

Instituto Superior Técnico

**Mestrado Integrado em Engenharia Electrotécnica e de  
Computadores**

**SIBD**

2nd Project Assignment

Group number: 12

Number: 84106

Name: John Daniel Fidalgo Mendonça

Number: 84180

Name: Rúben dos Santos Gomes

Number: 91410

Name: Renato Fraga Trevizan

# Queries

1. List the name, owner name, species, and age, for all animals that participated in consults with a veterinary doctor named John Smith.

```
select animal.name, person.name, species_name, age from animal inner join client inner join person
where exists(
  select consult.name, consult.VAT_owner from consult
  where consult.VAT_vet in(
    select VAT from person natural join veterinary
    where person.name = 'John Smith')
  and consult.name = animal.name
  and consult.VAT_owner = animal.VAT)
and animal.VAT = client.VAT
and client.VAT=person.VAT;
```

In this query, we select the animals whose primary key exists in consults where the veterinary is John Smith. The query result is presented in figure 1.

name	name	species_name	age
Garfield	Jon Arbuckle	cat	4
Odie	Jon Arbuckle	pug	3
Jerry	John Doe	mouse	3
Bob	Ethan Peacock	parrot	1
Oscar	Liam Morgan	labrador	4
Francis	Megan Leonard	siamese	2
Minnie	Megan Leonard	scottish fold	2
Luna	Henry Dennis	labrador	8
Bob	Ethan Williams	canary	0
Tom	Ethan Williams	scottish fold	3
Marley	Ethan Bennett	pug	6

Figure 1 - Query 1 result

2. List the name of all indicators measured in milligrams and with a reference value above 100. The names should be presented together with the corresponding reference value, and sorted according to the reference value, in descending order

```
select name, reference_value from indicator
where units = 'milligrams'
and reference_value > 100
order by reference_value desc;
```

name	reference_value
Kidney Enzyme 2	150.00
Kidney Enzyme	115.00

Figure 2 - Query 2 result

name	reference_value	units	description
Carnine Enzyme	250.00	picograms	Level of enzymes of Carnine
Kidney Enzyme	115.00	milligrams	Level of enzymes of kidney function
Kidney Enzyme 2	150.00	milligrams	Level of enzymes of kidney function 2
Kidney Enzyme 3	50.00	milligrams	Level of enzymes of kidney function 3
Poison	0.00	picograms	Amount of poison

Figure 3- List of all indicators

By analysing figure 3, we see the query 2 result lists only the indicators that meet the requirements.

3. List the name, owner name, species and age for all animals with the most recent weight above 30 kilograms, and where the objective part of any SOAP note, associated to consults of that animal, mentions terms like obesity or obese.

```
select animal.name, person.name, species_name, age from animal inner join person
where exists(
select distinct c1.name, date_timestamp from consult c1
where date_timestamp = (select max(date_timestamp) from consult c2 where c1.name = c2.name and c1.VAT_owner = c2.VAT_owner)
and c1.weight > 30
and c1.VAT_owner = animal.VAT
and c1.VAT_owner = person.VAT)
and exists(
select c2.name from consult c2
where c2.o like '%obese%' or c2.o like '%obesity%'
and c2.VAT_owner = animal.VAT
and c2.VAT_owner = person.VAT);
```

In this query, we select the animals that exists in the most recent consult with their owners, meeting *weight > 0* in that consult and that exists in any consult where the objective part of any SOAP note mentions terms like obesity or obese

name	name	species_name	age
Marley	Ethan Bennett	pug	6

Figure 4 - Query 3 result

In the next figure, we show the query result of listing all consults recorded in the database. The 1<sup>st</sup> marked row represents the most recent consult of Luna, where *weight* is above 30 kg but the objective part does not mention terms like obesity or obese. The 2<sup>nd</sup> marked row represents a consult of Oscar, where *weight* is above 30 kg and the objective part mentions terms like obesity or obese, but it is not the most recent consult of Oscar. The 3<sup>rd</sup> marked row represents the most recent consult of Marley where *weight* is above 30 kg and the objective part mentions obesity so this was the only row that was chosen.

name	VAT_owner	date_timestamp	o	weight
Luna 1	YL 71 92 45 B	2018-11-08 17:45:00	No indication of problems	30.10
Luna	YL 71 92 45 B	2018-11-07 17:00:00	Still lose a big quantity of blood	0.70
Luna	YL 71 92 45 B	2018-10-07 17:00:00	Lose a big quantity of blood	0.70
Luna	YL 71 92 45 B	2018-08-26 10:30:00	No indication of problems	13.00
Oscar	NH 08 51 10	2018-08-14 15:00:00	No indication of problems	28.20
Oreo	EM 97 43 84	2018-06-10 17:00:00	High temperature and reddish throat	4.00
Minnie	WT 72 72 72 A	2018-06-08 10:30:00	Eco shows 3 fetus	2.60
Francis	WT 72 72 72 A	2018-06-08 10:00:00	No indication of problems	2.00
Bob	ZJ 17 07 20	2018-05-30 14:00:00		0.50
Luna	YL 71 92 45 B	2018-05-12 14:00:00	No indication of problems	14.00
Oscar 2	NH 08 51 10	2018-04-14 15:00:00	Obese	40.20
Marley 3	ZJ 37 07 20	2018-03-14 10:00:00	Obesity	40.20
Bob	GS 14 83 38 A	2018-02-23 17:00:00	No indication of problems	0.70
Tom	ZJ 17 07 20	2018-02-24 14:00:00	full healing as shown radiography	1.70
Garfield	CB 46 16 95 B	2018-01-31 18:00:00	Obese	6.20
Shiro	EA 24 06 86	2018-01-30 17:45:00	No indication of problems	8.00
Odie	CB 46 16 95 B	2018-01-30 17:45:00	No indication of problems	10.00
Jerry	CB 46 92 95 B	2018-01-25 16:00:00	urinalysis shows no presence of toxin	0.20
Jerry	CB 46 92 95 B	2018-01-05 10:00:00	urinalysis shows presence of toxin	0.15
Tom	ZJ 17 07 20	2018-01-05 09:00:00	several fractures appeared on radiography	1.50
Francis	WT 72 72 72 A	2017-12-15 12:30:00	High temperature and reddish throat	0.70
Bob	GS 14 83 38 A	2017-10-12 17:45:00	No indication of problems	0.80
Marley	ZJ 37 07 20	2017-09-28 18:00:00	X-ray shows broken limb	27.00
Marley	ZJ 37 07 20	2017-09-08 08:00:00	High level of enzymes	30.80
Marley	ZJ 37 07 20	2017-06-12 15:00:00	No indication of problems	28.00

Figure 5 - List of all consults recorded in the database

- List the name, VAT and address of all clients of the hospital that are not owners of animals.

```
select p.name, p.VAT, p.address_street, p.address_city, p.address_zip from person p, client
where p.VAT = client.VAT and
client.VAT not in (select animal.VAT from animal);
```

name	VAT	address_street	address_city	address_zip
Walter White	ZA 16 10 97	Albuquerque	Albuquerque City	AB7 WVN

Figure 6- Query 4 result

Walter White is the only person that is recorded in *client* relation, but that any animal as his VAT in *animal.VAT*.

- For each possible diagnosis, list the number of distinct medication names that have been prescribed to treat that condition. Sort the results according to the number of distinct medication names, in ascending order.

```
select dc.code, count(distinct m.name) from medication m, diagnosis_code dc, prescription p
where m.name = p.name_med and m.lab = p.lab and m.dosage = p.dosage and
p.code = dc.code
group by dc.code
order by count(m.name) ;
```

code	count(distinct m.name)
KDNFLR	1
POSN	1
BRKTAIL	1
FLFLU	2

Figure 7- Query 5 result

By listing all the prescriptions, selecting the code of the diagnosis and the corresponding medication name we confirm the results. The list of all prescriptions is presented in figure 8.

code	name_med
FLFLU	Ibuprofen
FLFLU	Paracetamol
POSN	Tarinum Forte
BRKTAIL	Paracetamol
BRKTAIL	Paracetamol
KDNFLR	Adaytidah polipstiso
FLFLU	Ibuprofen
FLFLU	Paracetamol
BRKTAIL	Paracetamol

Figure 8- List of all prescriptions

6. Present the average number of assistants, procedures, diagnostic codes, and prescriptions involved in consults from the year of 2017.

```
select count(name_med)/(select count(*) from consult where year(date_timestamp) = 2017) as avg_prescriptions,
count(VAT_assistant)/(select count(*) from consult where year(date_timestamp) = 2017) as avg_assistants,
count(num)/(select count(*) from consult where year(date_timestamp) = 2017) as avg_procedures,
count(cd.code)/(select count(*) from consult where year(date_timestamp) = 2017) as avg_diagnosis_codes
from consult c left outer join prescription p on c.VAT_owner = p.VAT_owner and c.name = p.name and c.date_timestamp = p.date_timestamp
left outer join participation pa on c.VAT_owner = pa.VAT_owner and c.name = pa.name and c.date_timestamp = pa.date_timestamp
left outer join procedure_ pr on c.VAT_owner = pr.VAT_owner and c.name = pr.name and c.date_timestamp = pr.date_timestamp
left outer join consult_diagnosis cd on c.VAT_owner = cd.VAT_owner and c.name = cd.name and c.date_timestamp = cd.date_timestamp
where year(c.date_timestamp) = 2017;
```

In this query we count the total number of all assistants, procedures, diagnostic codes, and prescriptions involved in consults from the year of 2017 and we divide it by the total number of consults from the year of 2017.

avg_prescriptions	avg_assistants	avg_procedures	avg_diagnosis_codes
0.5000	0.0000	0.5000	0.5000

Figure 9- Query 6 result

From 2017 there were 3 prescriptions, 0 assistants, 3 procedures and 3 diagnosis in a list of 6 consults.

7. For each animal sub-species of dog, present the name of the most common disease (i.e., the name associated to the most frequent diagnostic code for consults involving animals of that species).

```
select species_name, name from
(
  select a.species_name, cd.code, dc.name, count(cd.code)
  from animal a inner join consult_diagnosis cd on a.name = cd.name and a.VAT = cd.VAT_owner,
  generalization_species, diagnosis_code as dc
  where a.species_name = generalization_species.name1
  and generalization_species.name2 = 'dog'
  and cd.code = dc.code
  group by a.species_name, cd.code
  order by count(cd.code) desc) as t
group by species_name;
```

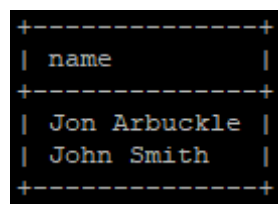
species_name	name
labrador	Broken tail
pug	Broken paw

Figure 10 - Query 7 result

With this implementation, if there are 2 diagnostic names that are equally frequent for a sub-specie of dog, only one of them will be shown.

8. List the names of individuals that are simultaneously clients of the hospital (i.e., owners of animals or clients that have brought animals to consults) and employees of the hospital (i.e., veterinary doctors or assistants).

```
select person.name from person natural join client
where client.VAT in
(select veterinary.VAT from veterinary union select assistant.VAT from assistant);
```



name
Jon Arbuckle
John Smith

Figure 11 - Query 8 result

9. List the names and addresses of clients that only own birds as their pets (i.e., the clients for whom all the owned animals contain the word bird as part of the species name).

```
select distinct p.name, p.address_street, p.address_city, p.address_zip from client c natural join person p
where exists(
  select animal.name, animal.VAT from animal
  where animal.species_name like '%bird%'
  and animal.VAT = c.VAT
  and not exists(
    select a2.name, a2.VAT from animal a2, client c2
    where a2.species_name not like '%bird%'
    and a2.VAT = c2.VAT
    and a2.VAT = animal.VAT
  )
);
```

In this query we select all the persons that have an animal containing the word bird as part of the species name and that have not any animal not containing the word bird as part of the species name.

name	address_street	address_city	address_zip
Ethan Peacock	Streatham Vale	London	SW16 5TB

Figure 12 - Query 9 result

owner	animal	species_name
Ethan Williams	Bobbie	bird
Ethan Peacock	Boby	bird
Ethan Williams	Bob	canary
Jon Arbuckle	Garfield	cat
Henry Dennis	Luna	labrador
Liam Morgan	Oscar	labrador
John Smith	Shiro	labrador
John Doe	Jerry	mouse
Ethan Peacock	Bob	parrot bird
Ethan Bennett	Marley	pug
Jon Arbuckle	Odie	pug
Megan Leonard	Minnie	scottish fold
Ethan Williams	Tom	scottish fold
Megan Leonard	Francis	siamese
Freya Bird	Oreo	siamese

Figure 13 - List of animals and corresponding owners

In the figure above, that represents the list of all animals and corresponding owners, we mark the 2 clients that have birds. We can see that Ethan Williams has two animals that have not the word *bird* in the *species\_name*. Ethan Peacock have 2 pets, and both have the word *bird* in the *species\_name* as we can see in query 9 result.



# Indexes

1.

```
create index person_client
```

```
on client(VAT);
```

```
create index animal_client
```

```
on animal(VAT);
```

```
create index vet_person
```

```
on veterinary(VAT);
```

To improve the performance of Query 1, we can use indexes that store records regarding VAT numbers of all tables involved. With these indexes we can quickly collect the VAT number of the veterinary 'John Smith' and associate the animal's owner VAT with his name on *person*. These indexes therefore reduce the overall time required to perform joins between tables.

id	select type	table	type	possible keys	key	key len	ref	rows	Extra
1	PRIMARY	client	index	person_client	person_client	258	NULL	10	Using index
1	PRIMARY	person	eq_ref	PRIMARY	PRIMARY	257	ist425486.client.VAT	1	
1	PRIMARY	animal	ref	animal_client	animal_client	257	ist425486.client.VAT	1	Using where
2	DEPENDENT SUBQUERY	consult	ref	PRIMARY	PRIMARY	514	ist425486.animal.name,ist425486.animal.VAT	1	Using where
3	DEPENDENT SUBQUERY	veterinary	ref	vet_person	vet_person	258	func	1	Using where; Using index
3	DEPENDENT SUBQUERY	person	eq_ref	PRIMARY	PRIMARY	257	func	1	Using where

6 rows in set (0.00 sec)

Figure 14 - explain query 1

As seen above, we cannot conclude about the performance gain because the table is too small, giving very low execution times. Nevertheless, the query utilizes all indexes created and therefore we will be able to view performance gains in larger databases.

2.

```
create index ind_units
```

```
on indicator(units);
```

```
create index ind_ref
```

```
on indicator(reference_value);
```

As requested in the query, we wish to locate indicators that have milligrams as their units and a reference value of over 100. Instead of traversing the whole table, we can instead index the required fields units and reference\_value. Although not implemented

in *mysql*, a conditional index on `reference_value > 100` would yield the best performance if we performed an intersection between this index and `ind_units`. This however would make this index only usable in similar situation, therefore the general `ind_ref` performs similarly and provides better performance on insertion and deletion.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	indicator	ALL	ind_units,ind_ref	NULL	NULL	NULL	5	Using where; Using filesort

Figure 15 - explain query 2

Although both indexes can be used as keys for the query, the *mysql* query optimizer opts to search the full table to fulfill the query requirements. We can explain this behavior by the existence of the **ORDER BY** clause, which forces the use of *filesort*(full table search). Furthermore, given the very low size of indicators in the table one can argue that these indexes are unnecessary and even worsen overall performance in small datasets.

We want to mention that B+ tree is the default index method in MySQL. A *B+ tree* index can be used for column comparisons in expressions that use the `=`, `>`, `>=`, `<`, `<=` or between operators because it organizes the data in an orderly manner. On the other hand, *hash* indexes can only be used for equality comparisons, but they are very fast. They are not used for comparison operators that find a range of values. By looking at queries one and two we conclude that the hash indexes would perform better for the first query, as the B+ tree would perform better for the second query.

# Database modifications

## 1. update person

```
set person.address_street = 'Main St',  
person.address_city = 'Croydon'  
  
where person.name = 'John Smith' and person.VAT in (select client.VAT from  
client);
```

```
MySQL [ist425480]> select * from person where person.name = 'John Smith';  
+-----+-----+-----+-----+-----+  
| VAT      | name      | address_street | address_city | address_zip |  
+-----+-----+-----+-----+-----+  
| EA 24 06 86 | John Smith | Crown Street   | London       | SW3 1AQ     |  
+-----+-----+-----+-----+-----+
```

Figure 16 - Information about John Smith before modification 1

```
MySQL [ist425480]> select * from person where person.name = 'John Smith';  
+-----+-----+-----+-----+-----+  
| VAT      | name      | address_street | address_city | address_zip |  
+-----+-----+-----+-----+-----+  
| EA 24 06 86 | John Smith | Main St       | Croydon      | SW3 1AQ     |  
+-----+-----+-----+-----+-----+
```

Figure 17 - Information about John Smith after modification 1

## 2. update indicator

```
set indicator.reference_value = indicator.reference_value * 1.1  
  
where indicator.units = 'milligrams' and indicator.name in  
(select produced_indicator.indicator_name  
from produced_indicator natural join test_procedure  
where test_procedure.type = 'blood');
```

```
MySQL [ist425480]> select * from indicator where indicator.units = 'milligrams' and indicator.name in  
-> (select produced_indicator.indicator_name  
-> from produced_indicator natural join test_procedure  
-> where test_procedure.type = 'blood');  
+-----+-----+-----+-----+  
| name          | reference_value | units      | description          |  
+-----+-----+-----+-----+  
| Creatinine level | 0.50 | milligrams | Level of creatinine |
```

Figure 18 - Indicator's reference\_value before modification 2

```
MySQL [ist425480]> select * from indicator where indicator.units = 'milligrams' and indicator.name in
-> (select produced_indicator.indicator_name
-> from produced_indicator natural join test_procedure
-> where test_procedure.type = 'blood');
+-----+-----+-----+-----+
| name          | reference_value | units   | description          |
+-----+-----+-----+-----+
| Creatinine level | 0.55 | milligrams | Level of creatinine |
| Kidney Enzyme   | 126.50 | milligrams | Level of enzymes of kidney function |
+-----+-----+-----+-----+
```

Figure 19 - Indicator's reference\_value after modification 2

### 3. delete from client

where client.VAT in (select person.VAT from person

where person.name = 'John Smith' and person.VAT = client.VAT);

```
MySQL [ist425480]> select person.name, person.VAT, animal.name from animal, person
-> where animal.VAT = person.VAT and person.name = 'John Smith';
+-----+-----+-----+
| name      | VAT      | name    |
+-----+-----+-----+
| John Smith | EA 24 06 86 | Shiro   |
+-----+-----+-----+
```

Figure 20 - John Smith's animal before modification 3

```
MySQL [ist425480]> select person.name, person.VAT, animal.name from animal, person
-> where animal.VAT = person.VAT and person.name = 'John Smith';
Empty set (0.01 sec)
```

Figure 21 - John Smith's modification after modification 3

On creating the database schema, we added the constraint “*on delete cascade*” for all foreign keys, thus, by deleting *John Smith* from the *Client* table, all his animals are deleted, and consequently all consults where they were involved in, and so on...

### 4. insert into diagnosis\_code values('RENDES','End-Stage renal disease');

update consult\_diagnosis as cd

set cd.code = 'RENDES',

cd.name = cd.name,

cd.VAT\_owner = cd.VAT\_owner,

cd.date\_timestamp = cd.date\_timestamp,

where exists (

select \* from produced\_indicator pi natural join test\_procedure tp

```

where pi.indicator_name = 'Creatinine level' and pi.value > 1.0

and tp.type = 'blood' and pi.name = cd.name and pi.VAT_owner = cd.VAT_owner

and pi.date_timestamp = cd.date_timestamp)

and cd.code = 'KDNFLR' ;

```

The relation *prescription* has a foreign key from *consult\_diagnosis* so, by updating a *code* on *consult\_diagnosis*, we have also to update the corresponding *code* in *prescription*. This was achieved by using the **ON UPDATE CASCADE** clause, which updates the child table's foreign key when an update is detected on the parent table, in this case *consult\_diagnosis*.

code	name	VAT_owner	date_timestamp
FLFLU	Francis	WT 72 72 72 A	2018-06-08 10:00:00
DIBT	Garfield	CB 46 16 95 B	2018-01-31 18:00:00
POSN	Jerry	CB 46 92 95 B	2018-01-05 10:00:00
BRKTAIL	Luna	YL 71 92 45 B	2018-10-07 17:00:00
BRKTAIL	Luna	YL 71 92 45 B	2018-11-07 17:00:00
BRKPAW	Marley	ZJ 37 07 20	2017-06-12 15:00:00
KDNFLR	Marley	ZJ 37 07 20	2017-09-08 08:00:00
DIBT	Marley	ZJ 37 07 20	2018-03-14 10:00:00
BRKPAW	Odie	CB 46 16 95 B	2017-11-12 18:00:00
BRKTAIL	Tom	ZJ 17 07 20	2018-01-05 09:00:00

Figure 22 - List of all *consult\_diagnosis* before update 4

name	VAT_owner	date_timestamp	num	indicator_name	value
Jerry	CB 46 92 95 B	2018-01-25 16:00:00	1	Poison	9
Marley	ZJ 37 07 20	2017-09-08 08:00:00	1	Creatinine level	2
Marley	ZJ 37 07 20	2017-09-08 08:00:00	1	Kidney Enzyme	2

Figure 23 - List of all produced\_indicators

As we can see, there is only one *consult\_diagnosis* corresponding to kidney failure meeting all the requirements to update to end-stage renal disease. In the next figure we present the list of all *consult\_diagnosis* after the update.

code	name	VAT_owner	date_timestamp
FLFLU	Francis	WT 72 72 72 A	2018-06-08 10:00:00
DIBT	Garfield	CB 46 16 95 B	2018-01-31 18:00:00
POSN	Jerry	CB 46 92 95 B	2018-01-05 10:00:00
BRKTAIL	Luna	YL 71 92 45 B	2018-10-07 17:00:00
BRKTAIL	Luna	YL 71 92 45 B	2018-11-07 17:00:00
BRKPAW	Marley	ZJ 37 07 20	2017-06-12 15:00:00
RENDES	Marley	ZJ 37 07 20	2017-09-08 08:00:00
DIBT	Marley	ZJ 37 07 20	2018-03-14 10:00:00
BRKPAW	Odie	CB 46 16 95 B	2017-11-12 18:00:00
BRKTAIL	Tom	ZJ 17 07 20	2018-01-05 09:00:00

Figure 24 - List of all consult\_diagnosis after update 4

As expected, the code *KDNFLR* was replaced by *RENDES*.

# Views

## 1. create view dim\_date as

```
(select distinct date_timestamp, day(date_timestamp) as day,  
month(date_timestamp) as month, year(date_timestamp) as year  
from consult);
```

```
MySQL [ist425480]> select date_timestamp  
-> from consult  
-> order by date_timestamp;  
+-----+  
| date_timestamp |  
+-----+  
| 2017-06-12 15:00:00 |  
| 2017-09-08 08:00:00 |  
| 2017-09-28 18:00:00 |  
| 2017-10-12 17:45:00 |  
| 2017-11-12 18:00:00 |  
| 2017-12-15 12:30:00 |  
| 2018-01-05 09:00:00 |  
| 2018-01-05 10:00:00 |  
| 2018-01-25 16:00:00 |  
| 2018-01-30 17:45:00 |  
| 2018-01-30 17:45:00 |  
| 2018-01-31 18:00:00 |  
| 2018-02-24 14:00:00 |  
| 2018-02-25 17:00:00 |  
| 2018-03-14 10:00:00 |  
| 2018-04-14 15:00:00 |  
| 2018-05-12 14:00:00 |  
| 2018-05-30 14:00:00 |  
| 2018-06-08 10:00:00 |  
| 2018-06-08 10:30:00 |  
| 2018-06-10 17:00:00 |  
| 2018-08-14 15:00:00 |  
| 2018-08-26 10:30:00 |  
| 2018-10-07 17:00:00 |  
| 2018-11-07 17:00:00 |  
+-----+
```

Figure 25 - List of dates of all consults ordered by date

```
MySQL [ist425480]> select * from dim_date  
-> order by date_timestamp;  
+-----+-----+-----+-----+  
| date_timestamp | day | month | year |  
+-----+-----+-----+-----+  
| 2017-06-12 15:00:00 | 12 | 6 | 2017 |  
| 2017-09-08 08:00:00 | 8 | 9 | 2017 |  
| 2017-09-28 18:00:00 | 28 | 9 | 2017 |  
| 2017-10-12 17:45:00 | 12 | 10 | 2017 |  
| 2017-11-12 18:00:00 | 12 | 11 | 2017 |  
| 2017-12-15 12:30:00 | 15 | 12 | 2017 |  
| 2018-01-05 09:00:00 | 5 | 1 | 2018 |  
| 2018-01-05 10:00:00 | 5 | 1 | 2018 |  
| 2018-01-25 16:00:00 | 25 | 1 | 2018 |  
| 2018-01-30 17:45:00 | 30 | 1 | 2018 |  
| 2018-01-30 17:45:00 | 30 | 1 | 2018 |  
| 2018-01-31 18:00:00 | 31 | 1 | 2018 |  
| 2018-02-24 14:00:00 | 24 | 2 | 2018 |  
| 2018-02-25 17:00:00 | 25 | 2 | 2018 |  
| 2018-03-14 10:00:00 | 14 | 3 | 2018 |  
| 2018-04-14 15:00:00 | 14 | 4 | 2018 |  
| 2018-05-12 14:00:00 | 12 | 5 | 2018 |  
| 2018-05-30 14:00:00 | 30 | 5 | 2018 |  
| 2018-06-08 10:00:00 | 8 | 6 | 2018 |  
| 2018-06-08 10:30:00 | 8 | 6 | 2018 |  
| 2018-06-10 17:00:00 | 10 | 6 | 2018 |  
| 2018-08-14 15:00:00 | 14 | 8 | 2018 |  
| 2018-08-26 10:30:00 | 26 | 8 | 2018 |  
| 2018-10-07 17:00:00 | 7 | 10 | 2018 |  
| 2018-11-07 17:00:00 | 7 | 11 | 2018 |  
+-----+-----+-----+-----+
```

Figure 26 - List of all attributes of dim\_date ordered by date

It was necessary to use *distinct* because *date\_timestamp* is not the primary key of *consult*, so if there were 2 consults with the same *date\_timestamp* of different animals, it would appear two equal *date\_timestamp* in relation *dim\_date*, but since *date\_timestamp* is its key, it cannot happen.

## 2. create view dim\_animal as

(select animal.name as animal\_name, animal.VAT as animal\_vat,  
animal.species\_name as species, animal.age as age from animal);

```
MySQL [ist425480]> select * from animal order by name;;
```

name	VAT	species_name	colour	gender	birth_year	age
Bob	GS 14 83 38 A	parrot bird	Green	M	2017	1
Bob	ZJ 17 07 20	canary	Grey	M	2018	0
Bobbie	ZJ 17 07 20	bird	Grey	M	2018	0
Boby	GS 14 83 38 A	bird	Blue	M	2017	1
Francis	WT 72 72 72 A	siamese	Black	M	2016	2
Garfield	CB 46 16 95 B	cat	Orange	M	2014	4
Jerry	CB 46 92 95 B	mouse	Brown	M	2015	3
Luna	YL 71 92 45 B	labrador	Black	F	2010	8
Marley	ZJ 37 07 20	pug	Brown	M	2012	6
Minnie	WT 72 72 72 A	scottish fold	Black	F	2016	2
Odie	CB 46 16 95 B	pug	White	M	2015	3
Oreo	EM 97 43 84	siamese	White	M	2017	1
Oscar	NH 08 51 10	labrador	Grey	M	2014	4
Shiro	EA 24 06 86	labrador	Brown	M	2017	1
Tom	ZJ 17 07 20	scottish fold	Grey	M	2015	3

Figure 27 - List of all animals ordered by name

```
MySQL [ist425480]> select * from dim_animal order by animal_name;
```

animal_name	animal_vat	species	age
Bob	GS 14 83 38 A	parrot bird	1
Bob	ZJ 17 07 20	canary	0
Bobbie	ZJ 17 07 20	bird	0
Boby	GS 14 83 38 A	bird	1
Francis	WT 72 72 72 A	siamese	2
Garfield	CB 46 16 95 B	cat	4
Jerry	CB 46 92 95 B	mouse	3
Luna	YL 71 92 45 B	labrador	8
Marley	ZJ 37 07 20	pug	6
Minnie	WT 72 72 72 A	scottish fold	2
Odie	CB 46 16 95 B	pug	3
Oreo	EM 97 43 84	siamese	1
Oscar	NH 08 51 10	labrador	4
Shiro	EA 24 06 86	labrador	1
Tom	ZJ 17 07 20	scottish fold	3

Figure 28 - List of all attributes of dim\_animal ordered by name

Here it was not necessary to use *distinct* because we are selecting rows of the relation *animal* including its primary key so there will not be two equal rows in relation *dim\_animal*.



3.

```
create view prc_tbl as (select da.animal_name, da.animal_vat,  
dd.date_timestamp,count(prc.num) as procedures  
  
from dim_date dd natural join consult c left outer join procedure_ prc  
  
on c.name = prc.name and c.VAT_owner = prc.VAT_owner and c.date_timestamp =  
prc.date_timestamp  
  
left outer join dim_animal as da  
  
on c.name = da.animal_name and c.VAT_owner = da.animal_vat  
  
group by dd.date_timestamp);
```

```
create view prs_tbl as (select da.animal_name, da.animal_vat,  
dd.date_timestamp,count(prs.name_med) as medications  
  
from dim_date dd natural join consult c left outer join prescription prs  
  
on c.name = prs.name and c.VAT_owner = prs.VAT_owner and c.date_timestamp =  
prs.date_timestamp  
  
left outer join dim_animal as da  
  
on c.name = da.animal_name and c.VAT_owner = da.animal_vat  
  
group by dd.date_timestamp);
```

```
create view facts_consults as  
  
select prc_tbl.animal_name as name, prc_tbl.animal_vat as vat, prc_tbl.date_timestamp as  
timestamp, procedures as num_procedures, medications as num_medications  
  
from prc_tbl left join prs_tbl  
  
on prc_tbl.animal_name= prs_tbl.animal_name and prc_tbl.animal_vat= prs_tbl.animal_vat  
and prc_tbl.date_timestamp = prs_tbl.date_timestamp  
  
union  
  
select prs_tbl.animal_name, prs_tbl.animal_vat, prs_tbl.date_timestamp, procedures,  
medications  
  
from prc_tbl right join prs_tbl  
  
on prc_tbl.animal_name= prs_tbl.animal_name and prc_tbl.animal_vat= prs_tbl.animal_vat  
and prc_tbl.date_timestamp = prs_tbl.date_timestamp;
```

This view was more complex due to the necessity to join more queries. We decided to create two views before creating *fact\_consults*, which corresponds to the union of both

to accomplish the view that was requested. The **UNION** paired with the **LEFT JOIN** and **RIGHT JOIN** is a workaround for the **FULL OUTER JOIN**, which is not implemented in *mysql*. A full outer join provides us with a table that contains both information regarding prescriptions and procedures, joining in the same line(consult) this information. Consults with only one of these would show the missing information as **NULL**, i.e. no medication/procedure was prescribed. The first auxiliary table *prc\_tbl* selects the information regarding procedures while *prs\_tbl* stores information regarding prescriptions(medication).

animal_name	animal_vat	date_timestamp	procedures	medications
Marley	ZJ 37 07 20	2017-06-12 15:00:00	0	0
Marley	ZJ 37 07 20	2017-09-08 08:00:00	1	1
Marley	ZJ 37 07 20	2017-09-28 18:00:00	1	0
Bob	GS 14 83 38 A	2017-10-12 17:45:00	0	0
Odie	CB 46 16 95 B	2017-11-12 18:00:00	1	0
Francis	WT 72 72 72 A	2017-12-15 12:30:00	0	2
Tom	ZJ 17 07 20	2018-01-05 09:00:00	0	0
Jerry	CB 46 92 95 B	2018-01-05 10:00:00	0	0
Jerry	CB 46 92 95 B	2018-01-25 16:00:00	1	1
Odie	CB 46 16 95 B	2018-01-30 17:45:00	0	0
Garfield	CB 46 16 95 B	2018-01-31 18:00:00	0	0
Tom	ZJ 17 07 20	2018-02-24 14:00:00	1	1
Bob	GS 14 83 38 A	2018-02-25 17:00:00	0	0
Marley	ZJ 37 07 20	2018-03-14 10:00:00	0	0
Oscar	NH 08 51 10	2018-04-14 15:00:00	0	0
Luna	YL 71 92 45 B	2018-05-12 14:00:00	0	0
Bob	ZJ 17 07 20	2018-05-30 14:00:00	0	0
Francis	WT 72 72 72 A	2018-06-08 10:00:00	0	0
Minnie	WT 72 72 72 A	2018-06-08 10:30:00	1	0
Oreo	EM 97 43 84	2018-06-10 17:00:00	0	2
Oscar	NH 08 51 10	2018-08-14 15:00:00	0	0
Luna	YL 71 92 45 B	2018-08-26 10:30:00	0	0
Luna	YL 71 92 45 B	2018-10-07 17:00:00	0	1
Luna	YL 71 92 45 B	2018-11-07 17:00:00	0	1

Figure 29 - List of all attributes of facts\_consults

The result met the expectations. As we can see, the animals that were not involved in procedures have the value 0 in the column *procedures*. Similarly, the animals that were not prescribed with any medications have the value 0 in column *medications*. All consults are listed in this view and show the correct number of procedures and medications prescribed.

# Database Schema Creation

Below we present the code used to create the database schema:

```
drop view if exists dim_animal, dim_date, facts_consults, prc_tbl, prs_tbl;
```

```
drop table if exists produced_indicator;  
drop table if exists test_procedure;  
drop table if exists radiography;  
drop table if exists indicator;  
drop table if exists performed;  
drop table if exists procedure_  
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 phone_number;  
drop table if exists client;  
drop table if exists person;
```

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

```
create table phone_number (  
    VAT varchar(255),  
    phone varchar(255),
```

```

        primary key(phone),
        foreign key(VAT) references person(VAT) on delete cascade on update cascade
    );

create table client (
    VAT varchar(255),
    foreign key(VAT) references person(VAT) on delete cascade on update cascade
);

create table veterinary (
    VAT varchar(255),
    specialization varchar(255),
    bio varchar(255),
    foreign key(VAT) references person(VAT) on delete cascade on update cascade
);

create table assistant (
    VAT varchar(255),
    foreign key(VAT) references person(VAT) on delete cascade on update cascade
);

create table species (
    name varchar(255),
    descc 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 on update cascade,
    foreign key (name2) references species(name) on delete cascade on update cascade
);

create table animal (
    name varchar(255),
    VAT varchar(255),
    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 on update cascade,
    foreign key(species_name) references species(name) on delete cascade on update cascade
);

create table consult (
    name varchar(255),
    VAT_owner varchar(255),
    date_timestamp timestamp,
    s varchar(255),
    o varchar(255),
    aa varchar(255),
    p varchar(255),
    VAT_client varchar(255),
    VAT_vet varchar(255),
    weight numeric(5,2),
    primary key(name, VAT_owner, date_timestamp),
    foreign key(name, VAT_owner) references animal(name,VAT) on delete cascade on update cascade,
    foreign key(VAT_client) references client(VAT) on delete cascade on update cascade,
    foreign key(VAT_vet) references veterinary(VAT) on delete cascade on update cascade,
    check(weight>0)
);

create table participation (
    name varchar(255),
    VAT_owner varchar(255),
    date_timestamp timestamp,
    VAT_assistant varchar(255),
    primary key(name, VAT_owner, date_timestamp, VAT_assistant),
    foreign key(name, VAT_owner, date_timestamp) references consult(name,VAT_owner,date_timestamp)
    foreign key(VAT_assistant) references assistant(VAT) on delete cascade on update cascade
);

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 varchar(255),
        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)
        foreign key(code) references diagnosis_code(code) on delete cascade on update cascade
    );

```

```

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 varchar(255),
    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(name, VAT_owner, date_timestamp) references consult(name,VAT_owner,date_timestamp)
    foreign key(name_med, lab, dosage) references medication(name,lab,dosage) on delete cascade on update cascade
    foreign key(code) references diagnosis_code(code) on delete cascade on update cascade
);

```

```

create table indicator (
    name varchar(255),
    reference_value numeric(10,2),
    units varchar(255),
    description varchar(255),
    primary key(name)
);

```

```

create table procedure_ (
    name varchar(255),
    VAT_owner varchar(255),
    date_timestamp timestamp,

```

```

        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)
    );

```

```

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

```

```

create table radiography (
    name varchar(255),
    VAT_owner varchar(255),
    date_timestamp timestamp,
    num int,
    file varchar(255),
    primary key(name, VAT_owner, date_timestamp, num),
    foreign key(name, VAT_owner, date_timestamp, num) references procedure_(name, VAT_owner, date_timestamp,num)
);

```

```

create table test_procedure (
    name varchar(255),
    VAT_owner varchar(255),
    date_timestamp timestamp,
    num int,
    type varchar(255),
    primary key(name, VAT_owner, date_timestamp, num),
    foreign key(name, VAT_owner, date_timestamp, num) references procedure_(name, VAT_owner, date_timestamp,num),
    check(type = 'blood' or type='urine')
);

```

```

create table produced_indicator (
    name varchar(255),
    VAT_owner varchar(255),
    date_timestamp timestamp,

```

```
num int,  
indicator_name varchar(255),  
value float,  
primary key(name, VAT_owner, date_timestamp, num, indicator_name),  
foreign key(name, VAT_owner, date_timestamp, num) references procedure_(name, VAT_owner, date_  
foreign key(indicator_name) references indicator(name) on delete cascade on update cascade  
);
```



# Database population

Below we present the code used to populate the database:

```
insert into species values ('mammal', 'Mammals are a clade of endothermic amniotes that possesses a neocortex, hair, three middle ear bones, and mammary glands. ');
```

```
insert into species values ('aviary', 'An aviary is a large enclosure for confining birds.');
```

```
insert into species values ('cat', 'A cat is a small, typically furry, carnivorous mammal');
```

```
insert into generalization_species values ('cat','mammal');
```

```
insert into species values ('dog','Dogs (Canis lupus familiaris) are domesticated mammals, not natural wild animals. They were originally bred from wolves');
```

```
insert into generalization_species values ('dog','mammal');
```

```
insert into species values ('bird', 'Birds are a group of vertebrates which evolved from dinosaurs. They are endothermic, with feathers');
```

```
insert into generalization_species values ('bird','aviary');
```

```
insert into species values ('mouse','A mouse is a rodent, which is a kind of mammal. It has a pointed snout, small rounded ears, a body-length scaly tail and a high breeding rate');
```

```
insert into generalization_species values ('mouse','mammal');
```

```
insert into species values ('labrador','Labrador is a type of retriever-gun dog. The Labrador is one of the most popular breeds of dog in Canada, the United Kingdom and the United States.');
```

```
insert into generalization_species values ('labrador','dog');
```

```
insert into species values ('parrot bird','Parrots, also known as psittacines are birds of the roughly 393 species in 92 genera that make up the order Psittaciformes, found in most tropical and subtropical regions');
```

```
insert into generalization_species values ('parrot bird','bird');
```

```
insert into species values ('canary','The domestic canary, often simply known
as the canary is a domesticated form of the wild canary, a small songbird in
the finch family originating from the Macaronesian Islands');
```

```
insert into generalization_species values ('canary','bird');
```

```
insert into species values ('pug','The Pug is a breed of dog with physically
distinctive features of a wrinkly, short-muzzled face, and curled tail.');
```

```
insert into generalization_species values ('pug','dog');
```

```
insert into species values ('siamese','The Siamese cat is one of the first
distinctly recognized breeds of Asian cat.');
```

```
insert into generalization_species values ('siamese','cat');
```

```
insert into species values ('scottish fold','The Scottish Fold is a breed of
domestic cat with a natural dominant-gene mutation that affects cartilage
throughout the body, causing the ears to fold.');
```

```
insert into generalization_species values ('scottish fold','cat');
```

```
-- Persons and their respective pets
```

```
--client
```

```
insert into person values ('EA 24 06 86','John Smith','Crown
Street','London','SW3 1AQ');
```

```
insert into phone_number values('EA 24 06 86','070 2699 7334');
```

```
insert into veterinary values('EA 24 06 86', 'General Practitioner', 'Pop
culture geek. Vet. Unapologetic social media lover. Travel fanatic.');
```

```
insert into client values('EA 24 06 86');
```

```
insert into animal values('Shiro','EA 24 06
86','labrador','Brown','M','2017','1');
```

```
--veterinary
```

```
insert into person values('CY 10 60 11 A',' Liz Wilson','Guild
Street','London','EC1P 1RE');
```

```
insert into phone_number values('CY 10 60 11 A','079 4132 9210');
```

```
insert into veterinary values('CY 10 60 11 A', 'Surgery', 'Coffee buff. Web
enthusiast. Surgery Vet. Gamer. Avid organizer.');
```

```
--assistant
```

```
insert into person values('SM 62 10 29 D','Joshua Pearson','Parker
Road','Croydon','CR7 QS8');
```

```
insert into phone_number values('SM 62 10 29 D','070 7514 9424');
```

```

insert into assistant values('SM 62 10 29 D');

--client
insert into person values('CB 46 16 95 B','Jon Arbuckle','Sumner
Road','Croydon','CR7 H3P');
insert into phone_number values('CB 46 16 95 B','070 2149 6062');
insert into assistant values('CB 46 16 95 B');
insert into client values('CB 46 16 95 B');
insert into animal values('Garfield','CB 46 16 95
B','cat','Orange','M','2014','4');
insert into animal values('Odie','CB 46 16 95 B','pug','White','M','2015','3');

--client without animal
insert into person values('ZA 16 10 97','Walter White',
'Albuquerque','Albuquerque City','AB7 WWN');
insert into phone_number values('ZA 16 10 97','078 6911 1109');
insert into client values('ZA 16 10 97');

--client
insert into person values('NH 08 51 10','Liam Morgan','Ecclesbourne
Road','Croydon','CR7 7BN');
insert into phone_number values('NH 08 51 10','078 6008 1591');
insert into client values('NH 08 51 10');
insert into animal values('Oscar','NH 08 51
10','labrador','Grey','M','2014','4');

--client
insert into person values('EM 97 43 84','Freya Bird','London
Road','London','SW16 4DP');
insert into phone_number values('EM 97 43 84','077 7080 5592');
insert into client values('EM 97 43 84');
insert into animal values('Oreo','EM 97 43
84','siamese','White','M','2017','1');

--client
insert into person values('GS 14 83 38 A','Ethan Peacock','Streatham
Vale','London','SW16 5TB');
insert into phone_number values('GS 14 83 38 A','079 7140 4420');
insert into client values('GS 14 83 38 A');
insert into animal values('Bob','GS 14 83 38 A','parrot
bird','Green','M','2017','1');
insert into animal values('Boby','GS 14 83 38 A','bird','Blue','M','2017','1');

--client
insert into person values('YL 71 92 45 B','Henry Dennis','Grove
Road','Mitcham','CR4 1AB');

```

```

insert into phone_number values('YL 71 92 45 B','079 1080 0169');
insert into client values('YL 71 92 45 B');
insert into animal values('Luna','YL 71 92 45
B','labrador','Black','F','2010','8');

--client
insert into person values('ZJ 37 07 20','Ethan Bennett','Jebb
Avenue','London','SW2 5XF');
insert into phone_number values('ZJ 37 07 20','079 7422 0554');
insert into client values('ZJ 37 07 20');
insert into animal values('Marley','ZJ 37 07 20','pug','Brown','M','2012','6');

--client
insert into person values('WT 72 72 72 A','Megan Leonard','Streatham
Road','London','SW16 1BS');
insert into phone_number values('WT 72 72 72 A','077 7408 9030');
insert into client values('WT 72 72 72 A');
insert into animal values('Francis','WT 72 72 72
A','siamese','Black','M','2016','2');
insert into animal values('Minnie','WT 72 72 72 A','scottish
fold','Black','F','2016','2');

--client
insert into person values('CB 46 92 95 B','John Doe','Crown Lane','London',
'SW16 3JG');
insert into phone_number values('CB 46 92 95 B','070 2179 6060');
insert into client values('CB 46 92 95 B');
insert into animal values('Jerry','CB 46 92 95
B','mouse','Brown','M','2015','3');

--client
insert into person values('ZJ 17 07 20','Ethan Williams','Sutton Ct Road',
'Sutton','SM1 4RQ');
insert into phone_number values('ZJ 17 07 20','079 7422 0004');
insert into client values('ZJ 17 07 20');
insert into animal values('Tom','ZJ 17 07 20','scottish
fold','Grey','M','2015','3');
insert into animal values('Bob','ZJ 17 07 20','canary','Grey','M','2018','0');
insert into animal values('Bobbie','ZJ 17 07 20','bird','Grey','M','2018','0');

--Consults and respective tests, diagnostics, medications

insert into consult values('Tom','ZJ 17 07 20','2018-01-05 09:00:00','Tail
appears broken after an iron fell on top of it','several fractures appeared on

```

```
radiography','several fractures from trauma','cast on the tail','ZJ 17 07 20','EA 24 06 86',1.5);
```

```
insert into consult values('Tom','ZJ 17 07 20','2018-02-24 14:00:00','Tom  
although disturbed from the presence of the cast, managed to continue normal  
life','full healing as shown radiography','clear','remove cast','ZJ 17 07  
20','EA 24 06 86',1.7);  
insert into diagnosis_code values('BRKTAIL','Broken tail');  
insert into procedure_ values ('Tom','ZJ 17 07 20','2018-02-24  
14:00:00',1,'Tail X-Ray');  
insert into radiography values ('Tom','ZJ 17 07 20','2018-02-24  
14:00:00',1,'C:x-ray');  
insert into consult_diagnosis values('BRKTAIL','Tom','ZJ 17 07 20','2018-01-05  
09:00:00');  
insert into performed values('Tom','ZJ 17 07 20','2018-02-24 14:00:00',1,'SM 62  
10 29 D');  
insert into medication values('Paracetamol','Dayer','100mg');  
insert into prescription values('BRKTAIL','Tom','ZJ 17 07 20','2018-02-24  
14:00:00','Paracetamol','Dayer','100mg','12/12h');
```

```
insert into consult values('Bob','ZJ 17 07 20','2018-05-30 14:00:00','Nothing  
of concern','','','ZJ 17 07 20','EA 24 06 86',0.5);
```

```
insert into consult values('Jerry','CB 46 92 95 B','2018-01-05 10:00:00','Jerry  
doesnt want to eat and remains still all the time','urinalysis shows presence  
of toxin','poisoning','admission to clinic and surveillance','CB 46 92 95  
B','EA 24 06 86',0.15);  
insert into consult values('Jerry','CB 46 92 95 B','2018-01-25 16:00:00','Jerry  
eats and plays','urinalysis shows no presence of  
toxin','clear','discharge','CB 46 92 95 B','EA 24 06 86',0.2);  
insert into diagnosis_code values('POSN','Poisoning');  
insert into procedure_ values ('Jerry','CB 46 92 95 B','2018-01-25  
16:00:00',1,'Poison');  
insert into test_procedure values ('Jerry','CB 46 92 95 B','2018-01-25  
16:00:00',1,'urine');
```

```
insert into indicator values ('Poison','0','picograms','Amount of poison');  
insert into indicator values ('Kidney Enzyme 2','150','milligrams','Level of  
enzymes of kidney function 2');  
insert into indicator values ('Creatinine level','0.5','milligrams','Level of  
creatinine');  
insert into indicator values ('Carnine Enzyme','250','picograms','Level of  
enzymes of Carnine');
```

```

insert into produced_indicator values ('Jerry','CB 46 92 95 B','2018-01-25
16:00:00',1,'Poison','9');

insert into medication values('Tarinum Forte','Dayer','10mg');
insert into prescription values('POSN','Jerry','CB 46 92 95 B','2018-01-25
16:00:00','Tarinum Forte','Dayer','10mg','12/12h for 5 days IV');
insert into consult_diagnosis values('POSN','Jerry','CB 46 92 95 B','2018-01-05
10:00:00');


insert into consult values('Marley','ZJ 37 07 20','2018-03-14 10:00:00','Looks
fat and is unable to jump','Obesity','high weight ','Medication, meal cut and
excercise','ZJ 37 07 20','EA 24 06 86',40.2);
insert into diagnosis_code values('DIBT','Diabetes');
insert into consult_diagnosis values('DIBT','Marley','ZJ 37 07 20','2018-03-14
10:00:00');


insert into consult values('Marley','ZJ 37 07 20','2017-09-08 08:00:00','Doesnt
want to eat or play','High level of enzymes','Kidney failure','Medication and
admission','ZJ 37 07 20','CY 10 60 11 A',30.8);
insert into diagnosis_code values('KDNFLR','Kidney failure');
insert into medication values('Adaytidah polipstiso','Dayer','50mg');
insert into prescription values('KDNFLR','Marley','ZJ 37 07 20','2017-09-08
08:00:00','Adaytidah polipstiso','Dayer','50mg','everyday for 2 weeks');
insert into procedure_ values ('Marley','ZJ 37 07 20','2017-09-08
08:00:00',1,'Enzyme test');
insert into test_procedure values ('Marley','ZJ 37 07 20','2017-09-08
08:00:00',1,'blood');
insert into indicator values ('Kidney Enzyme','115','milligrams','Level of
enzymes of kidney function');
insert into produced_indicator values ('Marley','ZJ 37 07 20','2017-09-08
08:00:00',1,'Kidney Enzyme','2');
insert into produced_indicator values ('Marley','ZJ 37 07 20','2017-09-08
08:00:00',1,'Creatinine level','2');
insert into consult_diagnosis values('KDNFLR','Marley','ZJ 37 07 20','2017-09-
08 08:00:00');
insert into participation values ('Marley','ZJ 37 07 20','2018-03-14
10:00:00','CB 46 16 95 B');


insert into consult values('Marley','ZJ 37 07 20','2017-06-12
15:00:00','Nothing in particular','No indication of problems','Healthy','Annual
vaccination','ZJ 37 07 20','EA 24 06 86',28);

```

```

insert into consult values('Marley','ZJ 37 07 20','2017-09-28 18:00:00','Limps
on the left paw','X-ray shows broken limb','Broken paw','Cast','ZJ 37 07
20','CY 10 60 11 A',27);
insert into diagnosis_code values('BRKPAW','Broken paw');
insert into procedure_ values ('Marley','ZJ 37 07 20','2017-09-28
18:00:00',1,'paw X-Ray');
insert into radiography values ('Marley','ZJ 37 07 20','2017-09-28
18:00:00',1,'C:x-ray');
insert into consult_diagnosis values('BRKPAW','Marley','ZJ 37 07 20','2017-06-
12 15:00:00');

```

```

insert into consult values('Francis','WT 72 72 72 A','2018-06-08
10:00:00','Nothing in particular','No indication of problems','Healthy','Annual
vaccination','WT 72 72 72 A','EA 24 06 86',2.0);
insert into consult values('Francis','WT 72 72 72 A','2017-12-15
12:30:00','Sneezes a lot, doesnt eat or play','High temperature and reddish
throat','Feline Flue','Medication','WT 72 72 72 A','CY 10 60 11 A',0.7);
insert into diagnosis_code values('FLFLU','Feline Flu');
insert into medication values('Ibuprofen','Mistic','35mg');
insert into prescription values('FLFLU','Francis','WT 72 72 72 A','2017-12-15
12:30:00','Paracetamol','Dayer','100mg','12/12h until next consult');
insert into prescription values('FLFLU','Francis','WT 72 72 72 A','2017-12-15
12:30:00','Ibuprofen','Mistic','35mg','12/12h until next consult');
insert into consult_diagnosis values('FLFLU','Francis','WT 72 72 72 A','2018-
06-08 10:00:00');

```

```

insert into consult values('Minnie','WT 72 72 72 A','2018-06-08
10:30:00','Large belly','Eco shows 3 fetus','Pregnant','No action required','WT
72 72 72 A','EA 24 06 86',2.6);
insert into procedure_ values ('Minnie','WT 72 72 72 A','2018-06-08
10:30:00',1,'Abdominal Eco');
insert into radiography values ('Minnie','WT 72 72 72 A','2018-06-08
10:30:00',1,'C:Ecos');

```

```

insert into consult values('Luna','YL 71 92 45 B','2018-05-12
14:00:00','Nothing in particular','No indication of problems','Healthy','Annual
vaccination','YL 71 92 45 B','EA 24 06 86',14);

```

```
insert into consult values('Luna','YL 71 92 45 B','2018-08-26
10:30:00','Nothing in particular','No indication of problems','Healthy','Annual
vaccination','YL 71 92 45 B','EA 24 06 86',13);
```

```
insert into consult values('Bob','GS 14 83 38 A','2017-10-12 17:45:00','Nothing
in particular','No indication of problems','Healthy','No action required','GS
14 83 38 A','EA 24 06 86',0.8);
```

```
insert into consult values('Bob','GS 14 83 38 A','2018-02-25 17:00:00','Found
outdoors, appears lost','No indication of problems','Healthy','No action
required','ZJ 17 07 20','EA 24 06 86',0.7);
```

```
insert into consult values('Oreo','EM 97 43 84','2018-06-10 17:00:00','Sneezes
a lot, doesnt eat or play','High temperature and reddish throat','Feline
Flue','Medication','EM 97 43 84','CY 10 60 11 A',4);
```

```
insert into prescription values('FLFLU','Oreo','EM 97 43 84','2018-06-10
17:00:00','Paracetamol','Dayer','100mg','12/12h until next consult');
```

```
insert into prescription values('FLFLU','Oreo','EM 97 43 84','2018-06-10
17:00:00','Ibuprofen','Mistic','35mg','12/12h until next consult');
```

```
insert into consult_diagnosis values('FLFLU','Oreo','EM 97 43 84','2018-06-10
17:00:00');
```

```
insert into consult values('Oscar','NH 08 51 10','2018-04-14 15:00:00','Looks
fat and is unable to jump','Obese','high weight and fat','Medication, meal cut
and exercise','NH 08 51 10','EA 24 06 86',40.2);
```

```
insert into consult values('Oscar','NH 08 51 10','2018-08-14 15:00:00','Nothing
in particular','No indication of problems','Healthy','No action required','NH
08 51 10','EA 24 06 86',28.2);
```

```
insert into consult values('Odie','CB 46 16 95 B','2018-01-30
17:45:00','Nothing in particular','No indication of problems','Healthy','No
action required','CB 46 16 95 B','EA 24 06 86',10);
```

```
insert into consult values('Odie','CB 46 16 95 B','2017-11-12 18:00:00','Limps
on the left paw','X-ray shows broken limb','Broken paw','Cast','CB 46 16 95
B','CY 10 60 11 A',12);
```

```
insert into procedure_ values ('Odie','CB 46 16 95 B','2017-11-12
18:00:00',1,'paw X-Ray');
```

```
insert into radiography values ('Odie','CB 46 16 95 B','2017-11-12
18:00:00',1,'C:x-ray');
```



```
insert into consult_diagnosis values('BRKPAW','Odie','CB 46 16 95 B','2017-11-12 18:00:00');
```

```
insert into consult values('Garfield','CB 46 16 95 B','2018-01-31 18:00:00','Looks fat and is unable to jump','Obese','High weight and fat percentage','Medication, meal cut and exercise','NH 08 51 10','EA 24 06 86',6.2);
```

```
insert into consult_diagnosis values ('DIBT','Garfield','CB 46 16 95 B','2018-01-31 18:00:00');
```

```
insert into consult values ('Shiro', 'EA 24 06 86', '2018-01-30 17:45:00', 'Nothing in particular', 'No indication of problems', 'Healthy', 'No action required', 'EA 24 06 86', 'CY 10 60 11 A', 8);
```

```
insert into consult values('Luna','YL 71 92 45 B','2018-10-07 17:00:00','Tail injury due a fight','Lose a big quantity of blood','Broken tail','Medication','ZJ 17 07 20','EA 24 06 86',0.7);
```

```
insert into prescription values('BRKTAIL','Luna','YL 71 92 45 B','2018-10-07 17:00:00','Paracetamol','Dayer','100mg','12/12h until next consult');
```

```
insert into consult_diagnosis values('BRKTAIL','Luna','YL 71 92 45 B','2018-10-07 17:00:00');
```

```
insert into consult values('Luna','YL 71 92 45 B','2018-11-07 17:00:00','Check tail injury due a fight','Still lose a big quantity of blood','Broken tail','Medication','ZJ 17 07 20','EA 24 06 86',0.7);
```

```
insert into prescription values('BRKTAIL','Luna','YL 71 92 45 B','2018-11-07 17:00:00','Paracetamol','Dayer','100mg','12/12h until next consult');
```

```
insert into consult_diagnosis values('BRKTAIL','Luna','YL 71 92 45 B','2018-11-07 17:00:00');
```