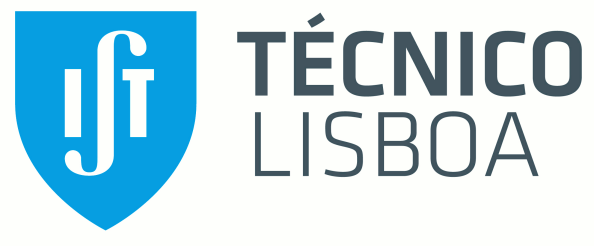


Sistemas de Informação e Bases de Dados

MEEC

Projeto 2 - Implementação da Base de Dados

Grupo N.º 30

Autores:

André Godinho – nº 84006
Diogo Rodrigues – n.º 84030
Pedro Moreira – n.º 84112

Turno:

Segunda-feira 14h00-15h30

9 de Novembro de 2018

Conteúdo

1 Criação da Base de Dados

```
drop table if exists produced_indicator;
drop table if exists test_procedure;
drop table if exists radiography;
drop table if exists performed;
drop table if exists procedures;
drop table if exists indicator;
drop table if exists prescription;
drop table if exists medication;
drop table if exists consult_diagnosis;
drop table if exists diagnosis_code;
drop table if exists participation;
drop table if exists consult;
drop table if exists animal;
drop table if exists generalization_species;
drop table if exists species;
drop table if exists assistant;
drop table if exists veterinary;
drop table if exists client;
drop table if exists phone_number;
drop table if exists person;

create table person
        (VAT integer,
         name varchar(255),
         address_street varchar(255),
         address_city varchar(255),
         address_zip varchar(255),
         primary key (VAT));

create table phone_number
        (VAT integer,
         phone integer,
         primary key (VAT, phone),
         foreign key (VAT) references person(VAT));

create table client
        (VAT integer,
         primary key (VAT),
         foreign key (VAT) references person(VAT));

create table veterinary
        (VAT integer,
         specialization varchar(255),
         bio varchar(255),
         primary key (VAT),
         foreign key (VAT) references person(VAT));

create table assistant
        (VAT integer,
```

```
        primary key(VAT),
        foreign key(VAT) references person(VAT));

create table species
       (name varchar(255),
        description varchar(255),
        primary key(name));

create table generalization_species
       (name1 varchar(255),
        name2 varchar(255),
        primary key(name1),
        foreign key(name1) references species(name),
        foreign key(name2) references species(name));

create table animal
       (name varchar(255),
        VAT integer,
        species_name varchar(255),
        colour varchar(255),
        gender varchar(255),
        birth_year timestamp,
        age integer,
        primary key(name, VAT),
        foreign key(VAT) references client(VAT)
               on delete cascade,
        foreign key(species_name) references species(name));

create table consult
       (name varchar(255),
        VAT_owner integer,
        date_timestamp timestamp,
        s varchar(255),
        o varchar(255),
        a varchar(255),
        p varchar(255),
        VAT_client integer,
        VAT_vet integer,
        weight numeric(20,2),
        primary key(name, VAT_owner, date_timestamp),
        foreign key(name, VAT_owner) references animal(name,
           VAT)
               on delete cascade,
        foreign key(VAT_client) references client(VAT)
               on delete cascade,
        foreign key(VAT_vet) references veterinary(VAT),
        check(weight >= 0));

create table participation
       (name varchar(255),
        VAT_owner integer,
```

```
        date_timestamp timestamp,
        VAT_assistant integer,
        primary key(name, VAT_owner, date_timestamp,
           VAT_assistant),
        foreign key(name, VAT_owner, date_timestamp)
           references consult(name, VAT_owner, date_timestamp)
               on delete cascade,
        foreign key(VAT_assistant) references assistant(VAT));

create table diagnosis_code
       (code varchar(255),
        name varchar(255),
        primary key(code));

create table consult_diagnosis
       (code varchar(255),
        name varchar(255),
        VAT_owner integer,
        date_timestamp timestamp,
        primary key(code, name, VAT_owner, date_timestamp),
        foreign key(name, VAT_owner, date_timestamp)
           references consult(name, VAT_owner, date_timestamp)
               on delete cascade,
        foreign key(code) references diagnosis_code(code));

create table medication
       (name varchar(255),
        lab varchar(255),
        dosage varchar(255),
        primary key(name, lab, dosage));

create table prescription
       (code varchar(255),
        name varchar(255),
        VAT_owner integer,
        date_timestamp timestamp,
        name_med varchar(255),
        lab varchar(255),
        dosage varchar(255),
        regime varchar(255),
        primary key(code, name, VAT_owner, date_timestamp,
           name_med, lab, dosage),
        foreign key(code, name, VAT_owner, date_timestamp)
           references consult_diagnosis(code, name, VAT_owner,
           date_timestamp)
               on delete cascade on update cascade,
        foreign key(name_med, lab, dosage) references
           medication(name, lab, dosage));

create table indicator
       (name varchar(255),
```

```
          reference_value numeric(20,2),
          units varchar(255),
          description varchar(255),
          primary key(name));

create table procedures
         (name varchar(255),
          VAT_owner integer,
          date_timestamp timestamp,
          num integer,
          description varchar(255),
          primary key(name, VAT_owner, date_timestamp, num),
          foreign key(name, VAT_owner, date_timestamp)
             references consult(name, VAT_owner, date_timestamp)
                 on delete cascade);

create table performed
         (name varchar(255),
          VAT_owner integer,
          date_timestamp timestamp,
          num integer,
          VAT_assistant integer,
          primary key(name, VAT_owner, date_timestamp, num),
          foreign key(name, VAT_owner, date_timestamp, num)
             references procedures(name, VAT_owner,
             date_timestamp, num)
                 on delete cascade,
          foreign key(VAT_assistant) references assistant(VAT));

create table radiography
         (name varchar(255),
          VAT_owner integer,
          date_timestamp timestamp,
          num integer,
          file varchar(255),
          primary key(name, VAT_owner, date_timestamp, num),
          foreign key(name, VAT_owner, date_timestamp, num)
             references procedures(name, VAT_owner,
             date_timestamp, num)
                 on delete cascade);

create table test_procedure
         (name varchar(255),
          VAT_owner integer,
          date_timestamp timestamp,
          num integer,
          type char(5),
          primary key(name, VAT_owner, date_timestamp, num),
          foreign key(name, VAT_owner, date_timestamp, num)
             references procedures(name, VAT_owner,
             date_timestamp, num)
```

```
                on delete cascade,
        check(num >= 1));

create table produced_indicator
        (name varchar(255),
         VAT_owner integer,
         date_timestamp timestamp,
         num integer,
         indicator_name varchar(255),
         value numeric(20,2),
         primary key(name, VAT_owner, date_timestamp, num,
            indicator_name),
         foreign key(name, VAT_owner, date_timestamp, num)
            references test_procedure(name, VAT_owner,
            date_timestamp, num)
                on delete cascade,
         foreign key(indicator_name) references
            indicator(name));
```

Código 1: Instrução SQL para a criação da base de dados

2 Preencher a Base de Dados

```
insert into person values (123456001, 'Frank'     , 'Dark
   Boulevard'   , 'Chicago'      , '1111-123');
insert into person values (123456002, 'Charles'   , 'Arsenal
   Street'   , 'London'       , '1100-123');
insert into person values (123456003, 'James'     , 'Flowers
   Avenue'   , 'San Francisco', '3300-123');
insert into person values (123456011, 'Andrew'    , 'Roses
   Boulevard'  , 'Chicago'      , '1111-123');
insert into person values (123456022, 'John Smith', 'Central
   Boulevard', 'London'       , '1100-123');
insert into person values (123456033, 'Ray'       , 'Second
   Boulevard' , 'San Francisco', '3300-123');
insert into person values (123456044, 'Mike'      , 'Happy
   Boulevard'  , 'Chicago'      , '1111-123');
insert into person values (123456004, 'Peter '    , 'Central
   Avenue'   , 'Seattle'      , '2200-123');
insert into person values (123456005, 'Jennifer'  , 'White
   Avenue'     , 'Chicago'      , '1111-124');
insert into person values (123456006, 'Jessica'   , 'Chelsea
   Street'   , 'London'       , '1100-124');
insert into person values (123456007, 'Caroline'  , 'Central
   Street'   , 'Seattle'      , '2200-124');
insert into person values (123456008, 'Bernard'   ,   'Central
   Street' , 'Seattle'      , '2200-125');
insert into person values (123456009, 'Anne'      ,  'Happy
   Boulevard' , 'Chicago'      , '1111-125');

insert into phone_number values (123456001, 961231231);
insert into phone_number values (123456002, 961231232);
insert into phone_number values (123456003, 961231233);
insert into phone_number values (123456011, 960000001);
insert into phone_number values (123456022, 960000002);
insert into phone_number values (123456033, 960000003);
insert into phone_number values (123456004, 961111114);
insert into phone_number values (123456005, 961111115);
insert into phone_number values (123456006, 961111116);
insert into phone_number values (123456007, 962222227);
insert into phone_number values (123456008, 962222228);
insert into phone_number values (123456009, 962222229);

insert into client values (123456001);
insert into client values (123456002);
insert into client values (123456003);
insert into client values (123456004);
insert into client values (123456011);
insert into client values (123456022);
insert into client values (123456033);
insert into client values (123456006);
insert into client values (123456009);
```

```
insert into veterinary values (123456004,'Oncologist' , 'Peter
   is a 30 year old specialist in Oncology.');
insert into veterinary values (123456005,'Nutrition'  ,
   'Jennifer is a 34 year old specialist in Nutrition.');
insert into veterinary values (123456006,'Orthopedics',
   'Jessica is a 26 year old specialist in Orthopedics.');
insert into veterinary values (123456022,'Oncologist' , 'John
   Smith is a 22 year old specialist in Oncologist.');

insert into assistant values (123456007);
insert into assistant values (123456008);
insert into assistant values (123456009);
insert into assistant values (123456022);

insert into species values ('Mammal'  , 'Distinguished from
   reptiles (including birds) by the possession of a neocortex
   (a region of the brain), hair, three middle ear bones, and
   mammary glands');
insert into species values ('Cat'     , 'Small, typically
   furry, carnivorous mammal');
insert into species values ('Dog'     , 'Part of the wolf-like
   canids and is the most widely abundant terrestrial
   carnivore');
insert into species values ('Bulldog' , 'Medium-sized breed of
   dog');
insert into species values ('Boxer'   , 'Medium-sized,
   short-haired breed of dog, developed in Germany');
insert into species values ('Pit Bull', 'Type of dog descended
   from bulldogs and terriers');
insert into species values ('Bird'    , 'Group of endothermic
   vertebrates');
insert into species values ('Aves'    , 'Group of endothermic
   vertebrates');

insert into generalization_species values ('Cat'     ,
   'Mammal');
insert into generalization_species values ('Dog'     ,
   'Mammal');
insert into generalization_species values ('Bulldog' , 'Dog');
insert into generalization_species values ('Boxer'   , 'Dog');
insert into generalization_species values ('Pit Bull', 'Dog');
insert into generalization_species values ('Bird'    , 'Aves');

insert into animal values ('Striker', 123456001, 'Bulldog' ,
   'brown' , 'male'  , '2008-11-05', 10);
insert into animal values ('Jackson', 123456022, 'Bulldog' ,
   'black' , 'female', '2009-12-30', 9);
insert into animal values ('Bob'    , 123456003, 'Bulldog' ,
   'white' , 'male'  , '2013-01-04', 1);
insert into animal values ('Thunder', 123456001, 'Cat'     ,
```

```
    'brown' , 'female', '2016-04-02', 2);
insert into animal values ('Sparky' , 123456002, 'Boxer'   ,
    'black' , 'female', '2014-10-10', 4);
insert into animal values ('Max'    , 123456006, 'Pit Bull',
    'yellow', 'male'  , '2007-10-15', 11);
insert into animal values ('Cookie' , 123456011, 'Bird'    ,
    'red'   , 'male'  , '2005-10-13', 13);
insert into animal values ('Rolls'  , 123456022, 'Bird'    ,
    'blue'  , 'male'  , '2013-05-01', 1);
insert into animal values ('Jim'    , 123456033, 'Bird'    ,
    'green' , 'female', '2010-02-05', 8);
insert into animal values ('Cooper' , 123456011, 'Bird'    ,
    'yellow', 'female', '2016-04-05', 2);

insert into consult values ('Striker', 123456001, '2018-01-01',
    'Nausea, fever and a black mark on the neck. Cancer', 'Black
    Mark.'                    , 'Cirurgy'     , 'No sun light.'
                               , 123456001, 123456022, 35);
insert into consult values ('Jackson', 123456022, '2017-02-01',
    'Nausea, fever and a black mark on the neck. Cancer', 'Black
    Mark.'                    , 'Cirurgy'     , 'No sun light.'
                               , 123456002, 123456004, 20);
insert into consult values ('Bob'    , 123456003, '2017-03-01',
    'Nausea, fever and a black mark on the neck. Cancer', 'Black
    Mark.'                    , 'Cirurgy'     , 'No sun light.'
                               , 123456003, 123456004, 32);
insert into consult values ('Thunder', 123456001, '2017-04-01',
    'Overweight.'                                          ,
    'Obesity'                        , 'Urine tests.', 'Lower
    caloric consumption and take pills.', 123456001, 123456005,
    47);
insert into consult values ('Sparky' , 123456002, '2017-05-01',
    'Overweight.'                                          ,
    'Obese'                          , 'Urine tests.', 'Lower
    caloric consumption and take pills.', 123456002, 123456022,
    25);
insert into consult values ('Sparky' , 123456002, '2018-06-01',
    'Overweight.'                                          ,
    'Obese'                          , 'Urine tests.', 'Lower
    caloric consumption and take pills.', 123456002, 123456022,
    35);
insert into consult values ('Max'    , 123456006, '2017-06-08',
    'Overweight.'                                          ,
    'Obesity'                        , 'Urine tests.', 'Lower
    caloric consumption and take pills.', 123456003, 123456005,
    40);
insert into consult values ('Max'    , 123456006, '2017-06-15',
    'Overweight.'                                          ,
    'Obesity'                        , 'Urine tests.', 'Lower
    caloric consumption and take pills.', 123456003, 123456005,
    35);
```

```
insert into consult values ('Max'    , 123456006, '2018-06-01',
   'Overweight.'                                 ,
   'Obesity'                        , 'Urine tests.', 'Lower
   caloric consumption and take pills.', 123456003, 123456005,
   25);
insert into consult values ('Cookie' , 123456011, '2017-07-01',
   'Dificulties on flying. Borken wing.'          , 'Blood
   analysis and radiography.', 'No touch.'   , 'Take pills.'
                              , 123456011, 123456022, 1.38);
insert into consult values ('Rolls'  , 123456022, '2017-08-01',
   'Dificulties on flying. Borken wing.'          , 'Blood
   analysis and radiography.', 'No touch.'   , 'Take pills.'
                              , 123456022, 123456006, 1.11);
insert into consult values ('Jim'    , 123456033, '2017-09-01',
   'Dificulties on flying. Borken wing.'          , 'Blood
   analysis and radiography.', 'No touch.'   , 'Take pills.'
                              , 123456033, 123456006, 1.42);
insert into consult values ('Cooper' , 123456011, '2017-10-01',
   'Dificulties on flying. Borken wing.'          , 'Blood
   analysis and radiography.', 'No touch.'   , 'No take pills.'
                             , 123456011, 123456006, 1.52);
insert into consult values ('Cooper' , 123456011, '2017-10-09',
   'Dificulties on flying. Borken wing.'          , 'Blood
   analysis'                 , 'No touch.'   , 'No take pills.'
                             , 123456011, 123456006, 1.52);

insert into participation values ('Striker', 123456001,
   '2018-01-01', 123456007);
insert into participation values ('Jackson', 123456022,
   '2017-02-01', 123456007);
insert into participation values ('Bob'    , 123456003,
   '2017-03-01', 123456022);
insert into participation values ('Thunder', 123456001,
   '2017-04-01', 123456008);
insert into participation values ('Sparky' , 123456002,
   '2017-05-01', 123456008);
insert into participation values ('Sparky' , 123456002,
   '2018-06-01', 123456008);
insert into participation values ('Max'    , 123456006,
   '2018-06-01', 123456022);
insert into participation values ('Max'    , 123456006,
   '2017-06-08', 123456022);
insert into participation values ('Max'    , 123456006,
   '2017-06-15', 123456022);
insert into participation values ('Cookie' , 123456011,
   '2017-07-01', 123456009);
insert into participation values ('Rolls'  , 123456022,
   '2017-08-01', 123456009);
insert into participation values ('Jim'    , 123456033,
   '2017-09-01', 123456009);
insert into participation values ('Cooper' , 123456011,
```

```
    '2017-10-01', 123456022);
insert into participation values ('Cooper' , 123456011,
    '2017-10-09', 123456022);

insert into diagnosis_code values ('AAHA-01', 'Cancer');
insert into diagnosis_code values ('AAHA-02', 'Overweight');
insert into diagnosis_code values ('AAHA-03', 'Broken limb');
insert into diagnosis_code values ('AAHA-04', 'kidney failure');

insert into consult_diagnosis values ('AAHA-01', 'Striker',
    123456001, '2018-01-01');
insert into consult_diagnosis values ('AAHA-01', 'Jackson',
    123456022, '2017-02-01');
insert into consult_diagnosis values ('AAHA-02', 'Bob'    ,
    123456003, '2017-03-01');
insert into consult_diagnosis values ('AAHA-02', 'Thunder',
    123456001, '2017-04-01');
insert into consult_diagnosis values ('AAHA-02', 'Sparky' ,
    123456002, '2017-05-01');
insert into consult_diagnosis values ('AAHA-02', 'Sparky' ,
    123456002, '2018-06-01');
insert into consult_diagnosis values ('AAHA-03', 'Max'    ,
    123456006, '2018-06-01');
insert into consult_diagnosis values ('AAHA-03', 'Max'    ,
    123456006, '2017-06-08');
insert into consult_diagnosis values ('AAHA-03', 'Max'    ,
    123456006, '2017-06-15');
insert into consult_diagnosis values ('AAHA-02', 'Cookie' ,
    123456011, '2017-07-01');
insert into consult_diagnosis values ('AAHA-03', 'Rolls'  ,
    123456022, '2017-08-01');
insert into consult_diagnosis values ('AAHA-02', 'Jim'    ,
    123456033, '2017-09-01');
insert into consult_diagnosis values ('AAHA-04', 'Cooper' ,
    123456011, '2017-10-01');
insert into consult_diagnosis values ('AAHA-04', 'Cooper' ,
    123456011, '2017-10-09');

insert into medication values ('med1'  , 'Lab-1'  , '200 mg');
insert into medication values ('med1'  , 'Lab-1'  , '300 mg');
insert into medication values ('med1'  , 'Lab-11' , '300 mg');
insert into medication values ('med11' , 'Lab-11' , '200 mg');
insert into medication values ('med111', 'Lab-111', '200 mg');
insert into medication values ('med2'  , 'Lab-2'  , '500 mg');
insert into medication values ('med22' , 'Lab-22' , '1000 mg');
insert into medication values ('med3'  , 'Lab-3'  , '1 g');

insert into prescription values ('AAHA-01', 'Striker',
    123456001, '2018-01-01', 'med1' , 'Lab-1' , '200 mg' , '1
    dose after dinnher. 24 hours between doses. 1 month of
    treatment. Take 200 mg');
```

```
insert into prescription values ('AAHA-01', 'Striker',
   123456001, '2018-01-01', 'med1' , 'Lab-1' , '300 mg' , '1
   dose after dinnher. 24 hours between doses. 1 month of
   treatment. Take 200 mg');
insert into prescription values ('AAHA-01', 'Striker',
   123456001, '2018-01-01', 'med1' , 'Lab-11' , '300 mg' , '1
   dose after dinnher. 24 hours between doses. 1 month of
   treatment. Take 200 mg');
insert into prescription values ('AAHA-01', 'Jackson',
   123456022, '2017-02-01', 'med11', 'Lab-11', '200 mg' , '1
   dose after dinnher. 24 hours between doses. 1 month of
   treatment. Take 200 mg');
insert into prescription values ('AAHA-02', 'Bob'    ,
   123456003, '2017-03-01', 'med1' , 'Lab-1' , '200 mg' , '1
   dose after dinnher. 24 hours between doses. 1 month of
   treatment. Take 200 mg');
insert into prescription values ('AAHA-02', 'Thunder',
   123456001, '2017-04-01', 'med1' , 'Lab-1' , '200 mg' , '1
   dose after lunch. 24 hours between doses. 1 month of
   treatment. Take 300 mg.');
insert into prescription values ('AAHA-02', 'Sparky' ,
   123456002, '2017-05-01', 'med2' , 'Lab-2' , '500 mg' , '1
   dose after lunch. 24 hours between doses. 1 month of
   treatment. Take 300 mg.');
insert into prescription values ('AAHA-03', 'Max'    ,
   123456006, '2018-06-01', 'med22', 'Lab-22', '1000 mg', '1
   dose after lunch. 24 hours between doses. 1 month of
   treatment. Take 300 mg.');
insert into prescription values ('AAHA-03', 'Max'    ,
   123456006, '2018-06-01', 'med2', 'Lab-2', '500 mg', '1 dose
   after lunch. 24 hours between doses. 1 month of treatment.
   Take 300 mg.');
insert into prescription values ('AAHA-03', 'Max'    ,
   123456006, '2018-06-01', 'med3', 'Lab-3', '1 g', '1 dose
   after lunch. 24 hours between doses. 1 month of treatment.
   Take 300 mg.');
insert into prescription values ('AAHA-02', 'Cookie' ,
   123456011, '2017-07-01', 'med22', 'Lab-22', '1000 mg', '2
   doses after dinner. 24 hours between doses. 1 month of
   treatment. Take 500 mg');
insert into prescription values ('AAHA-03', 'Rolls'  ,
   123456022, '2017-08-01', 'med3' , 'Lab-3' , '1 g'    , '2
   doses after dinner. 24 hours between doses. 1 month of
   treatment. Take 500 mg');
insert into prescription values ('AAHA-02', 'Jim'    ,
   123456033, '2017-09-01', 'med3' , 'Lab-3' , '1 g'    , '2
   doses after dinner. 24 hours between doses. 1 month of
   treatment. Take 500 mg');

insert into indicator values ('Nitrites'            , 100   ,
   'miligrams'                                       , 'This
```

```
   reference is the maximum for safety reasons.');
insert into indicator values ('Neurotrophils'       , 3000   ,
   'miligrams'                                       , 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('Lymphocytes'         , 300000,
   '% of the white blood cells present in the sample', 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('Monocytes'           , 4000   ,
   'miligrams'                                       , 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('creatine level'      , 200    ,
   'miligrams'                                       , 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('VO2 maximum'         , 50     ,
   'miligrams'                                       , 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('Ferritin'            , 1000   ,
   'miligrams'                                       , 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('Oxygen concentration', 2      ,
   'percentage'                                      , 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('Heartbeats'          , 10     ,
   'Bpm'                                             , 'This
   reference is the maximum for safety reasons.');
insert into indicator values ('Body fats'           , 100    ,
   'kilograms'                                       , 'This
   reference is the maximum for safety reasons.');

insert into procedures values ('Striker', 123456001,
   '2018-01-01', 11, 'Cirurgy');
insert into procedures values ('Jackson', 123456022,
   '2017-02-01', 12, 'Cirurgy');
insert into procedures values ('Bob'    , 123456003,
   '2017-03-01', 13, 'Cirurgy');
insert into procedures values ('Thunder', 123456001,
   '2017-04-01', 21, 'Urine');
insert into procedures values ('Max'    , 123456006,
   '2018-06-01', 22, 'Urine');
insert into procedures values ('Max'    , 123456006,
   '2017-06-08', 23, 'Urine');
insert into procedures values ('Max'    , 123456006,
   '2017-06-08', 24, 'Urine');
insert into procedures values ('Max'    , 123456006,
   '2017-06-15', 25, 'Urine');
insert into procedures values ('Max'    , 123456006,
   '2017-06-15', 26, 'Urine');
insert into procedures values ('Max'    , 123456006,
   '2017-06-15', 27, 'Urine');
insert into procedures values ('Cookie' , 123456011,
   '2017-07-01', 41, 'Blood analysis and radiography');
```

```
insert into procedures values ('Rolls'  , 123456022,
   '2017-08-01', 42, 'Blood analysis and radiography');
insert into procedures values ('Jim'    , 123456033,
   '2017-09-01', 43, 'Blood analysis and radiography');
insert into procedures values ('Cooper' , 123456011,
   '2017-10-01', 44, 'Blood analysis and radiography');
insert into procedures values ('Cooper' , 123456011,
   '2017-10-09', 51, 'Blood analysis');

insert into performed values ('Striker', 123456001,
   '2018-01-01', 11, 123456007);
insert into performed values ('Jackson', 123456022,
   '2017-02-01', 12, 123456007);
insert into performed values ('Bob'    , 123456003,
   '2017-03-01', 13, 123456022);
insert into performed values ('Thunder', 123456001,
   '2017-04-01', 21, 123456008);
insert into performed values ('Max'    , 123456006,
   '2018-06-01', 22, 123456022);
insert into performed values ('Max'    , 123456006,
   '2017-06-08', 23, 123456022);
insert into performed values ('Max'    , 123456006,
   '2017-06-08', 24, 123456022);
insert into performed values ('Max'    , 123456006,
   '2017-06-15', 25, 123456022);
insert into performed values ('Max'    , 123456006,
   '2017-06-15', 26, 123456022);
insert into performed values ('Max'    , 123456006,
   '2017-06-15', 27, 123456022);
insert into performed values ('Cookie' , 123456011,
   '2017-07-01', 41, 123456009);
insert into performed values ('Rolls'  , 123456022,
   '2017-08-01', 42, 123456009);
insert into performed values ('Jim'    , 123456033,
   '2017-09-01', 43, 123456009);
insert into performed values ('Cooper' , 123456011,
   '2017-10-01', 44, 123456022);

insert into radiography values ('Cookie', 123456011,
   '2017-07-01', 41, 'INSERT PATH');
insert into radiography values ('Rolls' , 123456022,
   '2017-08-01', 42, 'INSERT PATH');
insert into radiography values ('Jim'   , 123456033,
   '2017-09-01', 43, 'INSERT PATH');
insert into radiography values ('Cooper', 123456011,
   '2017-10-01', 44, 'INSERT PATH');

insert into test_procedure values ('Thunder', 123456001,
   '2017-04-01', 21, 'Urine');
insert into test_procedure values ('Max'    , 123456006,
   '2018-06-01', 22, 'Urine');
```

```
insert into test_procedure values ('Max'    , 123456006,
   '2017-06-08', 23, 'Urine');
insert into test_procedure values ('Max'    , 123456006,
   '2017-06-08', 24, 'Urine');
insert into test_procedure values ('Max'    , 123456006,
   '2017-06-15', 25, 'Urine');
insert into test_procedure values ('Max'    , 123456006,
   '2017-06-15', 26, 'Urine');
insert into test_procedure values ('Max'    , 123456006,
   '2017-06-15', 27, 'Urine');
insert into test_procedure values ('Cookie' , 123456011,
   '2017-07-01', 41, 'Blood');
insert into test_procedure values ('Rolls'  , 123456022,
   '2017-08-01', 42, 'Blood');
insert into test_procedure values ('Jim'    , 123456033,
   '2017-09-01', 43, 'Blood');
insert into test_procedure values ('Cooper' , 123456011,
   '2017-10-01', 44, 'Blood');
insert into test_procedure values ('Cooper' , 123456011,
   '2017-10-09', 51, 'Blood');

insert into produced_indicator values ('Thunder', 123456001,
   '2017-04-01', 21, 'Nitrites'      , 30);
insert into produced_indicator values ('Max'    , 123456006,
   '2018-06-01', 22, 'Nitrites'      , 50);
insert into produced_indicator values ('Max'    , 123456006,
   '2017-06-08', 23, 'Nitrites'      , 50);
insert into produced_indicator values ('Max'    , 123456006,
   '2017-06-08', 24, 'Nitrites'      , 50);
insert into produced_indicator values ('Max'    , 123456006,
   '2017-06-15', 25, 'Nitrites'      , 50);
insert into produced_indicator values ('Max'    , 123456006,
   '2017-06-15', 26, 'Nitrites'      , 50);
insert into produced_indicator values ('Max'    , 123456006,
   '2017-06-15', 27, 'Nitrites'      , 50);
insert into produced_indicator values ('Cookie' , 123456011,
   '2017-07-01', 41, 'Monocytes'     , 100000);
insert into produced_indicator values ('Rolls'  , 123456022,
   '2017-08-01', 42, 'Neurotrophils' , 200000);
insert into produced_indicator values ('Jim'    , 123456033,
   '2017-09-01', 43, 'Lymphocytes'   , 2500000);
insert into produced_indicator values ('Cooper' , 123456011,
   '2017-10-01', 44, 'Lymphocytes'   , 50000);
insert into produced_indicator values ('Cooper' , 123456011,
   '2017-10-09', 51, 'creatine level', 5);
```

Código 2: Instrução SQL para o preenchimento da base de dados

3 Queries

3.1 Query 1

Na figura 1 pode se ver os atributos em destaque para esta *query*, em que cada linha corresponde a uma consulta e os seus dados mais importantes respetivamente.

VAT	name	name	species_name	VAT_vet	name
123456011	Andrew	Cookie	Bird	123456022	John Smith
123456011	Andrew	Cooper	Bird	123456006	Jessica
123456011	Andrew	Cooper	Bird	123456006	Jessica
123456033	Ray	Jim	Bird	123456006	Jessica
123456022	John Smith	Rolls	Bird	123456006	Jessica
123456002	Charles	Sparky	Boxer	123456022	John Smith
123456002	Charles	Sparky	Boxer	123456022	John Smith
123456003	James	Bob	Bulldog	123456004	Peter
123456022	John Smith	Jackson	Bulldog	123456004	Peter
123456001	Frank	Striker	Bulldog	123456022	John Smith
123456001	Frank	Thunder	Cat	123456005	Jennifer
123456006	Jessica	Max	Pit Bull	123456005	Jennifer
123456006	Jessica	Max	Pit Bull	123456005	Jennifer
123456006	Jessica	Max	Pit Bull	123456005	Jennifer

Figura 1: Tabela com os atributos relevantes

```
SELECT DISTINCT c.name AS animal_name, owner.name AS
   owner_name, species_name AS species, age
FROM person owner, person vet, consult c, animal a
WHERE c.name = a.name
AND owner.VAT = c.VAT_owner
AND vet.VAT = c.VAT_vet
AND vet.name = 'John Smith';
```

Código 3: SQL query 1

```
mysql> source query1.sql
+-------------+------------+---------+------+
| animal_name | owner_name | species | age  |
+-------------+------------+---------+------+
| Cookie      | Andrew     | Bird    |   13 |
| Sparky      | Charles    | Boxer   |    4 |
| Striker     | Frank      | Bulldog |   10 |
+-------------+------------+---------+------+
3 rows in set (0,00 sec)
```

Figura 2: Resultado da *query* 1

3.2 Query 2

```
MySQL [ist426527]> select name, reference_value, units from indicator;
+----------------------+-----------------+------------+
| name                 | reference_value | units      |
+----------------------+-----------------+------------+
| Body fats            |          100.00 | kilograms  |
| creatine level       |          200.00 | miligrams  |
| Ferritin             |         1000.00 | miligrams  |
| Heartbeats           |           10.00 | Bpm        |
| Lymphocytes          |       300000.00 | percentage |
| Monocytes            |         4000.00 | miligrams  |
| Neurotrophils        |         3000.00 | miligrams  |
| Nitrites             |          100.00 | miligrams  |
| Oxygen concentration |            2.00 | percentage |
| VO2 maximum          |           50.00 | miligrams  |
+----------------------+-----------------+------------+
```

Figura 3: Tabela `indicator` com os atributos relevantes

```sql
SELECT name, reference_value
FROM indicator
WHERE reference_value > 100
ORDER BY reference_value DESC;
```

Código 4: SQL query 2

```
MySQL [ist426527]> source query2.sql
+-----------------+-----------------+
| name            | reference_value |
+-----------------+-----------------+
| Lymphocytes     |       300000.00 |
| Monocytes       |         4000.00 |
| Neurotrophils   |         3000.00 |
| Ferritin        |         1000.00 |
| creatine level  |          200.00 |
+-----------------+-----------------+
```

Figura 4: Resultado da *query* 2

3.3 Query 3

```
mysql> SELECT name, VAT_owner, date_timestamp, o, weight FROM consult;
+---------+-----------+---------------------+---------------------------------+--------+
| name    | VAT_owner | date_timestamp      | o                               | weight |
+---------+-----------+---------------------+---------------------------------+--------+
| Bob     | 123456003 | 2017-03-01 00:00:00 | Black Mark.                     |  32.00 |
| Cookie  | 123456011 | 2017-07-01 00:00:00 | Blood analysis and radiography. |   1.38 |
| Cooper  | 123456011 | 2017-10-01 00:00:00 | Blood analysis and radiography. |   1.52 |
| Cooper  | 123456011 | 2017-10-09 00:00:00 | Blood analysis                  |   1.52 |
| Jackson | 123456022 | 2017-02-01 00:00:00 | Black Mark.                     |  20.00 |
| Jim     | 123456033 | 2017-09-01 00:00:00 | Blood analysis and radiography. |   1.42 |
| Max     | 123456006 | 2017-06-08 00:00:00 | Obesity                         |  40.00 |
| Max     | 123456006 | 2017-06-15 00:00:00 | Obesity                         |  35.00 |
| Max     | 123456006 | 2018-06-01 00:00:00 | Obesity                         |  25.00 |
| Rolls   | 123456022 | 2017-08-01 00:00:00 | Blood analysis and radiography. |   1.11 |
| Sparky  | 123456002 | 2017-05-01 00:00:00 | Obese                           |  25.00 |
| Sparky  | 123456002 | 2018-06-01 00:00:00 | Obese                           |  35.00 |
| Striker | 123456001 | 2018-01-01 00:00:00 | Black Mark.                     |  35.00 |
| Thunder | 123456001 | 2017-04-01 00:00:00 | Obesity                         |  47.00 |
+---------+-----------+---------------------+---------------------------------+--------+
14 rows in set (0,01 sec)
```

Figura 5: Tabela `consult` com os atributos relevantes

```
SELECT c.name AS animal_name, p.name AS owner_name,
   species_name AS species, age, weight
FROM (consult c natural join animal AS a) INNER JOIN person p
   ON (a.VAT = p.VAT)
WHERE weight > 30
AND (o LIKE '%obesity%' OR o LIKE '%obese%')
GROUP BY a.name, a.VAT, date_timestamp
HAVING date_timestamp IN (SELECT max(date_timestamp)
                          FROM (consult c1 NATURAL JOIN animal
                             AS a1) INNER JOIN person p1 ON
                             (a1.VAT = p1.VAT)
                          WHERE a1.name = a.name
                          AND a1.VAT = a.VAT
                          GROUP BY a.name, a.VAT);
```

Código 5: SQL query 3

```
mysql> source query3.sql
+-------------+------------+---------+------+--------+
| animal_name | owner_name | species | age  | weight |
+-------------+------------+---------+------+--------+
| Sparky      | Charles    | Boxer   |    4 |  35.00 |
| Thunder     | Frank      | Cat     |    2 |  47.00 |
+-------------+------------+---------+------+--------+
2 rows in set (0,00 sec)
```

Figura 6: Resultado da *query* 3

Como forma de verificar a correta implementação da *query* 3, na figura 5 deve-se destacar o animal "`Max`". Este animal nas 2 primeiras consultas tinha um peso superior a 30 quilogramas,

contudo na útlima já tinha um peso inferior a 30 quilogramas, daí não aparecer como resultado da *query 3*.

3.4 Query 4

```
MySQL [ist426527]> select p.name as client, a.name as animal
    -> from person p natural join client c
    -> left outer join animal a on a.VAT = c.VAT;
+------------+---------+
| client     | animal  |
+------------+---------+
| Frank      | Striker |
| Frank      | Thunder |
| Charles    | Sparky  |
| James      | Bob     |
| John Smith | NULL    |
| Jessica    | Max     |
| Anne       | NULL    |
| Andrew     | Cookie  |
| Andrew     | Cooper  |
| Peter      | Jackson |
| Peter      | Rolls   |
| Ray        | Jim     |
+------------+---------+
```

Figura 7: Tabela com os atributos relevantes

```sql
SELECT *
FROM person NATURAL JOIN client
WHERE vat NOT IN (SELECT vat
                  FROM animal)
```

Código 6: SQL query 4

```
MySQL [ist426527]> source query4.sql
+-----------+------------+-----------------+--------------+-------------+
| VAT       | name       | address_street  | address_city | address_zip |
+-----------+------------+-----------------+--------------+-------------+
| 123456004 | John Smith | Central Avenue  | Seattle      | 2200-123    |
| 123456009 | Anne       | Happy Boulevard | Chicago      | 1111-125    |
+-----------+------------+-----------------+--------------+-------------+
```

Figura 8: Resultado da *query* 4

3.5 Query 5

code	diagnosis_name	animal_name	date_timestamp	name_med	lab	dosage
AAHA-01	Cancer	Jackson	2018-02-01 00:00:00	med11	Lab-11	200 mg
AAHA-01	Cancer	Striker	2018-01-01 00:00:00	med1	Lab-1	200 mg
AAHA-01	Cancer	Striker	2018-01-01 00:00:00	med1	Lab-1	300 mg
AAHA-01	Cancer	Striker	2018-01-01 00:00:00	med1	Lab-11	300 mg
AAHA-02	Overweight	Bob	2018-03-01 00:00:00	med1	Lab-1	200 mg
AAHA-02	Overweight	Cookie	2018-07-01 00:00:00	med22	Lab-22	1000 mg
AAHA-02	Overweight	Jim	2018-09-01 00:00:00	med3	Lab-3	1 g
AAHA-02	Overweight	Sparky	2018-05-01 00:00:00	med2	Lab-2	500 mg
AAHA-02	Overweight	Sparky	2018-06-01 00:00:00	NULL	NULL	NULL
AAHA-02	Overweight	Thunder	2018-04-01 00:00:00	med1	Lab-1	200 mg
AAHA-03	Broken limb	Max	2018-06-01 00:00:00	med2	Lab-2	500 mg
AAHA-03	Broken limb	Max	2018-06-01 00:00:00	med22	Lab-22	1000 mg
AAHA-03	Broken limb	Max	2018-06-01 00:00:00	med3	Lab-3	1 g
AAHA-03	Broken limb	Max	2018-06-08 00:00:00	NULL	NULL	NULL
AAHA-03	Broken limb	Max	2018-06-15 00:00:00	NULL	NULL	NULL
AAHA-03	Broken limb	Rolls	2018-08-01 00:00:00	med3	Lab-3	1 g
AAHA-04	kidney failure	Cooper	2018-10-01 00:00:00	NULL	NULL	NULL
AAHA-04	kidney failure	Cooper	2018-10-08 00:00:00	NULL	NULL	NULL

Figura 9: Tabela com os atributos relevantes

```
SELECT diagnosis_code.name AS possible_diagnosis,
   COUNT(DISTINCT name_med) AS
   number_of_distinct_medication_name
FROM (consult_diagnosis NATURAL LEFT OUTER JOIN prescription)
   INNER JOIN diagnosis_code ON (consult_diagnosis.code =
   diagnosis_code.code)
GROUP BY diagnosis_code.name
ORDER BY COUNT(DISTINCT name_med);
```

Código 7: SQL query 5

```
mysql> source query5.sql;
+--------------------+-----------------------------------------+
| possible_diagnosis | number_of_distinct_medication_name |
+--------------------+-----------------------------------------+
| kidney failure     |                                       0 |
| Cancer             |                                       2 |
| Broken limb        |                                       3 |
| Overweight         |                                       4 |
+--------------------+-----------------------------------------+
4 rows in set (0,02 sec)
```

Figura 10: Resultado da *query* 5

De notar aqui que só são verificados os nomes distintos de `medication`, ou seja, mais nenhum atributo desta tabela está a ser tido em conta, desta forma, nem o laboratório de concessão nem a dosagem são considerados.

3.6 Query 6

Desta vez optou-se por não incluir no relatório imagens dos atributos relevantes para visualizar o futuro resultado da *query*, pois seria necessário um número significativo de imagens para fazê-lo. Segue-se então, de forma mais preceptível, à enumeração das inserções relevantes para esta *query*, presentes no ficheiro `populating_veterinary_hospital.sql` (que populou as tabelas criadas):

1. Um total de 11 inserções na tabela `consult` no ano 2017;
2. Um total de 11 inserções na tabela `participation` no ano 2017;
3. Um total de 13 inserções na tabela `procedures` no ano 2017;
4. Um total de 11 inserções na tabela `consult_diagnosis` no ano 2017;
5. Um total de 7 inserções na tabela `prescription` no ano 2017.

```
SELECT
COUNT(distinct pa.name, pa.VAT_owner, pa.date_timestamp,
   pa.VAT_assistant)/COUNT(distinct c.name, c.VAT_owner,
   c.date_timestamp) AS participations_average,
COUNT(distinct p.name, p.VAT_owner, p.date_timestamp,
   p.num)/COUNT(distinct c.name, c.VAT_owner, c.date_timestamp)
   AS procedures__average,
COUNT(distinct cd.code, cd.name, cd.VAT_owner,
   cd.date_timestamp)/COUNT(distinct c.name, c.VAT_owner,
   c.date_timestamp) AS consult_diagnosis_average,
COUNT(distinct pres.code, pres.name, pres.VAT_owner,
   pres.date_timestamp, pres.name_med, pres.lab,
   pres.dosage)/COUNT(distinct c.name, c.VAT_owner,
   c.date_timestamp) AS prescriptions_average
FROM
consult c NATURAL LEFT OUTER JOIN participation pa
NATURAL LEFT OUTER JOIN procedures p NATURAL LEFT OUTER JOIN
consult_diagnosis cd NATURAL LEFT OUTER JOIN prescription pres
WHERE
year(c.date_timestamp) = 2017;
```

Código 8: SQL query 6

```
MySQL [ist426527]> source query6.sql
+-----------------------+--------------------+--------------------------+----------------------+
| participations_average | procedures__average | consult_diagnosis_average | prescriptions_average |
+-----------------------+--------------------+--------------------------+----------------------+
|                 1.0000 |             1.1818 |                   1.0000 |               0.6364 |
+-----------------------+--------------------+--------------------------+----------------------+
```

Figura 11: Resultado da query 6

NOTA: visto que a versão de mysql utilizada não permite guardar valores em variávies auxiliares, não se econtrou nenhuma alternativa que evitasse ser feita a mesma contagem na tabela `consult` para o cálculo de cada média.

3.7 Query 7

sub_species	diagnosis_name	frequency_of_disease
Boxer	Overweight	2
Bulldog	Cancer	2
Bulldog	Overweight	1
Pit Bull	Broken limb	3

Figura 12: Tabela com os atributos relevantes

```
SELECT a.species_name AS sub_species, dc.name AS
   most_common_disease
FROM ((((consult_diagnosis cd NATURAL JOIN animal AS a) INNER
   JOIN species s ON (a.species_name = s.name)) INNER JOIN
   generalization_species gs ON (s.name = gs.name1)) INNER JOIN
   diagnosis_code dc ON (cd.code = dc.code))
WHERE gs.name2 = 'dog'
GROUP BY a.species_name, dc.name
HAVING COUNT(*) >= ALL(SELECT count(*)
                       FROM ((((consult_diagnosis cd1 NATURAL
                          JOIN animal AS a1) INNER JOIN species
                          s1 ON (a1.species_name = s1.name))
                          INNER JOIN generalization_species gs1
                          ON (s1.name = gs1.name1)) INNER JOIN
                          diagnosis_code dc1 ON (cd1.code =
                          dc1.code))
                           WHERE a.species_name =
                              a1.species_name
                           GROUP BY dc1.name);
```

Código 9: SQL query 7

```
mysql> source query7.sql;
+--------------+----------------------+
| sub_species  | most_common_disease  |
+--------------+----------------------+
| Boxer        | Overweight           |
| Bulldog      | Cancer               |
| Pit Bull     | Broken limb          |
+--------------+----------------------+
3 rows in set (0,02 sec)
```

Figura 13: Resultado da *query* 7

Para esta questão, no caso de múltiplas doenças aparecerem o mesmo número de vezes para a mesma sub-espécie de `dog`, então a *query* 7 irá colocar todas essas doenças no resultado (este caso foi testado e verificado com sucesso).

3.8 Query 8

(A)

client_name	vat
Frank	123456001
Charles	123456002
James	123456003
John Smith	123456004
Jessica	123456006
Anne	123456009
Andrew	123456011
Peter	123456022
Ray	123456033

(B)

veterinary_name	vat
John Smith	123456004
Jennifer	123456005
Jessica	123456006
Peter	123456022

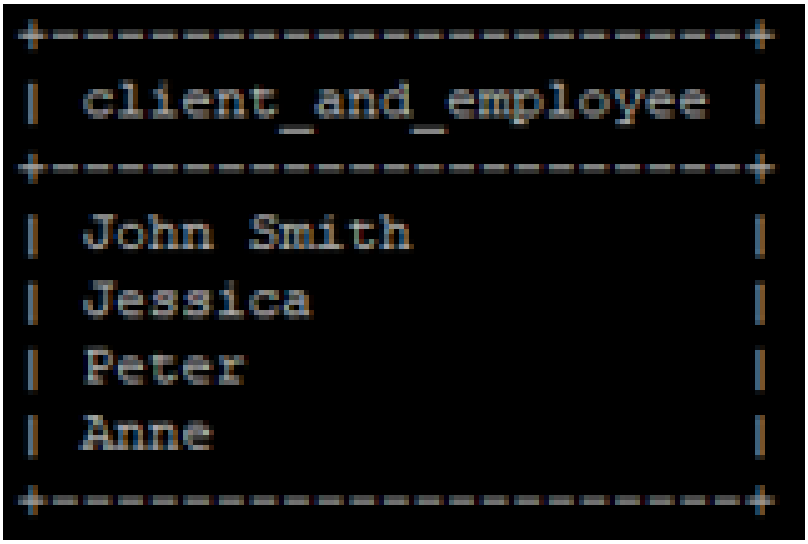
(C)

assistant_name	vat
Caroline	123456007
Bernard	123456008
Anne	123456009
Peter	123456022

Figura 14: Tabela com os atributos relevantes: (A) Clientes do hospital; (B) Veterinários do hospital; (C) Assistentes do hospital

```
(SELECT person.name FROM person NATURAL JOIN client NATURAL
   JOIN veterinary)
UNION
(SELECT person.name FROM person NATURAL JOIN client NATURAL
   JOIN assistant)
```

Código 10: SQL query 8



client_and_employee
John Smith
Jessica
Peter
Anne

Figura 15: Resultado da *query* 8

3.9 Query 9

VAT	client_name	animal_name	species_name
123456011	Andrew	Cookie	Bird
123456011	Andrew	Cooper	Bird
123456033	Ray	Jim	Bird
123456022	Peter	Rolls	Bird
123456002	Charles	Sparky	Boxer
123456003	James	Bob	Bulldog
123456022	Peter	Jackson	Bulldog
123456001	Frank	Striker	Bulldog
123456001	Frank	Thunder	Cat
123456006	Jessica	Max	Pit Bull

Figura 16: Tabela com os atributos relevantes

```
SELECT distinct p.name AS client_name, p.address_street AS
   street, p.address_city AS city, p.address_zip AS zip
FROM person p INNER JOIN animal a  on (p.VAT = a.VAT)
WHERE p.VAT NOT IN (SELECT p.VAT
                    FROM person p, animal a
                    WHERE p.VAT = a.VAT
                    AND (species_name NOT LIKE '%bird%'));
```

Código 11: SQL query 9

```
mysql> source query9.sql;
+-------------+------------------+---------------+----------+
| client_name | street           | city          | zip      |
+-------------+------------------+---------------+----------+
| Andrew      | Roses Boulevard  | Chicago       | 1111-123 |
| Ray         | Second Boulevard | San Francisco | 3300-123 |
+-------------+------------------+---------------+----------+
2 rows in set (0,02 sec)
```

Figura 17: Resultado da *query* 9

A destacar na figura 16 o cliente "`Peter`"que tem tanto um animal da espécie "`Bulldog`"como da espécie "`Bird`", pelo que não aparece nos resultados da figura 17.

4 Índices para a Base de Dados

4.1 Índice para a Query 1

Com o auxílio do comando `explain`, é possível verificar quantas linhas a *query* 1 inspeciona para mostrar o resultado final, sendo que é possível ver na figura 18 que são 10.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	a	ALL	PRIMARY	NULL	NULL	NULL	10	Using temporary
1	SIMPLE	c	ref	PRIMARY,VAT_vet	PRIMARY	257	ist425330.a.name	1	
1	SIMPLE	vet	eq_ref	PRIMARY	PRIMARY	4	ist425330.c.VAT_vet	1	Using where
1	SIMPLE	owner	eq_ref	PRIMARY	PRIMARY	4	ist425330.c.VAT_owner	1	

Figura 18: Resultado do comando *explain* para a *query* 1 sem o uso de indexes

Com a criação de índices é possível acelerar bastante o acesso aos dados desejados. Com esse intuito e tendo em conta a *query* 1, foi então criado um índice para o atributo `name` na tabela `person`, pois nesta *query* é necessário aceder ao nome não só para juntar as tabelas como para procurar pelo veterinário "`John Smith`".

Não foi necessário criar qualquer outro indíce, pois o sistema em si já cria automaticamente índices sobre os atributos que são *primary key*.

```
CREATE INDEX name_index ON
person(name);
```

Código 12: Instrução SQL para a criação do índice para *query* 1

Na figura 19 é possível confirmar a correta criação do índice. É possível também verificar que este índice foi criado do tipo *B+-tree*, visto que se trata de uma base de dados de dimensão reduzida. Caso estivéssemos a trabalhar com bastantes mais dados, o sistema talvez já criasse índices do tipo *Hash*. Este último tipo de índices tem a vantagem de não ser necessário ordenar os dados, mas pelo contrário requer que exista uma boa função de dispersão de forma a que haja poucas colisões e a complexidade de acesso seja o mais próximo possível de O(1) para que não se tenha que fazer uma grande procura nos *"buckets"*.

```
mysql> show indexes from person;
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table  | Non_unique | Key_name   | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| person |          0 | PRIMARY    |            1 | VAT         | A         |           2 |     NULL | NULL   |      | BTREE      |         |               |
| person |          1 | name_index |            1 | name        | A         |          13 |     NULL | NULL   | YES  | BTREE      |         |               |
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
2 rows in set (0,00 sec)
```

Figura 19: Confirmação da criação do index na tabela *person*

Teoricamente para melhorar ao máximo a performance da *query 1* com uma base de dados de grande dimensão beneficiaríamos de indíces do tipo *Hash*. Como o objetivo nesta *query* é apenas verificar igualdades, com o uso de uma boa função de dispersão, este tipo de índices permitir-nos-ia encontrar o valor pretendido "imediatamente".

Contudo, como se trata de uma dimensão muito reduzida e apesar de termos de percorrer a árvore verifica-se, pela figura 23, que isto não é crítico e com uma organização dos índices em *B+-tree* já se consegue reduzir a inspeção de linhas para mostrar o resultado final ao mínimo (1 linha).

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	vet	ref	PRIMARY,name_index	name_index	258	const	1	Using where; Using index; Using temporary
1	SIMPLE	c	ref	PRIMARY,VAT_vet	VAT_vet	5	ist425330.vet.VAT	1	Using where; Using index
1	SIMPLE	owner	eq_ref	PRIMARY	PRIMARY	4	ist425330.c.VAT_owner	1	
1	SIMPLE	a	ref	PRIMARY	PRIMARY	257	ist425330.c.name	1	

Figura 20: Resultado do comando *explain* para a *query 2* com o uso de indexes

Por último, elimina-se o índice para não influnciar as questões posteriores.

```
DROP INDEX name_index ON
person;
```

Código 13: Instrução SQL para a eliminação do *index* para *query 1*

4.2 Índice para a Query 2

Analogamente ao índice criado para a *query* 1, primeiramente, com o comando `explain`, verificou-se que esta *query* sem qualquer índice implementado inspeciona 13 linhas.

```
+----+-------------+-----------+------+---------------+------+---------+------+------+----------------------------+
| id | select_type | table     | type | possible_keys | key  | key_len | ref  | rows | Extra                      |
+----+-------------+-----------+------+---------------+------+---------+------+------+----------------------------+
|  1 | SIMPLE      | indicator | ALL  | NULL          | NULL | NULL    | NULL |   13 | Using where; Using filesort |
+----+-------------+-----------+------+---------------+------+---------+------+------+----------------------------+
```

Figura 21: Resultado do comando *explain* para a *query 2* sem o uso de indexes

De forma a acelarar a procura na tabela em causa para esta questão criou-se um índice para o atributo `reference_value` na tabela `indicator`. Como esta *query* apenas avaliava um atributo este índice é suficiente para acelerar o acesso aos dados para o mínimo possível.

```
CREATE INDEX reference_value_index ON
indicator(reference_value);
```

Código 14: Instrução SQL para a criação do index para *query* 2

Na figura 22 confirma-se a criação correta do índice. A par do que aconteceu na questão anterior o índice criado é do tipo *B+-tree*, pelos mesmos motivos.

```
mysql> show index from indicator;
+-----------+------------+-------------------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table     | Non_unique | Key_name                | Seq_in_index | Column_name     | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-----------+------------+-------------------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| indicator |          0 | PRIMARY                 |            1 | name            | A         |           2 |     NULL | NULL   |      | BTREE      |         |               |
| indicator |          1 | reference_value_index   |            1 | reference_value | A         |          13 |     NULL | NULL   | YES  | BTREE      |         |               |
+-----------+------------+-------------------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
2 rows in set (0,01 sec)
```

Figura 22: Confirmação da criação do `index` na tabela *indicator*

Para esta *query*, teoricamente, para melhorar ao máximo a perfomance (mesmo que para uma base de dados real, ou seja, grande dimensão) o tipo de índice mais adequado seria mesmo *B+-tree*, pois para este tipo de índice ordenam-se os dados e os caminhos da raíz às folhas são sempre do mesmo tamanho (ou seja, metade está em cada lado). Como neste caso o objetivo é encontrar um `reference_value` a partir do qual é superior a 100, estes dados irão estar todos do mesmo lado da árvore pelo que bastará procurar pelo primeiro valor e seguir a árvore para encontrar todos os valores. Contrariamente, no tipo *Hash* sempre que um novo valor aparecer terá sempre de se calcular a função de dispersão e ir procurar ao *"bucket"* correspondente.

```
+----+-------------+-----------+-------+-----------------------+-----------------------+---------+------+------+--------------------------+
| id | select_type | table     | type  | possible_keys         | key                   | key_len | ref  | rows | Extra                    |
+----+-------------+-----------+-------+-----------------------+-----------------------+---------+------+------+--------------------------+
|  1 | SIMPLE      | indicator | range | reference_value_index | reference_value_index | 10      | NULL |    5 | Using where; Using index |
+----+-------------+-----------+-------+-----------------------+-----------------------+---------+------+------+--------------------------+
```

Figura 23: Resultado do comando *explain* para a *query 2* com o uso de indexes

Por último, mais uma vez, elimina-se o índice para não influnciar as questões posteriores.

```
DROP INDEX reference_value_index ON
indicator;
```

Código 15: Instrução SQL para a criação do index para *query* 2

5 Alterações na Base de Dados

5.1 Alteração 1

```
MySQL [ist426527]> select * from client natural join person where name = 'John Smith';
+-----------+------------+----------------+--------------+-------------+
| VAT       | name       | address_street | address_city | address_zip |
+-----------+------------+----------------+--------------+-------------+
| 123456004 | John Smith | Central Avenue | Seattle      | 2200-123    |
+-----------+------------+----------------+--------------+-------------+
```

Figura 24: Cliente de relevo antes da alteração

```
UPDATE person
SET address_street = 'White Avenue', address_city = 'New York'
WHERE name = 'John Smith';
```

Código 16: update/deletion 1

```
MySQL [ist426527]> select * from client natural join person where name = 'John Smith';
+-----------+------------+----------------+--------------+-------------+
| VAT       | name       | address_street | address_city | address_zip |
+-----------+------------+----------------+--------------+-------------+
| 123456004 | John Smith | White Avenue   | New York     | 2200-123    |
+-----------+------------+----------------+--------------+-------------+
```

Figura 25: Cliente de relevo após a alteração

5.2 Alteração 2

```
MySQL [ist426527]> select type, units, reference_value from produced_indicator
    -> natural join test_procedure inner join indicator
    -> on produced_indicator.indicator_name = indicator.name;
+-------+------------+-----------------+
| type  | units      | reference_value |
+-------+------------+-----------------+
| Blood | miligrams  |          200.00 |
| Blood | percentage |       300000.00 |
| Blood | percentage |       300000.00 |
| Blood | miligrams  |         4000.00 |
| Blood | miligrams  |         3000.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
+-------+------------+-----------------+
```

Figura 26: Atributos de relevo antes da alteração

```
UPDATE ((produced_indicator NATURAL JOIN test_procedure) INNER
   JOIN indicator ON produced_indicator.indicator_name =
   indicator.name)
SET reference_value = reference_value*1.1
WHERE type = 'Blood'
```

```
AND units = 'miligrams';
```

Código 17: update/deletion 2

```
MySQL [ist426527]> select type, units, reference_value from produced_indicator
    -> natural join test_procedure inner join indicator
    -> on produced_indicator.indicator_name = indicator.name;
+-------+------------+-----------------+
| type  | units      | reference_value |
+-------+------------+-----------------+
| Blood | miligrams  |          220.00 |
| Blood | percentage |       300000.00 |
| Blood | percentage |       300000.00 |
| Blood | miligrams  |         4400.00 |
| Blood | miligrams  |         3300.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
| Urine | miligrams  |          100.00 |
+-------+------------+-----------------+
```

Figura 27: Atributos de relevo após a alteração

5.3 Alteração 3

Para esta alteração, há que notar na utilização da instrução `on delete cascade` na criação das tabelas de forma a que a ação de eliminar uma linha da tabela associada a uma *primary key* associe diretamente as *foreign key* correspondentes e estas linhas sejam também eliminadas.

Desta forma, consegue-se que quando o cliente "`John Smith`"for eliminado toda a sua informação seja eliminada, nomeadamente de todos os animais do qual ele é o dono e todas as consultas (incluindo `procedures`, `diagnosis` e `prescriptions` no qual ele esteja envolvido). Contudo, é ainda de salientar que "`John Smith`"só deverá ser retirado da base de dados onde ele se comporta como cliente, pelo que se irá manter como `veterinary` e `assistant`, se for o caso.

A figura 28 mostra apenas um exemplo de como estavam povoadas as tabelas `consult` e `participation` antes desta instrução, sendo que o `VAT` de "`John Smith`"é o "`123456022`". Assim sendo, o que está assinalado a vermelho deverá ser eliminado pois faz parte da informação referente ao cliente e o que está a verde deverá manter-se pois faz parte da informação que não faz parte do cliente (ou seja, é onde o "`John Smith`"é `veterinary` ou `assistant`).

```
mysql> SELECT consult.name as animal_name, consult.VAT_owner, VAT_client, VAT_vet, VAT_assistant
    -> FROM consult NATURAL JOIN participation;
+-------------+-----------+------------+-----------+---------------+
| animal_name | VAT_owner | VAT_client | VAT_vet   | VAT_assistant |
+-------------+-----------+------------+-----------+---------------+
| Jackson     | 123456022 |  123456002 | 123456004 |     123456007 |
| Striker     | 123456001 |  123456001 | 123456022 |     123456007 |
| Sparky      | 123456002 |  123456002 | 123456022 |     123456008 |
| Sparky      | 123456002 |  123456002 | 123456022 |     123456008 |
| Thunder     | 123456001 |  123456001 | 123456005 |     123456008 |
| Cookie      | 123456011 |  123456011 | 123456022 |     123456009 |
| Jim         | 123456033 |  123456033 | 123456006 |     123456009 |
| Rolls       | 123456022 |  123456022 | 123456006 |     123456009 |
| Bob         | 123456003 |  123456003 | 123456004 |     123456022 |
| Cooper      | 123456011 |  123456011 | 123456006 |     123456022 |
| Cooper      | 123456011 |  123456011 | 123456006 |     123456022 |
| Max         | 123456006 |  123456003 | 123456005 |     123456022 |
| Max         | 123456006 |  123456003 | 123456005 |     123456022 |
| Max         | 123456006 |  123456003 | 123456005 |     123456022 |
+-------------+-----------+------------+-----------+---------------+
14 rows in set (0,00 sec)
```

Figura 28: Tabela exemplificativa com os atributos em destaque

É relevante referir que a instrução apresentada cobre todos os casos onde este cliente está envolvido, sendo que a figura 28 é apenas ilustrativa do seu correto funcionamento.

```
DELETE client
FROM client
INNER JOIN person ON person.VAT = client.VAT
WHERE name = 'John Smith';
```

Código 18: Instrução SQL para a eliminação do cliente "`John Smith`"

Com a figura 29 confirma-se que o "`John Smith`"se manteve como "`veterinary`"e "`assistant`"(assinalado a azul), tendo sido retirado de todas as tabelas onde é cliente e tudo o que daí descende.

```
mysql> source update_deletion_instruction3.sql
Query OK, 1 row affected (0,00 sec)

mysql> SELECT consult.name as animal_name, consult.VAT_owner, VAT_client, VAT_vet, VAT_assistant
    -> FROM consult NATURAL JOIN participation;
+-------------+-----------+------------+-----------+---------------+
| animal_name | VAT_owner | VAT_client | VAT_vet   | VAT_assistant |
+-------------+-----------+------------+-----------+---------------+
| Striker     | 123456001 |  123456001 | 123456022 |     123456007 |
| Sparky      | 123456002 |  123456002 | 123456022 |     123456008 |
| Sparky      | 123456002 |  123456002 | 123456022 |     123456008 |
| Thunder     | 123456001 |  123456001 | 123456005 |     123456008 |
| Cookie      | 123456011 |  123456011 | 123456022 |     123456009 |
| Jim         | 123456033 |  123456033 | 123456006 |     123456009 |
| Bob         | 123456003 |  123456003 | 123456004 |     123456022 |
| Cooper      | 123456011 |  123456011 | 123456006 |     123456022 |
| Cooper      | 123456011 |  123456011 | 123456006 |     123456022 |
| Max         | 123456006 |  123456003 | 123456005 |     123456022 |
| Max         | 123456006 |  123456003 | 123456005 |     123456022 |
| Max         | 123456006 |  123456003 | 123456005 |     123456022 |
+-------------+-----------+------------+-----------+---------------+
12 rows in set (0,01 sec)
```

Figura 29: Resultado da tabela exemplificativa com os atributos em destaque após a execução da instrução em cima apresentada

5.4 Alteração 4

Como se pode observar na figura 30, antes de implementar a alteração pedida, a tabela `diagnosis_code` não contém nenhuma entrada para a condição "`end-stage renal disease`" e da figura 31 vê-se que a única linha que contém todas as entradas a cumprir os critérios é a terceira apresentada.

```
MySQL [ist426527]> select * from diagnosis_code;
+----------+-----------------+
| code     | name            |
+----------+-----------------+
| AAHA-01  | Cancer          |
| AAHA-02  | Overweight      |
| AAHA-03  | Broken limb     |
| AAHA-04  | kidney failure  |
+----------+-----------------+
```

Figura 30: Tabela "`diagnosis_code`"antes da quarta alteração

```
MySQL [ist426527]> select cd.code, cd.name, type, indicator_name, value from
    -> consult_diagnosis cd NATURAL JOIN test_procedure NATURAL JOIN produced_indicator;
+---------+---------+-------+----------------+------------+
| code    | name    | type  | indicator_name | value      |
+---------+---------+-------+----------------+------------+
| AAHA-02 | Cookie  | Blood | Monocytes      |  100000.00 |
| AAHA-04 | Cooper  | Blood | Lymphocytes    |   50000.00 |
| AAHA-04 | Cooper  | Blood | creatine level |       5.00 |
| AAHA-02 | Jim     | Blood | Lymphocytes    | 2500000.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Rolls   | Blood | Neurotrophils  |  200000.00 |
| AAHA-02 | Thunder | Urine | Nitrites       |      30.00 |
+---------+---------+-------+----------------+------------+
```

Figura 31: Pacientes de relevo antes da quarta alteração

Depois implementar as alterações obtém-se os resultados, agora apresentados, nas figuras 32 e 33.

```
INSERT INTO diagnosis_code VALUES ('AAHA-15', 'end-stage renal
    disease');

UPDATE
consult_diagnosis cd NATURAL JOIN test_procedure NATURAL JOIN
    produced_indicator, diagnosis_code dc
SET cd.code = dc.code, cd.name = cd.name, cd.VAT_owner =
    cd.VAT_owner, cd.date_timestamp = cd.date_timestamp
WHERE
cd.code IN (SELECT code FROM diagnosis_code WHERE name =
    'kidney failure') AND
dc.name = 'end-stage renal disease' AND
type = 'Blood' AND indicator_name = 'creatine level' AND value
    > 1.0;
```

Código 19: update/deletion 4

```
MySQL [ist426527]> select * from diagnosis_code;
+---------+--------------------------+
| code    | name                     |
+---------+--------------------------+
| AAHA-01 | Cancer                   |
| AAHA-02 | Overweight               |
| AAHA-03 | Broken limb              |
| AAHA-04 | kidney failure           |
| AAHA-15 | end-stage renal disease  |
+---------+--------------------------+
```

Figura 32: Tabela "`diagnosis_code`"depois da quarta alteração

```
MySQL [ist426527]> select cd.code, cd.name, type, indicator_name, value from
    -> consult_diagnosis cd NATURAL JOIN test_procedure NATURAL JOIN produced_indicator;
+---------+---------+-------+----------------+------------+
| code    | name    | type  | indicator_name | value      |
+---------+---------+-------+----------------+------------+
| AAHA-02 | Cookie  | Blood | Monocytes      |  100000.00 |
| AAHA-04 | Cooper  | Blood | Lymphocytes    |   50000.00 |
| AAHA-15 | Cooper  | Blood | creatine level |       5.00 |
| AAHA-02 | Jim     | Blood | Lymphocytes    | 2500000.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Max     | Urine | Nitrites       |      50.00 |
| AAHA-03 | Rolls   | Blood | Neurotrophils  |  200000.00 |
| AAHA-02 | Thunder | Urine | Nitrites       |      30.00 |
+---------+---------+-------+----------------+------------+
```

Figura 33: Pacientes de relevo depois da quarta alteração

Agora já se verifica a nova condição na tablea `diagnosis_code`, e vê-se também que a entrada da coluna `code` correspondente define agora a condição do elemento que cumpria todos os critérios.

6 Criação de *views*

6.1 *View 1*

```
CREATE VIEW dim_date AS
SELECT DISTINCT date_timestamp, year(date_timestamp) AS year,
   month(date_timestamp) AS month, day(date_timestamp) AS day
   FROM consult
```

Código 20: view 1

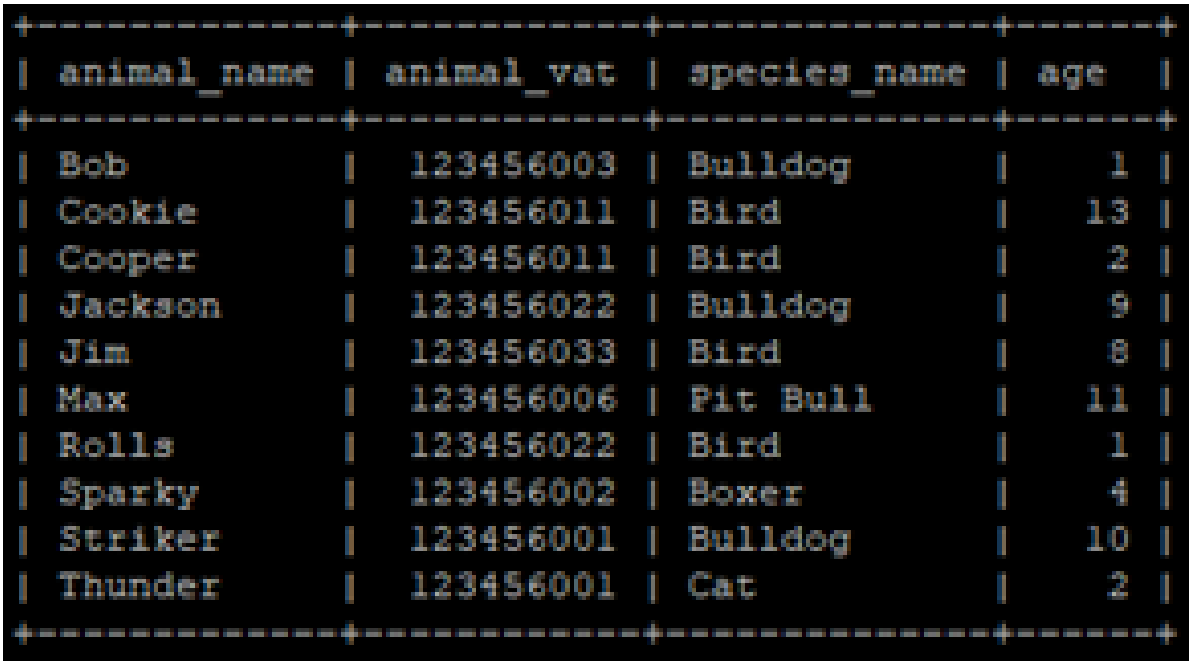
date_timestamp	year	month	day
2018-01-01 00:00:00	2018	1	1
2017-04-01 00:00:00	2017	4	1
2017-02-01 00:00:00	2017	2	1
2017-05-01 00:00:00	2017	5	1
2018-06-01 00:00:00	2018	6	1
2017-03-01 00:00:00	2017	3	1
2017-06-08 00:00:00	2017	6	8
2017-06-15 00:00:00	2017	6	15
2017-07-01 00:00:00	2017	7	1
2017-10-01 00:00:00	2017	10	1
2017-10-09 00:00:00	2017	10	9
2017-08-01 00:00:00	2017	8	1
2017-09-01 00:00:00	2017	9	1

Figura 34: Visualização da view 1

6.2 *View 2*

```
CREATE VIEW dim_animal AS
SELECT name AS animal_name, vat AS animal_vat, species_name,
   age FROM animal;
```

Código 21: view 2



animal_name	animal_vat	species_name	age
Bob	123456003	Bulldog	1
Cookie	123456011	Bird	13
Cooper	123456011	Bird	2
Jackson	123456022	Bulldog	9
Jim	123456033	Bird	8
Max	123456006	Pit Bull	11
Rolls	123456022	Bird	1
Sparky	123456002	Boxer	4
Striker	123456001	Bulldog	10
Thunder	123456001	Cat	2

Figura 35: Visualização da view 2

6.3 *View 3*

Para a criação desta *view*, existem agora algumas especificidades que têm de ser tidas em consideração, nomeadamente o facto de haver entradas em `consult` onde não haja correspondência na tabela `procedures`, pelo que se deve apresentar o respetivo valor de `num_procedures` igual a zero nestas situações.

```
SELECT animal_name AS name, animal_vat AS vat,
   dd.date_timestamp AS timestamp, COUNT(distinct num) AS
   num_procedures, COUNT(distinct name_med, dosage, lab) AS
   num_medications
FROM (dim_animal da NATURAL JOIN animal) NATURAL JOIN (dim_date
   dd NATURAL JOIN consult) NATURAL LEFT OUTER JOIN procedures
   NATURAL LEFT OUTER JOIN prescription
GROUP BY timestamp, name
ORDER BY name;
```

Código 22: view 3

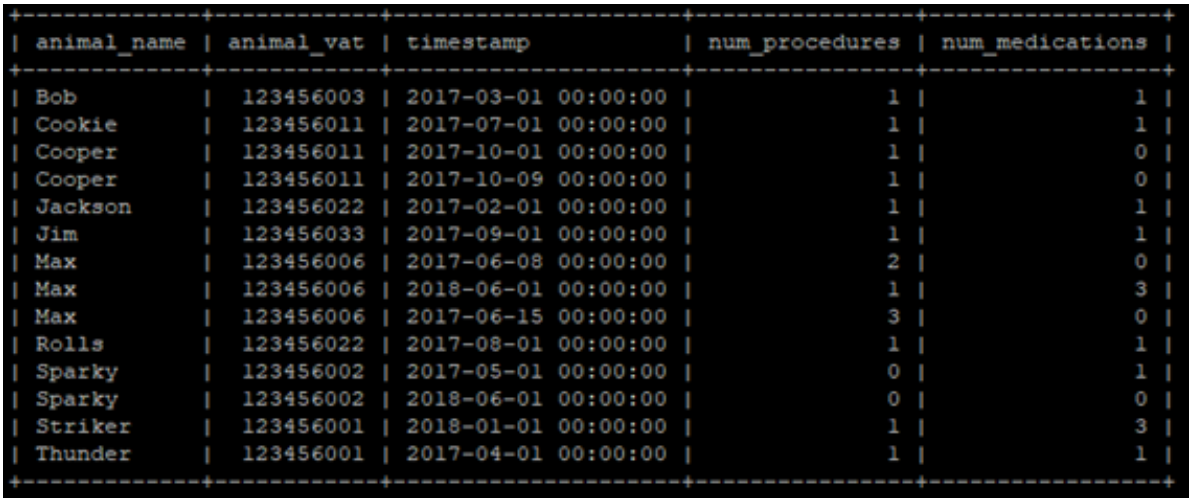
animal_name	vat_owner	date_timestamp	num
Bob	123456003	2017-03-01 00:00:00	13
Cookie	123456011	2017-07-01 00:00:00	41
Cooper	123456011	2017-10-01 00:00:00	44
Cooper	123456011	2017-10-09 00:00:00	51
Jackson	123456022	2017-02-01 00:00:00	12
Jim	123456033	2017-09-01 00:00:00	43
Max	123456006	2017-06-08 00:00:00	23
Max	123456006	2017-06-08 00:00:00	24
Max	123456006	2017-06-15 00:00:00	25
Max	123456006	2017-06-15 00:00:00	26
Max	123456006	2017-06-15 00:00:00	27
Max	123456006	2018-06-01 00:00:00	22
Rolls	123456022	2017-08-01 00:00:00	42
Striker	123456001	2018-01-01 00:00:00	11
Thunder	123456001	2017-04-01 00:00:00	21

Figura 36: Tabela dos `procedures` realizados (sem a descrição)

Por último, tem que se ter em conta o facto de não existirem prescrições de medicamentos nas consultas ou, poderem haver prescrições com o nome medicamento e laboratório igual, mas com dosagem diferente (ou dosagem igual, mas com laboratório diferente). Assim, a instrução `DINSTINCT` é necessária. É importante referir que não podem existir medicamentos com o mesmo nome, dose e laboratório porque estes atributos são primary key.

code	animal_name	vat_owner	date_timestamp	name_med	lab	dosage
AAHA-02	Bob	123456003	2017-03-01 00:00:00	med1	Lab-1	200 mg
AAHA-02	Cookie	123456011	2017-07-01 00:00:00	med22	Lab-22	1000 mg
AAHA-01	Jackson	123456022	2017-02-01 00:00:00	med11	Lab-11	200 mg
AAHA-02	Jim	123456033	2017-09-01 00:00:00	med3	Lab-3	1 g
AAHA-03	Max	123456006	2018-06-01 00:00:00	med2	Lab-2	500 mg
AAHA-03	Max	123456006	2018-06-01 00:00:00	med3	Lab-3	1 g
AAHA-03	Max	123456006	2018-06-01 00:00:00	med22	Lab-22	1000 mg
AAHA-03	Rolls	123456022	2017-08-01 00:00:00	med3	Lab-3	1 g
AAHA-02	Sparky	123456002	2017-05-01 00:00:00	med2	Lab-2	500 mg
AAHA-01	Striker	123456001	2018-01-01 00:00:00	med1	Lab-1	200 mg
AAHA-01	Striker	123456001	2018-01-01 00:00:00	med1	Lab-1	300 mg
AAHA-01	Striker	123456001	2018-01-01 00:00:00	med1	Lab-11	300 mg
AAHA-02	Thunder	123456001	2017-04-01 00:00:00	med1	Lab-1	200 mg

Figura 37: Tabela das `prescriptions` realizadas (sem o regime)



animal_name	animal_vat	timestamp	num_procedures	num_medications
Bob	123456003	2017-03-01 00:00:00	1	1
Cookie	123456011	2017-07-01 00:00:00	1	1
Cooper	123456011	2017-10-01 00:00:00	1	0
Cooper	123456011	2017-10-09 00:00:00	1	0
Jackson	123456022	2017-02-01 00:00:00	1	1
Jim	123456033	2017-09-01 00:00:00	1	1
Max	123456006	2017-06-08 00:00:00	2	0
Max	123456006	2018-06-01 00:00:00	1	3
Max	123456006	2017-06-15 00:00:00	3	0
Rolls	123456022	2017-08-01 00:00:00	1	1
Sparky	123456002	2017-05-01 00:00:00	0	1
Sparky	123456002	2018-06-01 00:00:00	0	0
Striker	123456001	2018-01-01 00:00:00	1	3
Thunder	123456001	2017-04-01 00:00:00	1	1

Figura 38: Visualização da *view 3*

Os resultados são de acordo com o pretendido pois, como se pode observar, o animal "`Sparky`", não recebeu qualquer `procedure`. Foram prescritos três medicamentos (com o mesmo `name_med` mas com `dosage` ou `lab` diferentes) para o animal "`Striker`". Por último, o animal `"Max"` não recebeu qualquer medicamento em duas das consultas.