995);
insert into Region **values** (17, 1, 0.567, 0.890, 0.345, 0.789);
insert into Region **values** (17, 2, 0.465, 0.443, 0.556, 0.954);
insert into Region **values** (18, 1, 0.586, 0.669, 0.665, 0.753);
insert into Region **values** (18, 2, 0.123, 0.456, 0.789, 0.890);
insert into Region **values** (19, 1, 0.643, 0.717, 0.542, 0.999);
insert into Region **values** (19, 2, 0.545, 0.878, 0.688, 0.727);
insert into Region **values** (20, 1, 0.534, 0.796, 0.968, 0.998);
insert into Region **values** (20, 2, 0.321, 0.543, 0.765, 0.987);
insert into Region **values** (21, 1, 0.199, 0.299, 0.399, 0.599);
insert into Region **values** (21, 2, 0.209, 0.309, 0.409, 0.509);
insert into Region **values** (22, 1, 0.534, 0.756, 0.867, 0.957);
insert into Region **values** (22, 2, 0.476, 0.874, 0.565, 0.878);
insert into Region **values** (22, 3, 0.223, 0.345, 0.678, 0.967);
insert into Region **values** (22, 4, 0.234, 0.678, 0.456, 0.789);

VI. Expected Results

- Figure 1 shows the result of the first query, which are the patients with the highest number of readings of 'LDL cholesterol in mg/dL' above 200 in the past 90 days:

```
mysql> select patient_name
    -> from Patient natural join Reading natural join Sensor natural join Wears
    -> where value > 200 and units like 'LDL cholesterol in mg/dL' and
    ->      TIMESTAMPDIFF(day, read_datetime, CURRENT_TIMESTAMP()) <= 90
    -> group by patient_name
    -> having count(value) >= all(select count(value)
    ->                            from Patient natural join Reading natural join Sensor natural join Wears
    ->                            where value > 200 and units like 'LDL cholesterol in mg/dL' and
    ->                              TIMESTAMPDIFF(day, read_datetime, CURRENT_TIMESTAMP()) <= 90
    ->                            group by patient_name);
+--------------+
| patient_name |
+--------------+
| John Smith   |
+--------------+
1 row in set (0.00 sec)
```

Figure 1 – Result of the first query

- Figure 2 shows the result of the second query, which are the patients who have been subject of studies with all devices of manufacturer 'Medtronic':

```
mysql> select patient_name
    -> from Patient as p
    -> where not exists(select serialnum
    ->                  from Device as d
    ->                  where manufacturer like 'Medtronic'
    ->                  and serialnum not in (select serialnum
    ->                                        from Study as s, Request as r, Patient as p2
    ->                                        where s.request_number = r.request_number and
    ->                                        r.patient_number = p2.patient_number and
    ->                                        YEAR(s.study_date) = YEAR(CURRENT_DATE()) - 1 and
    ->                                        p.patient_name = p2.patient_name));
+--------------+
| patient_name |
+--------------+
| James Bond   |
+--------------+
1 row in set (0.01 sec)
```

Figure 2 - Result of the second query

- Figures 3.1 and 3.2 show examples of the implementation of trigger 1, one for inserting a new record (prevent_insert) and another one for updating an existing row (prevent_update), in the table Study, respectively:

```
mysql> insert into Request values (22, 'P-1', 'D-101', '2017-05-07');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Study values (22, 'X-ray right foot', '2017-07-02', 'D-101','Medtronic', 'M1');
ERROR 1305 (42000): PROCEDURE healthcarecentre.doctor_who_prescribes_an_exame_cannot_perform_the_same_exam does not exist
```

Figure 4.1 - Trigger 1 (prevent_insert) example for trying to insert a record in Study table that has the same doctor that requested the exam

```
mysql> update Study set doctor_id = 'D-101' where request_number = 1;
ERROR 1305 (42000): PROCEDURE healthcarecentre.doctor_who_prescribes_an_exame_cannot_perform_the_same_exam does not exist
```

Figure 3.2 - Trigger 1 (prevent_update) example for updating an existing row, trying to change the doctor of a study to the same doctor that made the request

- Figures 4.1 and 4.2 show examples of the implementation of trigger 2, one for inserting a new record (prevent_device_association_insert) and another one for updating an existing row (prevent_device_association_update) in the table Wears, respectively:

```
mysql> insert into Period values ('2017-02-01 10:10:10', '2020-02-01 00:00:00');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Period values ('2016-01-01 10:10:10', '2017-01-01 00:00:00');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Wears values ('2017-02-01 10:10:10', '2020-02-01 00:00:00', 'P-1', 'A1', 'AirSense');
ERROR 1644 (45000): Overlapping Periods
mysql> insert into Wears values ('2017-02-01 10:10:10', '2020-02-01 00:00:00', 'P-1', 'A2', 'AirSense');
ERROR 1644 (45000): Overlapping Periods
mysql> insert into Wears values ('2017-02-01 10:10:10', '2020-02-01 00:00:00', 'P-1', 'A5', 'AirSense');
ERROR 1644 (45000): Overlapping Periods
mysql> insert into Wears values ('2016-01-01 10:10:10', '2017-01-01 00:00:00', 'P-1', 'E2', 'Envitec');
ERROR 1644 (45000): Overlapping Periods
```

Figure 4.1 - Trigger 2 (prevent_device_association_insert) example for trying to insert records that associates the same device, in overlapping periods, for two different patients, in Wears table

```
mysql> update Wears
    -> set serialnum = 'S1', manufacturer = 'Siemens'
    -> where start_date = '2016-07-31 15:03:41' and end_date = '2017-09-17 13:00:07' and patient_number = 'P-7';
ERROR 1644 (45000): Overlapping Periods
mysql> update Wears
    -> set start_date = '2017-04-06 12:07:44', end_date = '2030-12-31 00:00:00', serialnum = 'B2', manufacturer = 'Bosch'
    -> where start_date = '2016-01-02 09:30:27' and end_date = '2016-01-07 18:20:24' and patient_number = 'P-10';
ERROR 1644 (45000): Overlapping Periods
```

Figure 4.2 - Trigger 2 (prevent_device_association_update) example for updating an existing row, trying to associate a device to different patients, in overlapping periods

- Figure 5.1, 5.2 and 5.3 show examples of the implementation of function *region_overlaps_element()*:

```
mysql> select region_overlaps_element(1, 1, 0.120, 0.700, 0.170, 0.750);
+--------------------------------------------------------+
| region_overlaps_element(1, 1, 0.120, 0.700, 0.170, 0.750) |
+--------------------------------------------------------+
|                                                      1 |
+--------------------------------------------------------+
1 row in set (0.00 sec)

mysql> select region_overlaps_element(1, 1, 0.100, 0.500, 0.200, 0.800);
+--------------------------------------------------------+
| region_overlaps_element(1, 1, 0.100, 0.500, 0.200, 0.800) |
+--------------------------------------------------------+
|                                                      1 |
+--------------------------------------------------------+
1 row in set (0.00 sec)
```

Figure 5.1 - Region B overlaps with region of element A – returns 1, that means TRUE

```
mysql> select region_overlaps_element(1, 1, 0.190, 0.638, 0.300, 0.638);
+-----------------------------------------------------------+
| region_overlaps_element(1, 1, 0.190, 0.638, 0.300, 0.638) |
+-----------------------------------------------------------+
|                                                         0 |
+-----------------------------------------------------------+
1 row in set (0.00 sec)

mysql> select region_overlaps_element(1, 1, 0.300, 0.125, 0.755, 0.800);
+-----------------------------------------------------------+
| region_overlaps_element(1, 1, 0.300, 0.125, 0.755, 0.800) |
+-----------------------------------------------------------+
|                                                         0 |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

Figure 5.2 – Region B don't overlap with region of element A – returns 0, that means FALSE

```
mysql> select region_overlaps_element(2, 1, 0.100, 0.100, 0.200, 0.200);
+-----------------------------------------------------------+
| region_overlaps_element(2, 1, 0.100, 0.100, 0.200, 0.200) |
+-----------------------------------------------------------+
|                                                         1 |
+-----------------------------------------------------------+
1 row in set (0.00 sec)

mysql> select region_overlaps_element(2, 1, 0.050, 0.050, 0.090, 0.090);
+-----------------------------------------------------------+
| region_overlaps_element(2, 1, 0.050, 0.050, 0.090, 0.090) |
+-----------------------------------------------------------+
|                                                         0 |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

Figure 5.3 – In this case, element 1 from series_id = 2, has two regions and in the first case the given region B overlaps with one of them and in the second case don't overlap with any of them