Instituto Superior Técnico

Mestrado em Engenharia Eletrotécnica e de Computadores

Sistemas de Informação e Bases de Dados

2018/2019 1º Semestre

Assignment 2 - Database Implementation

Grupo: 4

84037 – Eduardo Costa

84038 – Eduardo Melo

84087 – João Sebastião

Docente: Bruno Martins

9 de Novembro de 2018

1 – Database Creation

The following code represents the SQL instructions to create the database.

```
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 medical_procedure;
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 phone_number;
drop table if exists assistant;
drop table if exists veterinary;
drop table if exists client;
drop table if exists person;


create table person(
                VAT int CHECK( VAT > 0),
                name varchar(255) not null,
                address_street varchar(255) not null,
                address_city varchar(255) not null,
                address_zip varchar(255) not null,
                primary key(VAT));


create table client(
                VAT int CHECK( VAT > 0),
                primary key(VAT),
                foreign key(VAT) references person(VAT) ON DELETE CASCADE);
```

```
create table veterinary(
                VAT int CHECK( VAT > 0),
                specialization varchar(255),
                bio varchar(255),
                primary key(VAT),
                foreign key(VAT) references person(VAT) ON DELETE CASCADE);
create table assistant(
                VAT int CHECK( VAT > 0),
                primary key(VAT),
                foreign key(VAT) references person(VAT) ON DELETE CASCADE);


create table phone_number(
                VAT int CHECK( VAT > 0),
                phone varchar(255),
                primary key(VAT, phone),
                foreign key(VAT) references person(VAT) ON DELETE CASCADE);

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) ON DELETE CASCADE,
                foreign key(name2) references species(name) ON DELETE CASCADE);


create table animal(
                name varchar(255),
                VAT int CHECK( VAT > 0),
                species_name varchar(255),
                colour varchar(255),
                gender varchar(255),
                birth_year year,
                age int,
                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 int CHECK( VAT_owner > 0),
		date_timestamp date,
		s varchar(255),
		o varchar(255),
		a varchar(255),
		p varchar(255),
		VAT_client int CHECK( VAT_client > 0),
		VAT_vet int CHECK( VAT_vet > 0),
		weight real CHECK( weight > 0),
		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),
		foreign key(VAT_vet) references veterinary(VAT));


create table participation (
		name varchar(255),
		VAT_owner int CHECK( VAT_owner > 0),
		date_timestamp date,
		VAT_assistant int CHECK( VAT_assistant > 0),
		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) ON DELETE CASCADE);


create table diagnosis_code(
		code int CHECK( code > 0),
		name varchar(255),
		primary key(code));


create table consult_diagnosis(
		code int CHECK( code > 0),
		name varchar(255),
		VAT_owner int CHECK( VAT_owner > 0),
		date_timestamp date,
		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) ON DELETE CASCADE);
```

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


create table prescription(
		code int CHECK( code > 0),
		name varchar(255),
		VAT_owner int CHECK( VAT_owner > 0),
		date_timestamp date,
		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) ON DELETE CASCADE);


create table indicator(
		name varchar(255),
		reference_value real CHECK( reference_value > 0),
		units varchar(255),
		description varchar(255),
		primary key(name));


create table medical_procedure(
		name varchar(255),
		VAT_owner int CHECK( VAT_owner > 0),
		date_timestamp date,
		num int,
		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 int CHECK( VAT_owner > 0),
		date_timestamp date,
		num int,
		VAT_assistant int CHECK( VAT_assistant > 0),
		primary key(name, VAT_owner, date_timestamp, num, VAT_assistant),
		foreign key(name, VAT_owner, date_timestamp, num) references
medical_procedure(name, VAT_owner, date_timestamp, num) ON DELETE CASCADE,
		foreign key(VAT_assistant) references assistant(VAT) ON DELETE CASCADE);


create table radiography(
		name varchar(255),
		VAT_owner int CHECK( VAT_owner > 0),
		date_timestamp date,
		num int,
		rad_file varchar(255),
		primary key(name, VAT_owner, date_timestamp, num),
		foreign key(name, VAT_owner, date_timestamp, num) references
medical_procedure(name, VAT_owner, date_timestamp, num) ON DELETE CASCADE);


create table test_procedure(
		name varchar(255),
		VAT_owner int CHECK( VAT_owner > 0),
		date_timestamp date,
		num int,
		test_type varchar(255) CHECK (test_type = 'blood' OR test_type = 'urine'),
		primary key(name, VAT_owner, date_timestamp, num),
		foreign key(name, VAT_owner, date_timestamp, num) references
medical_procedure(name, VAT_owner, date_timestamp, num) ON DELETE CASCADE);


create table produced_indicator(
		name varchar(255),
		VAT_owner int CHECK( VAT_owner > 0),
		date_timestamp date,
		num int,
		indicator_name varchar(255),
		indicator_value real CHECK( indicator_value > 0),
		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) ON DELETE
CASCADE);
```

2 – Populate the Database

The following code represents the SQL instructions to populate the tables in the database.

```
INSERT INTO person(VAT, name, address_street, address_city, address_zip)
VALUES
        (123456780, 'John Smith', 'Avenida da Liberdade', 'Lisbon', 13824),
        (123456781, 'John Smith', 'Avenida de Berlim', 'Lisbon', 13927),
        (123456782, 'Thomas Edison', 'Avenida das Forças Armadas', 'Lisbon', 13525),
        (123456783, 'Erwin Schrodinger', 'Avenida de Roma', 'Lisbon', 13734),
        (123456784, 'Sir Isaac Newton', 'Avenida do Brasil', 'Lisbon', 13626),
        (123456785, 'Albert Einstein', 'Rua Garret', 'Lisbon', 13439),
        (123456786, 'Marie Curie', 'Avenida dos Aliados', 'Porto', 43201),
        (123456787, 'Charles Darwin', 'Avenida da Boavista', 'Porto', 43224),
        (123456788, 'James Maxwell', 'Rua das Flores', 'Porto', 42985),
        (123456789, 'Archimedes', 'Rua Santa Catarina', 'Porto', 43253),
        (123456790, 'John Smith', 'Avenida Luis Silva', 'Porto', 42155);


INSERT INTO client(VAT)
VALUES
        (123456780),
        (123456782),
        (123456783),
        (123456784),
        (123456785),
        (123456788),
        (123456789),
        (123456790);


INSERT INTO veterinary(VAT, specialization, bio)
VALUES
        (123456780, 'Ophthalmology', 'Someone that enjoys their personal freedom' ),
        (123456781, 'Nutrition', 'Occasionally visits Germany'),
        (123456786, 'Radiology', 'Always supportive towards my allies'),
        (123456790, 'Orthopedy', 'Taught by the great Luis Silva');


INSERT INTO assistant(VAT)
VALUES
        (123456782),
        (123456787);
```

```
INSERT INTO phone_number(VAT, phone)
VALUES
        (123456780, 908765432),
        (123456781, 918765432),
        (123456782, 928765432),
        (123456783, 938765432),
        (123456784, 948765432),
        (123456785, 958765432),
        (123456786, 968765432),
        (123456787, 978765432),
        (123456788, 988765432),
        (123456789, 998765432);


INSERT INTO species(name, description)
VALUES
        ('Mammal', 'Hot blood animals with fur'),
        ('Canis Lupus', 'Includes all sub-species of wolf'),
        ('Canis Lupus arctos','The greatest wolf of them all'),
        ('Dog', 'Canis Lupus Familiaris - Mans best friend'),
        ('Chihuahua', 'Very small dog. Shakes, trembles and barks with every inch of its body.
Basically a demon trapped into a baby dog’s body'),
        ('Boxer','Very good boy. Leaves saliva everywhere'),
        ('Husky', 'The fluffiest. Very good boy'),
        ('Golden Retriever','The best boy of them all'),
        ('Fish', 'Swims very good'),
        ('Carassius auratus','The classic golden fish'),
        ('Felis', 'Small felines'),
        ('Felis Bieti','Chinese wild cat'),
        ('Felis Catus', 'Doesnt care about anything'),
        ('Bird', 'Likes to fly'),
        ('Cacatuidae','Very intelligent bird. Likes to party'),
        ('Melopsittacus Undulatus', 'Small bird. Can be very mean'),
        ('Agapornis Roseicollis', 'Very beautiful bird, sings very nice'),
        ('Reptile', 'Cold blood animals - do not confuse with fish'),
        ('Testudinidae', 'Very very slow'),
        ('Crocodylidae','Very very big reptile. He doesnt bite, he just wants to play'),
        ('Cheloniidae', 'Turtle that is still slow in land but swims very good');
```

```
INSERT INTO generalization_species(name1, name2)
VALUES
        ('Canis Lupus', 'Mammal'),
        ('Canis lupus arctos', 'Canis Lupus'),
        ('Dog', 'Canis Lupus'),
        ('Chihuahua', 'Dog'),
        ('Boxer', 'Dog'),
        ('Husky', 'Dog'),
        ('Golden Retriever', 'Dog'),
        ('Felis', 'Mammal'),
        ('Felis Catus','Felis'),
        ('Felis Bieti', 'Felis'),
        ('Carassius auratus','Fish'),
        ('Cacatuidae', 'Bird'),
        ('Melopsittacus Undulatus', 'Bird'),
        ('Agapornis Roseicollis', 'Bird'),
        ('Testudinidae','Reptile'),
        ('Crocodylidae','Reptile'),
        ('Cheloniidae','Reptile');



INSERT INTO animal(name, VAT, species_name, colour, gender, birth_year, age)
VALUES
        ('Paradox', 123456783, 'Felis Catus', 'black', 'male', 2013, year(current_timestamp) -
birth_year),
        ('Galileo', 123456784, 'Golden Retriever', 'golden', 'male', 2011,
year(current_timestamp) - birth_year),
        ('Galileia', 123456784, 'Golden Retriever', 'golden', 'female', 2011,
year(current_timestamp) - birth_year),
        ('Jacinto', 123456785, 'Bird', 'green', 'male', 2015, year(current_timestamp) -
birth_year),
        ('Descartes', 123456789, 'Bird', 'white', 'male', 2017, year(current_timestamp) -
birth_year),
        ('Atila', 123456784, 'Boxer', 'brown', 'male', 2014, year(current_timestamp) -
birth_year),
        ('Faraday', 123456789, 'Cheloniidae', 'green', 'female', 2006,
year(current_timestamp) - birth_year),
        ('Gragas', 123456780, 'Crocodylidae', 'green', 'male', 1970, year(current_timestamp)
- birth_year),
        ('Alex', 123456780, 'Boxer', 'brown', 'male', 2012, year(current_timestamp) -
birth_year),
        ('Jacinta', 123456785, 'Bird', 'red', 'female', 2015, year(current_timestamp) -
birth_year);
```

```
INSERT INTO consult(name, VAT_owner, date_timestamp, s, o, a, p, VAT_client, VAT_vet, weight)
VALUES
	('Gragas', 123456780, '2017-05-30', 'had teeth pain', 'found fish stuck between teeth', 'clean teeth', 'but dental floss', 123456780, 123456790, 1135.40),

	('Alex', 123456780, '2016-12-11', 'had teeth pain', 'found cavity', 'remove tooth', 'but dental floss', 123456780, 123456790, 35.25),

	('Paradox', 123456783, '2017-01-04', 'complained alot', 'swelled belly', 'kidney malfunction', 'get some meds', 123456783, 123456780, 3.89),

	('Galileo', 123456784, '2017-09-17', 'spots on the skin', 'fungal infection on the skin', 'ringworm', 'get some meds', 123456784, 123456780, 40.10),

	('Galileia', 123456784, '2017-09-17', 'complained alot', 'swelled belly', 'stomach failure', 'get some rest', 123456785, 123456781, 35.90),

	('Gragas', 123456780, '2016-05-30', 'routine check-up', 'found nothing noticeable', 'make blood analysis', 'rest', 123456780, 123456790, 1122.30),

	('Galileo', 123456784, '2017-09-18', 'more spots on the skin', 'fungal infection on the skin', 'ringworm', 'get ultra meds', 123456784, 123456780, 39.90),

	('Jacinto', 123456785, '2017-05-30', 'has beak pain', 'beak was normal but bird is a bit obese', 'stop being a crybird', 'go home', 123456782, 123456781, 0.38),

	('Descartes', 123456789, '2017-04-17', 'didnt dance to music', 'had obstructed ears', 'use cotton swab', 'buy cotton swab', 123456788, 123456786, 0.41),

	('Faraday', 123456789, '2018-01-22', 'didnt move', 'shows signs of obesity', 'ate a lot of junk food', 'Moderate and balanced food', 123456789, 123456786, 160.23),

	('Faraday', 123456789, '2018-02-22', 'still didnt move', 'still looks obese and stomach ache', 'still ate a lot of junk food', 'Moderate and balanced food and meds', 123456789, 123456786, 180.23),

	('Atila', 123456784, '2017-02-15', 'checkup consult', 'healthy but a little obese', 'obese', 'go run', 123456784, 123456780, 31.23),

	('Atila', 123456784, '2017-03-15', '2nd checkup consult', 'lost signs of obesity', 'healthy', 'keep diet', 123456784, 123456780, 29.23);
```

```
INSERT INTO participation(name, VAT_owner, date_timestamp, VAT_assistant)
VALUES
        ('Galileo', 123456784, '2017-09-17', 123456782),
        ('Descartes', 123456789, '2017-04-17', 123456787),
        ('Faraday', 123456789, '2018-01-22', 123456787),
        ('Gragas', 123456780, '2016-05-30', 123456787),
        ('Gragas', 123456780, '2016-05-30', 123456782),
        ('Gragas', 123456780, '2017-05-30', 123456787),
        ('Gragas', 123456780, '2017-05-30', 123456782),
        ('Atila', 123456784, '2017-02-15', 123456782);



INSERT INTO diagnosis_code(code, name)
VALUES
        (6281, 'Kidney Failure'),
        (5683, 'Stomach Failure'),
        (0076, 'Cancer'),
        (4409, 'Ear Infection'),
        (8320, 'Ringworm'),
        (1542, 'Inflamed Gum'),
        (1532, 'Teeth Pain');



INSERT INTO consult_diagnosis(code, name, VAT_owner, date_timestamp)
VALUES
        (6281, 'Paradox', 123456783, '2017-01-04'),
        (6281, 'Faraday', 123456789, '2018-02-22'),
        (4409, 'Atila', 123456784, '2017-03-15'),
        (8320, 'Galileo', 123456784, '2017-09-17'),
        (5683, 'Galileia', 123456784, '2017-09-17'),
        (8320, 'Galileo', 123456784, '2017-09-18'),
        (1542, 'Gragas', 123456780, '2017-05-30'),
        (1532, 'Gragas', 123456780, '2017-05-30'),
        (6281, 'Gragas', 123456780, '2016-05-30'),
        (1532, 'Alex', 123456780, '2016-12-11');



INSERT INTO medication(name, lab, dosage)
VALUES
        ('Amoxicillin', 'AMOCLAVAM', '80 to 90 mg per kg'),
        ('Imodium A-D', 'IMODIUM', '0.3 to 0.6 mL per pound'),
        ('Ultra Imodium A-D', 'IMODIUM', '1.3 to 1.6 mL per pound'),
        ('No Teeth Pain', 'TEETH LAB', '90 mg per kg'),
        ('Get Good Gum', 'BIG GREEN BOYS', '0.6 g per kg'),
        ('Gum Supplement', 'BIG GREEN BOYS', '0.3 g per kg');
```

```
INSERT INTO prescription(code, name, VAT_owner, date_timestamp, name_med, lab, dosage, regime)
VALUES
	(6281, 'Paradox', 123456783, '2017-01-04', 'Amoxicillin', 'AMOCLAVAM', '80 to 90 mg per kg', 'every day'),

	(6281, 'Faraday', 123456789, '2018-02-22', 'Amoxicillin', 'AMOCLAVAM', '80 to 90 mg per kg', 'every day'),

	(8320, 'Galileo', 123456784, '2017-09-17', 'Imodium A-D', 'IMODIUM', '0.3 to 0.6 mL per pound', 'two to three times a day'),

	(8320, 'Galileo', 123456784, '2017-09-18', 'Ultra Imodium A-D', 'IMODIUM', '1.3 to 1.6 mL per pound', 'two times a day'),

	(1532, 'Gragas', 123456780, '2017-05-30', 'No Teeth Pain', 'TEETH LAB', '90 mg per kg', 'fit the pills box into his favourite food'),

	(1542, 'Gragas', 123456780, '2017-05-30', 'Get Good Gum', 'BIG GREEN BOYS', '0.6 g per kg', 'fit the pills box into his favourite food'),

	(1542, 'Gragas', 123456780, '2017-05-30', 'Gum Supplement', 'BIG GREEN BOYS', '0.3 g per kg', 'spray on mouth after brushing teeth'),

	(1532, 'Alex', 123456780, '2016-12-11', 'No Teeth Pain', 'TEETH LAB', '90 mg per kg', 'three times per day');



INSERT INTO indicator(name, reference_value, units, description)
VALUES
	('Microalbumin', 310.00, 'milligrams' , 'urine protein'),
	('Glucose level', 110.00, 'milligrams', 'blood sugar'),
	('Cholesterol level', 1.20, 'grams', 'blood fat level'),
	('Acidosis', 150.00, 'milliliters', 'Blood acid level'),
	('Creatinine level', 1.00, 'milligrams', 'proportional to muscle');



INSERT INTO medical_procedure(name, VAT_owner, date_timestamp, num, description)
VALUES
	('Paradox', 123456783, '2017-01-04', 0, 'radiography exam'),
	('Paradox', 123456783, '2017-01-04', 1, 'blood test procedure'),
	('Paradox', 123456783, '2017-01-04', 2, 'urine test procedure'),
	('Gragas', 123456780, '2017-05-30', 0, 'radiography exam'),
	('Gragas', 123456780, '2016-05-30', 0, 'blood test procedure');
```

```
INSERT INTO performed(name, VAT_owner, date_timestamp, num, VAT_assistant)
VALUES
        ('Paradox', 123456783, '2017-01-04', 0, 123456782),
        ('Paradox', 123456783, '2017-01-04', 1, 123456787),
        ('Paradox', 123456783, '2017-01-04', 2, 123456787),
        ('Gragas', 123456780, '2017-05-30', 0, 123456787),
        ('Gragas', 123456780, '2016-05-30', 0, 123456782);


INSERT INTO radiography(name, VAT_owner, date_timestamp, num, rad_file)
VALUES
        ('Paradox', 123456783, '2017-01-04', 0,
'/Documents/Radiography_Exams/Paradox123456783'),
        ('Gragas', 123456780, '2017-05-30', 0,
'/Documents/Radiography_Exams/Gragas123456780');


INSERT INTO test_procedure(name, VAT_owner, date_timestamp, num, test_type)
VALUES
        ('Paradox', 123456783, '2017-01-04', 1, 'blood'),
        ('Paradox', 123456783, '2017-01-04', 2, 'urine'),
        ('Gragas', 123456780, '2016-05-30', 0, 'blood');


INSERT INTO produced_indicator(name, VAT_owner, date_timestamp, num,
indicator_name, indicator_value)
VALUES
        ('Paradox', 123456783, '2017-01-04', 1, 'Creatinine level', 2.3),
        ('Paradox', 123456783, '2017-01-04', 2, 'Microalbumin', 300.00),
        ('Gragas', 123456780, '2016-05-30', 0, 'Creatinine level', 0.9);
```

3 – Queries and Results

1 .

```
select distinct a.name as Animal_Name, p2.name as Owner_Name, species_name as
Species_Name, age as Animal_Age
from consult as c, animal as a, veterinary as v, client as owner, person as p1, person as p2
where v.VAT = c.VAT_vet
and v.VAT = p1.VAT
and p1.name = 'John Smith'
and c.name = a.name
and c.VAT_owner = a.VAT
and owner.VAT = a.VAT
and owner.VAT = p2.VAT;
```

```
MySQL [ist425337]> source Query1.sql;
+-------------+-------------------+------------------+------------+
| Animal_Name | Owner_Name        | Species_Name     | Animal_Age |
+-------------+-------------------+------------------+------------+
| Atila       | Sir Isaac Newton  | Boxer            |          4 |
| Galileo     | Sir Isaac Newton  | Golden Retriever |          7 |
| Paradox     | Erwin Schrodinger | Felis Catus      |          5 |
| Galileia    | Sir Isaac Newton  | Golden Retriever |          7 |
| Jacinto     | Albert Einstein   | Bird             |          3 |
| Alex        | John Smith        | Boxer            |          6 |
| Gragas      | John Smith        | Crocodylidae     |         48 |
+-------------+-------------------+------------------+------------+
7 rows in set (0.00 sec)

MySQL [ist425337]>
```

2.

```
select name as Indicator_Name, reference_value as Reference_Value
from indicator
where units = 'milligrams'
and reference_value > 100
group by reference_value DESC;
```

```
MySQL [ist425337]> source Query2.sql
+----------------+-----------------+
| Indicator_Name | Reference_Value |
+----------------+-----------------+
| Microalbumin   |             310 |
| Glucose level  |             110 |
+----------------+-----------------+
2 rows in set (0.00 sec)

MySQL [ist425337]>
```

3.

```
select distinct a.name as Animal_Name, p.name as Owner_Name, a.species_name as
Species_Name, a.age as Animal_Age
from animal as a, person as p, consult as c
where c.date_timestamp = (
      select max(date_timestamp)
      from consult as c2
      where c2.name = c.name
      and c.weight > 30)
and (LOCATE('obese', c.o) or LOCATE('obesity', c.o))
and a.name = c.name
and a.VAT = c.VAT_owner
and p.VAT = a.VAT;
```

```
MySQL [ist425337]> source Query3.sql
+-------------+------------+---------------+------------+
| Animal_Name | Owner_Name | Species_Name | Animal_Age |
+-------------+------------+---------------+------------+
| Faraday     | Archimedes | Cheloniidae  |         12 |
+-------------+------------+---------------+------------+
1 row in set (0.00 sec)

MySQL [ist425337]>
```

4.

```
select distinct p.name as Name, p.VAT as VAT, address_street as Street, address_city as
City, address_zip as ZIP
from animal as a, person as p, client as c
where c.VAT = p.VAT
and p.VAT not in(select VAT from animal);
```

```
MySQL [ist425337]> source Query4.sql
+--------------+-----------+----------------------------+--------+-------+
| Name         | VAT       | Street                     | City   | ZIP   |
+--------------+-----------+----------------------------+--------+-------+
| Thomas Edison | 123456782 | Avenida das Forças Armadas | Lisbon | 13525 |
| James Maxwell | 123456788 | Rua das Flores             | Porto  | 42985 |
| John Smith    | 123456790 | Avenida Luis silva         | Porto  | 42155 |
+--------------+-----------+----------------------------+--------+-------+
3 rows in set (0.00 sec)

MySQL [ist425337]>
```

5.

```
select d.name as Medical_Condition, count(distinct p.name_med) as Number_Meds
from prescription as p, diagnosis_code as d
where d.code = p.code
group by d.code
order by Number_Meds;
```

```
MySQL [ist425337]> source Query5.sql
+--------------------+-------------+
| Medical_Condition | Number_Meds |
+--------------------+-------------+
| Kidney Failure    |           1 |
| Teeth Pain        |           1 |
| Inflamed Gum      |           2 |
| Ringworm          |           2 |
+--------------------+-------------+
4 rows in set (0.01 sec)

MySQL [ist425337]>
```

6.

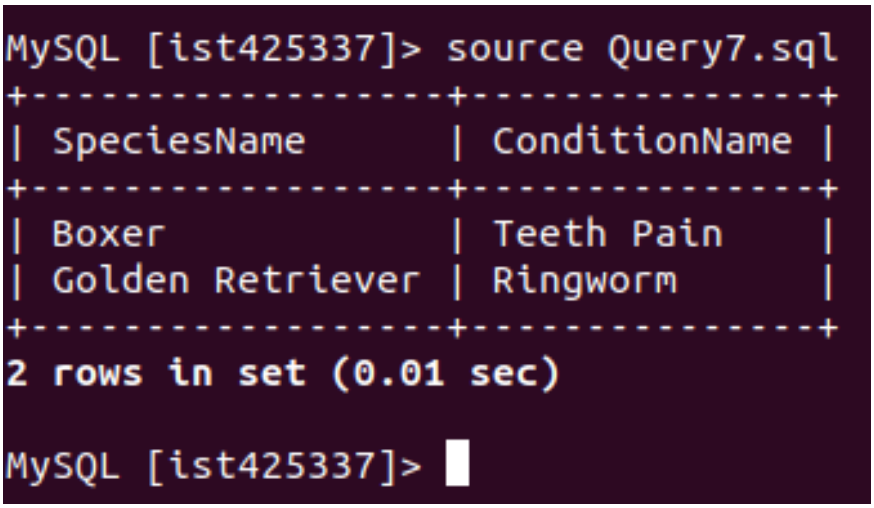
```
select (select count(*)
from participation as par
where extract(year from par.date_timestamp) = 2017
    )/count(*) as Avg_Assistants,
    (select count(*)
    from medical_procedure as pro
    where extract(year from pro.date_timestamp) = 2017
    )/count(*) as Avg_Procedures,
    (select count(*)
    from consult_diagnosis as d
    where extract(year from d.date_timestamp) = 2017
    )/count(*) as Avg_Diagnostic_codes,
    (select count(*)
    from prescription as pre
    where extract(year from pre.date_timestamp) = 2017
    )/count(*) as Avg_Prescriptions
from consult
where extract(year from date_timestamp) = 2017;
```

```
MySQL [ist425337]> source Query6.sql;
+----------------+----------------+----------------------+-------------------+
| Avg_Assistants | Avg_Procedures | Avg_Diagnostic_codes | Avg_Prescriptions |
+----------------+----------------+----------------------+-------------------+
|         0.5556 |         0.4444 |               0.7778 |            0.6667 |
+----------------+----------------+----------------------+-------------------+
1 row in set (0.00 sec)

MySQL [ist425337]>
```

7.

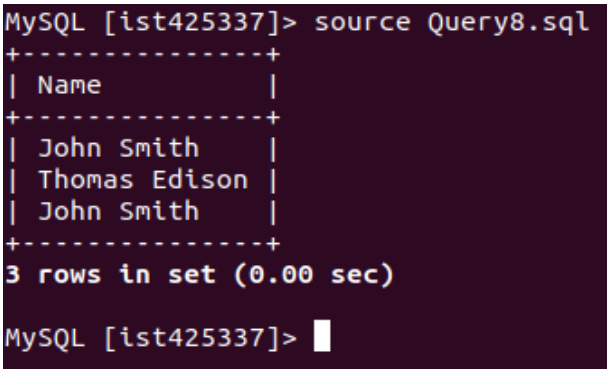
```
select SpeciesName, dc.name as ConditionName
from(
select species_name as SpeciesName, code
      from consult_diagnosis natural join animal, generalization_species
      where name2 = 'Dog'
      and species_name = name1
      group by species_name, code
      having count(*) = (
            select max(c.num)
            from (
                  select species_name as sn, count(*) as num
                  from consult_diagnosis natural join animal, generalization_species
                  where name2 = 'Dog'
                  and species_name = name1
                  group by species_name, code
                  ) as c
            where sn = SpeciesName)
      ) as SpeciesNameCount, diagnosis_code as dc
where dc.code = SpeciesNameCount.code
group by SpeciesName;
```



It was considered that in case of existing 2 different diseases with the same number of cases within the same species, only one of them is displayed.

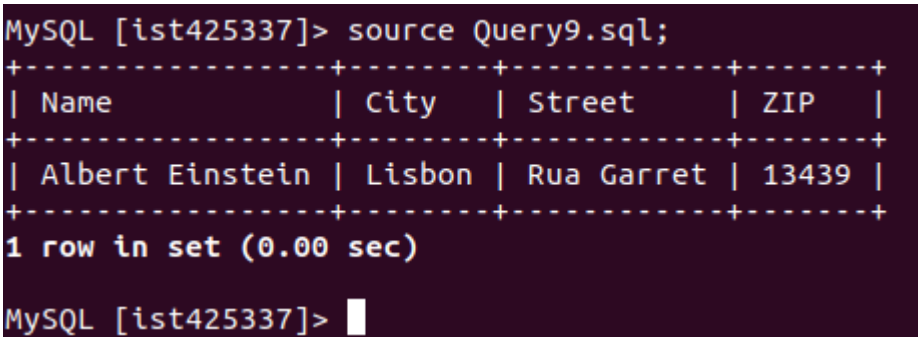
8.

```
select p.name as Name
from person AS p, client AS c, veterinary AS v, assistant AS a
where p.VAT = c.VAT
and (p.VAT = a.VAT OR p.VAT = v.VAT)
group by p.VAT;
```



9.

```
select p.name as Name, p.address_city as City, p.address_street as Street, p.address_zip
as ZIP
from animal as a1, person as p
where not exists(
        select a2.vat
        from animal as a2
        where a2.vat = a1.vat and a2.species_name <> 'Bird'
        )
and p.vat = a1.vat
group by p.vat;
```



4 – Indexes

In the following text we present the indexes created to improve the performance of queries 1 and 2 from section 3.
It should also be noted that no indexes of primary or foreign keys where created because these are automatically created by MySQL.

1.

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

An index on person(name) was created in order to speed up the search of the name “John Smith” in table Person while executing query 1 from section 3.

2.

```
CREATE INDEX indicator_unit ON indicator (units);
CREATE INDEX indicator_ref ON indicator (reference_value);
```

Query 2 from section 3 searches all indicators with units = “milligrams” and reference_value > 100 so two indexes were created, one for the attribute units and another for the attribute reference_value in order to speed up the search of specific values of these parameters.

5 – Updates to the Database

1.

```
update person as p
set address_city = 'Coimbra',
address_street = 'Avenida das Flores'
where exists(
select p.vat
from client as c
where p.name = 'John Smith'
and c.vat = p.vat);
```

It was considered that if there are multiple clients named ‘John Smith’, all of them would have their addresses changed.

2.

```
update indicator as i
set i.reference_value = i.reference_value*1.1
where exists(
select indicator_name
from produced_indicator as pi, test_procedure as tp
where tp.name = pi.name
and tp.vat_owner = pi.vat_owner
and pi.date_timestamp = tp.date_timestamp
and tp.num = pi.num
and tp.test_type = 'blood'
and i.name = pi.indicator_name) and i.units = 'milligrams';
```

3.

```
delete from client
where exists(
select p.name
from person as p
where p.name = 'John Smith'
and client.vat = p.vat);
```

It was considered that if there are multiple clients named ‘John Smith’, all of them would be deleted. It’s notable that the data from ‘John Smith’ is not deleted from the table person.

4.

```
select code
from diagnosis_code
where name = 'kidney failure';

insert into diagnosis_code(code, name)
values(6282, 'end-stage renal disease');

update consult_diagnosis as c
set c.code = 6282
where exists(
select * from produced_indicator as pi
where pi.name = c.name
and pi.VAT_owner = c.VAT_owner
and pi.date_timestamp = c.date_timestamp
and pi.indicator_name = 'Creatinine level'
and pi.indicator_value > 1.0)
and c.code = 6281;
```

6 – View Creation

1.

```
create view dim_date as
select date_timestamp as ConsultDate,
      extract(day from date_timestamp) as Day,
      extract(month from date_timestamp) as Month,
      extract(year from date_timestamp) as Year
 from consult;
```

```
MySQL [ist425337]> select * from dim_date;
+-------------+------+-------+------+
| ConsultDate | Day  | Month | Year |
+-------------+------+-------+------+
| 2016-12-11  |   11 |    12 | 2016 |
| 2016-05-30  |   30 |     5 | 2016 |
| 2017-05-30  |   30 |     5 | 2017 |
| 2017-05-30  |   30 |     5 | 2017 |
| 2017-01-04  |    4 |     1 | 2017 |
| 2017-02-15  |   15 |     2 | 2017 |
| 2017-03-15  |   15 |     3 | 2017 |
| 2017-09-17  |   17 |     9 | 2017 |
| 2017-09-18  |   18 |     9 | 2017 |
| 2017-09-17  |   17 |     9 | 2017 |
| 2017-04-17  |   17 |     4 | 2017 |
| 2018-01-22  |   22 |     1 | 2018 |
| 2018-02-22  |   22 |     2 | 2018 |
+-------------+------+-------+------+
13 rows in set (0.00 sec)

MySQL [ist425337]>
```

2.

```
create view dim_animal as
select name as Animal_Name,
VAT as Animal_Vat,
species_name as Species, Age
from animal;
```

```
MySQL [ist425337]> select * from dim_animal;
+-------------+------------+------------------+------+
| Animal_Name | Animal_Vat | Species          | Age  |
+-------------+------------+------------------+------+
| Alex        |  123456780 | Boxer            |    6 |
| Atila       |  123456784 | Boxer            |    4 |
| Descartes   |  123456789 | Bird             |    1 |
| Faraday     |  123456789 | Cheloniidae      |   12 |
| Galileia    |  123456784 | Golden Retriever |    7 |
| Galileo     |  123456784 | Golden Retriever |    7 |
| Gragas      |  123456780 | Crocodylidae     |   48 |
| Jacinta     |  123456785 | Bird             |    3 |
| Jacinto     |  123456785 | Bird             |    3 |
| Paradox     |  123456783 | Felis Catus      |    5 |
+-------------+------------+------------------+------+
10 rows in set (0.00 sec)

MySQL [ist425337]>
```

3.

```
create view facts_consults as
select distinct
        animal_name as Name,
        animal_vat as Vat,
        ConsultDate as Timestamp,
        (select count(*) from medical_procedure as p where p.name = animal_name and
p.vat_owner = animal_vat and p.date_timestamp = ConsultDate) as Num_Procedures,
        (select count(*) from prescription as pre where pre.name = animal_name and
pre.vat_owner = animal_vat and pre.date_timestamp = ConsultDate) as Num_Medications
    from dim_animal, dim_date, consult as c
    where c.name = animal_name
    and c.vat_owner = animal_vat
    and c.date_timestamp = ConsultDate;
```

```
MySQL [ist425337]> select * from facts_consults;
+-----------+-----------+------------+----------------+-----------------+
| Name      | Vat       | Timestamp  | Num_Procedures | Num_Medications |
+-----------+-----------+------------+----------------+-----------------+
| Alex      | 123456780 | 2016-12-11 |              0 |               1 |
| Atila     | 123456784 | 2017-02-15 |              0 |               0 |
| Atila     | 123456784 | 2017-03-15 |              0 |               0 |
| Descartes | 123456789 | 2017-04-17 |              0 |               0 |
| Faraday   | 123456789 | 2018-01-22 |              0 |               0 |
| Faraday   | 123456789 | 2018-02-22 |              0 |               1 |
| Galileia  | 123456784 | 2017-09-17 |              0 |               0 |
| Galileo   | 123456784 | 2017-09-17 |              0 |               1 |
| Galileo   | 123456784 | 2017-09-18 |              0 |               1 |
| Gragas    | 123456780 | 2016-05-30 |              1 |               0 |
| Gragas    | 123456780 | 2017-05-30 |              1 |               3 |
| Jacinto   | 123456785 | 2017-05-30 |              0 |               0 |
| Paradox   | 123456783 | 2017-01-04 |              3 |               1 |
+-----------+-----------+------------+----------------+-----------------+
13 rows in set (0.00 sec)

MySQL [ist425337]>
```