

Instituto Superior Técnico

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

SIBD

3rd 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

1. A Web application using the database

Nota: A aplicação Web pode ser encontrada em:

<http://web.tecnico.ulisboa.pt/ist425480/checkanimal.php>

De seguida apresenta-se todo o código da aplicação, bem como uma explicação do código, acompanhado de um exemplo de teste com o animal:

- Nome: 'Marley' (cujo dono é Ethan Bennett – VAT: ZJ 37 07 20)
- VAT: 'ZJ 37 07 20'
- Porção de nome do dono: 'Ethan'

1.1. Task 1

checkanimal.php

```
<html>
    <body>
        <form action="check.php" method="post">
            <h3>Check if the animal exists</h3>
            <p>Animal name: <input type="text" name="animal_name"/></p>
            <p>    VAT: <input type="text" name="animal_vat"/></p>
            <p>    Owner: <input type="text" name="owner_name"/></p>

            <p><input type="submit" value="Submit"/></p>
        </form>
    </body>
</html>
```

Este ficheiro representa o formulário onde se coloca o nome do animal, o VAT do cliente e uma porção do nome do dono do animal.

Check if the animal exists

Animal name:

VAT:

Owner:

Figura 1 - Form inicial

check.php

```
<htm
l>

<body>
<?php
$host    =      "db.tecnico.ulisboa.pt";
$user    =      "ist425480";
$pass    =      "ipnr6905";
$dsn     =      "mysql:host=$host;dbname=$user";
Try
{
    $connection    =      new      PDO($dsn,      $user,  $pass);
}
catch(PDOException      $exception)
{
    echo("<p>Error:      ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
}
$animal_name    =      $_REQUEST[ 'animal_name' ];
//echo("<p>a$animal_name</p>\n");
$animal_vat    =      $_REQUEST[ 'animal_vat' ];
//echo("<p>$animal_vat</p>\n");
$owner_name    =      $_REQUEST[ 'owner_name' ];
//echo("<p>$owner_name</p>\n");
$check_client = $connection->prepare("SELECT * FROM client WHERE client.VAT =
:vat");
//$/check_client->debugDumpParams();
$check_client->bindParam(':vat', $animal_vat, PDO::PARAM_STR);
$check_client->execute();
$num_c = $check_client->rowCount();
echo("<p>$num_c clients retrieved</p>\n");
$check_animal = $connection->prepare("SELECT * from animal a where a.name =
:aname and a.VAT in (SELECT VAT from person p where p.name like
:owner_name)");
$check_animal->bindValue(':aname', $animal_name);
$check_animal->bindValue(':owner_name', "%{$owner_name}%");
$check_animal->execute();
$num_animals = $check_animal->rowCount();
echo("<p>$num_animals animals with the same VAT(owner) and name</p>\n");
$check_animal_query = $connection->prepare("SELECT distinct a.name as aname,
p.name as pname, p.VAT as vat, a.species_name as specie, a.gender as gd, a.age
as ag FROM animal a, client c, person p WHERE a.VAT = p.VAT and a.name =
:animal_name and p.name like :owner_name and exists (SELECT * FROM consult c1
WHERE c1.VAT_client = :vat and c1.name = a.name and c1.VAT_owner = p.VAT) ");

```

```

//$check_animal_query->debugDumpParams();
/**/
$check_animal_query->bindValue(':animal_name', $animal_name);
$check_animal_query->bindValue(':owner_name', "{$owner_name}");
$check_animal_query->bindValue(':vat', $animal_vat);
$check_animal_query->execute();
$num = $check_animal_query->rowCount();
echo("<p>$num animals accomplishing the requirements retrieved</p>\n");
//if $num >= 1
if($num >= 1){
    echo("<table border=\"1\">\n");
    echo("<tr><td>animal</td><td>person</td><td>vat</td><td>specie</td><td>
gender</td><td>age</td></tr>\n");
    foreach($check_animal_query as $row)
    {
        echo("<tr><td>");
        echo($row["aname"]);
        echo("</td><td>");
        echo($row["pname"]);
        echo("</td><td>");
        echo($row["vat"]);
        echo("</td><td>");
        echo($row["specie"]);
        echo("</td><td>");
        echo($row["gd"]);
        echo("</td><td>");
        echo($row["ag"]);
        echo("</td><td>");
        echo("a href=\"check_consults.php?aname=");
        echo($row['aname']);
        echo("&vat=");
        echo($row['vat']);
        echo("\">Check consults</a></td>\n");
        echo("</tr>\n");
    }
    echo("</table>\n");
}
//Option to add an animal
if($num == 0 && $num_animals == 0){
    echo("<form
action='addanimal.php?vat=$animal_vat&aname=$animal_name'
method='post'>\n <h3>No animal found. You can insert
$animal_name.</h3>");
    echo("<p>Animal name: $animal_name</p>");
    echo("<p>Owner VAT: $animal_vat</p>");
    echo("<p>specie:      <input type='text'      name='species_name' /></p>\n"

```

```

<p>colour: <input type='text' name='colour'/></p>\n<p>gender(M/F):<br>
<input type='text' name='gender' /></p><p>birth year: <input type='text' name='birth_year' /></p>\n<p>age: <input type='text' name='age' /></p><input type='submit' value='Add new animal' /></p>\n </form>");<br>
} else if($num == 0 && $num_animals > 0){<br>
    echo("<h3>Animal is already in the database but any consult with the VAT_client provided was found.</h3>\n");<br>
}<br>
//Option to go back to check the animals<br>
echo("<h3>Check the existence of another animal</h3>");<br>
echo("<form action='checkanimal.php' method='post'>\n </p>\n <p><input type='submit' value='Check another animal' /></p>\n </form>");<br>
$connection = null;?>
</body>
</html>

```

Neste ficheiro, com informação do VAT do cliente, do nome do animal e de uma porção do nome do dono do animal, provenientes do formulário inicial, fazem-se três verificações:

- Verifica-se se o cliente existe na base de dados (\$check_client);
- Verifica-se se o animal existe na base de dados, através do seu nome e do VAT associado à porção do nome do dono (\$check_animal);
- Verifica-se se existe algum animal na base de dados com o dono e que tenha tido pelo menos uma consulta cujo *VAT_client* seja igual ao VAT do cliente introduzido. (\$check_animal_query).

Em caso afirmativo apresenta-se os animais obtidos. Se não existir nenhum animal na base de dados com aquele nome e aquele dono, permite-se a inserção do mesmo através de um formulário. Se existir o animal na base de dados mas não houver nenhuma consulta associado ao VAT do cliente, informa-se o utilizador mas não se permite a inserção, tal como é pedido no enunciado, embora não faça muito sentido num contexto de uma aplicação real com todas as funcionalidades, dado que se um animal ainda não tiver nenhuma consulta nunca será apresentado nos resultados. Na figura 2 aparece o resultado da procura, com os campos do exemplo em teste.

1 clients retrieved

1 animals with the same VAT(owner) and name

1 animals accomplishing the requirements retrieved

animal	person	vat	specie	gender	age	
Marley	Ethan Bennett	ZJ 37 07 20	pug	M	6	Check consults

Check the existence of another animal

[Check another animal](#)

Figura 2 - Animal encontrado com as informações do formulário

Testou-se a introdução dos campos ‘Marley’, ‘ZJ 37 07 200’, ‘Ethan’, que apresenta um VAT inexistente, mas existe um animal com o nome ‘Marley’ e com um dono chamado ‘Ethan’, pelo que aparece a seguinte página:

0 clients retrieved

1 animals with the same VAT(owner) and name

0 animals accomplishing the requirements retrieved

Animal is already in the database but any consult with the VAT_client provided was found.

Check the existence of another animal

[Check another animal](#)

Figura 3 - Animal já existe, mas não existe consulta associada a esse VAT

A outra opção é introduzir um animal que não existe na base de dados. Para testar isto, introduziu-se os campos ‘Marley’, ‘ZJ 37 07 200’, ‘Etan’, resultando na seguinte página:

0 clients retrieved
 0 animals with the same VAT(owner) and name
 0 animals accomplishing the requirements retrieved

No animal found. You can insert Marley .

Animal name: Marley
 Owner VAT: ZJ 37 07 200

specie:

colour:

gender(M/F):

birth year:

age:

Check the existence of another animal

Figura 4 - Animal não existe na base de dados

A inserção do novo animal leva a aplicação a correr o código do ficheiro *addanimal.php*, que se apresenta de seguida.

addanimal.php

```
<html>
<body>
<?php
    $host    =      "db.tecnico.ulisboa.pt";
    $user    =      "ist425480";
    $pass    =      "ipnr6905";
    $dsn     =      "mysql:host=$host;dbname=$user";
    Try
    {
        $connection = new PDO($dsn, $user,
        $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error:      ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }
    //Information about the animal
    //$animal_vat = $_REQUEST['animal_vat'];

```

```

//$animal_name =      $_REQUEST['animal_name'];
$animal_vat    =      $_GET['vat'];
$animal_name   =      $_GET['aname'];
$species_name =      $_REQUEST['species_name'];
$colour       =      $_REQUEST['colour'];
$gender       =      $_REQUEST['gender'];
$birth_year   =      $_REQUEST['birth_year'];
$age          =      $_REQUEST['age']; //should convert this into
another integer variable?

//Check if the animal exists
$ccheck_animal = $connection->prepare("SELECT * from animal
WHERE VAT = :animal_vat and name = :animal_name");
$ccheck_animal->bindParam(':animal_name', $animal_name,
PDO::PARAM_STR);
$ccheck_animal->bindParam(':animal_vat', $animal_vat,
PDO::PARAM_STR);

$ccheck_animal->execute();
$num_a = $ccheck_animal->rowCount();
$num_a = $ccheck_animal->rowCount();
//echo("<p> $num_a retrieved.</p>");
if($num_a == 0 ){
    //Add a new animal.
    $add_animal = $connection->prepare("INSERT into animal
values (:animal_name, :animal_vat, :species_name, :colour, :gender,
:birth_year, :age)");
    $add_animal->bindParam(':animal_name', $animal_name,
PDO::PARAM_STR);
    $add_animal->bindParam(':animal_vat', $animal_vat,
PDO::PARAM_STR);
    $add_animal->bindParam(':species_name', $species_name,
PDO::PARAM_STR);
    $add_animal->bindParam(':colour', $colour,
PDO::PARAM_STR);
    $add_animal->bindParam(':gender', $gender,
PDO::PARAM_STR);
    $add_animal->bindParam(':birth_year', $birth_year,
PDO::PARAM_STR);
    $add_animal->bindParam(':age', $age, PDO::PARAM_STR);

    $add_animal->execute();

    echo("<h3> New animal added to the database</h3>");

} else {
    echo("<h3> This animal is already in the

```

```

        database</h3>");
    }

    echo("<p> Check consults of this animal</p>");
    echo("<form
        action='check_consults.php?aname=$animal_name&vat=$animal_vat'
        method='post'>\n    </p>\n    <p><input type='submit' value='Check
    consults' /></p>\n    </form>");

    echo("<h3>Check the existence of another animal</h3>");
    echo("<form    action='checkanimal.php'      method='post'>\n
</p>\n    <p><input      type='submit'  value='Go to homepage' /></p>\n
    </form>");

    $connection = null;
?>
</body>
</html>

```

No caso de o animal inserido não existir na base de dados é para assumir que o seu dono é o correspondente ao VAT introduzido. Embora a verificação de existência do animal seja feita no ficheiro check.php, neste ficheiro verifica-se que não existe nenhum animal com aquele nome e cujo dono foi o cliente introduzido. Caso não haja nenhuma correspondência, introduz-se o animal na base de dados. Embora na secção seguinte seja feito o trigger que calcula a idade do animal assim que se insira uma consulta, no formulário de adição de um novo animal permite-se a introdução do campo age. Após a inserção do animal, permite-se ao utilizador entrar na web page associada ao ficheiro check_consults.php, como se mostra na seguinte figura.

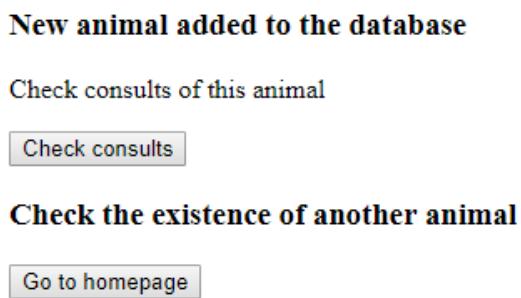


Figura 5 - Animal adicionado à base de dados

1.2. Task 2

check_consults.php

```
<html>
    <body>
        <?php
            $host    =      "db.ist.utl.pt";
            $user    =      "ist425480";
            $pass    =      "ipnr6905";
            $dsn     =      "mysql:host=$host;dbname=$user";
            Try
            {
                $connection    =      new      PDO($dsn,      $user, $pass);
            }
            catch(PDOException    $exception)
            {
                echo("<p>Error:      ");
                echo($exception->getMessage());
                echo("</p>");
                exit();
            }
            $aname   =      $_GET['aname'];
            $vat     =      $_GET['vat'];
            //echo("<p>Aname is: $aname</p>");
            //echo("<p>VAT is: $vat</p>");
            $show_animal_consults_query = $connection->prepare("SELECT c.name as aname,
c.VAT_owner as vatt, c.date_timestamp as date FROM animal a, consult c WHERE
a.VAT = c.VAT_owner and a.name = c.name and a.name =:aname and a.VAT =
:vat");
            $show_animal_consults_query->bindParam(':aname', $aname,
PDO::PARAM_STR);
            $show_animal_consults_query->bindParam(':vat', $vat, PDO::PARAM_STR);
            $show_animal_consults_query->execute();
            $num = $show_animal_consults_query->rowCount();
            //echo("<p>$num records retrieved:</p>\n");
            if($num >= 1){
                echo("<table border=\"1\">\n");
                echo("<tr><td>animal</td><td>vat</td><td>date</td></tr>\n");
                foreach($show_animal_consults_query as $row)
                {
                    echo("<tr><td>");
                    echo($row["aname"]);
                    echo("</td><td>");
                    echo($row["vatt"]);
                    echo("</td><td>");
                }
            }
        
```

```

        echo($row["date"]);
        //add link href
        echo("<td> <a href=\"consult_info.php?aname=\"");
        echo($row['aname']);
        echo("&vat=");
        echo($row['vatt']);
        echo("&date=");
        echo($row['date']);
        echo("\">Consult info</a></td>\n");
        echo("<td> <a href=\"enter_results.php?aname=\"");
        echo($row['aname']);
        echo("&vat=");
        echo($row['vatt']);
        echo("&date=");
        echo($row['date']);
        echo("\">enter results</a></td>\n");
        echo("</tr>\n");
    }
    echo("</table>\n");
}
$sql = "SELECT VAT from veterinary;";
$result = $connection->query($sql);
echo("<h3>Check the existence of another animal</h3>");
echo("<form action='checkanimal.php' method='post'>\n </p>\n<p><input type='submit' value='Go to homepage'/'></p>\n </form>");
//passar parametros atraves do formulario --> não está a passar corretamente
echo("<form action='addconsult.php?aname=$aname&vat=$vat' method='post'>\n <h3>Add a new consult to the database for ($aname, $vat)</h3> <h3>Consult information</h3>\n<p>date(YYYY-MM-DD hh:mm:ss):<input type='text' name='date'/'></p> <p>s: <input type='text' name='s'/'></p>\n <p>o: <input type='text' name='o'/'></p>\n <p>a: <input type='text' name='aa'/'></p><p>p:<input type='text' name='p'/'></p> ");
echo("\n<p>veterinary VET: <select name='VAT_vet'>");
foreach($result as $row)
{
    $VAT_vet = $row['VAT'];
    echo("<option value=\"$VAT_vet\">$VAT_vet</option>");
}
echo("</select></p>\n <p>weight: <input type='text' name='weight'/'></p>\n <p>diagnostic code: <input type='text' name='code'/'></p><p><input type='submit' value='Add new consult'/'></p>\n </form>");
$connection = null;
?>
```

```
</body>
</html>
```

Nesta *web page* apresenta-se as consultas do animal inserido, permitindo-se consultar a informação mais detalhada de cada uma, bem como inserir resultados associados a um *blood test* para cada consulta. Além disto permite-se a inserção de uma nova consulta. A inserção do veterinário é feita através de um *drop down* e não é permitido inserir em branco, dado que este é um campo obrigatório. É de notar que o diagnóstico também poderia ser feito da mesma forma, com a opção de um campo em branco, mas decidiu-se implementar através de um *input text*.

Na figura seguinte apresenta-se o resultado desta página referente ao exemplo inicial, o Marley.

animal	vat	date		
Marley	ZJ 37 07 20	2017-06-12 15:00:00	Consult info	enter results
Marley	ZJ 37 07 20	2017-09-08 08:00:00	Consult info	enter results
Marley	ZJ 37 07 20	2017-09-28 18:00:00	Consult info	enter results
Marley	ZJ 37 07 20	2018-03-14 10:00:00	Consult info	enter results

Check the existence of another animal

[Go to homepage](#)

Add a new consult to the database for (Marley, ZJ 37 07 20)

Consult information

date(YYYY-MM-DD hh:mm:ss):

s:

o:

a:

p:

veterinary VET:

weight:

diagnostic code:

[Add new consult](#)

Figura 6 - Lista de consultas do Marley

Preenchendo o formulário e carregando em “Add new consult”, a aplicação corre o código do ficheiro *addconsult.php*, que se descreve de seguida.

addconsult.php

```
<htm
1>
<body>
<?php
    $host  =      "db.tecnico.ulisboa.pt";
    $user  =      "ist425480";
```

```

$pass      =      "ipnr6905";
$dsn       =      "mysql:host=$host;dbname=$user";
Try
{
    $connection      =      new      PDO($dsn,      $user,
$pass);
}
catch(PDOException      $exception)
{
    echo("<p>Error:      ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
}
//Request parameters about new consult
$aname     =      $_GET['aname'];
$vat       =      $_GET['vat'];
//echo("<p>Animal name is: $aname");
//echo("<p>Vat is: $vat");
$date      =      $_REQUEST['date'];
$s         =      $_REQUEST['s'];
$o         =      $_REQUEST['o'];
$aa        =      $_REQUEST['aa'];
$p         =      $_REQUEST['p'];
$VAT_vet   =      $_REQUEST['VAT_vet'];
$weight    =      $_REQUEST['weight'];
$code      =      $_REQUEST['code'];
//Verify existence of the consult with that key
$check_consult = $connection->prepare("SELECT * from consult
where name = :name and VAT_owner = :vat_owner and date_timestamp = :dateeee");
$check_consult->bindParam(':name', $aname, PDO::PARAM_STR);
$check_consult->bindParam(':vat_owner', $vat, PDO::PARAM_STR);
$check_consult->bindParam(':dateeee', $date, PDO::PARAM_STR);
$check_consult->execute();
$num_c = $check_consult->rowCount();
if($num_c == 0){
    //Insert consult to the database
    $add_consult = $connection->prepare("INSERT into consult
values (:aname,:vat, :datee, :s, :o, :aa, :p, :VAT_client, :VAT_vet,
:weight)");
    $add_consult->bindParam(':aname', $aname,
PDO::PARAM_STR);
    $add_consult->bindParam(':vat', $vat, PDO::PARAM_STR);
    $add_consult->bindParam(':datee', $date,
PDO::PARAM_STR);
    $add_consult->bindParam(':s', $s, PDO::PARAM_STR);
}

```

```

        $add_consult->bindParam(':o', $o, PDO::PARAM_STR);
        $add_consult->bindParam(':aa', $aa, PDO::PARAM_STR);
        $add_consult->bindParam(':p', $p, PDO::PARAM_STR);
        $add_consult->bindParam(':VAT_client', $vat,
PDO::PARAM_STR);
        $add_consult->bindParam(':VAT_vet', $VAT_vet,
PDO::PARAM_STR);
        $add_consult->bindParam(':weight', $weight,
PDO::PARAM_STR);
        $add_consult->execute();
//-----
-----
$add_cd = $connection->prepare("INSERT into
consult_diagnosis values(:code, :aname, :vat, :datee)");
$add_cd->bindParam(':code', $code, PDO::PARAM_STR);
$add_cd->bindParam(':aname', $aname, PDO::PARAM_STR);
$add_cd->bindParam(':vat', $vat, PDO::PARAM_STR);
$add_cd->bindParam(':datee', $date, PDO::PARAM_STR);
$add_cd->execute();
echo("<h3> New consult added to the database</h3>");
} else {
    echo("<h3> Impossible to add this consult</h3>");
}

echo("<p> Check consults of this animal</p>");
echo("<form action='check_consults.php?aname=$aname&vat=$vat'
method='post'>\n </p>\n <p><input type='submit' value='Check
consults'/></p>\n </form>\"");
echo("<h3>Check the existence of another animal</h3>");
echo("<form action='checkanimal.php' method='post'>\n
</p>\n <p><input type='submit' value='Go to homepage'/'></p>\n </form>");

$connection = null;
?>
</body>
</html>

```

Neste ficheiro, através das informações da consulta a introduzir provenientes da página anterior, começa-se por verificar se já existe uma consulta com a mesma *primary key*. Em caso negativo, adiciona-se um registo na tabela *consult* e outro na tabela *consult_diagnosis*, que só é adicionado caso o código do diagnóstico não seja vazio.

Caso a adição não seja interrompida com nenhum erro apresenta-se a seguinte página ao utilizador:

New consult added to the database

Check consults of this animal

[Check consults](#)

Check the existence of another animal

[Go to homepage](#)

Figura 7 - Confirmação da inserção da consulta na base de dados

consult_info.php

```
<ht  
ml>  
    <body>  
        <?php  
        $host  =      "db.ist.utl.pt";  
        $user  =      "ist425480";  
        $pass  =      "ipnr6905";  
        $dsn   =      "mysql:host=$host;dbname=$user";  
        try  
        {  
            $connection = new PDO($dsn, $user, $pass);  
        }  
        catch(PDOException $exception)  
        {  
            echo("<p>Error:      </p>");  
            echo($exception->getMessage());  
            echo("</p>");  
            exit();  
        }  
        $aname =      $_GET['aname'];  
        $vat   =      $_GET['vat'];  
        $date  =      $_GET['date'];  
        $sql1 = $connection->prepare("SELECT c.name as name, c.VAT_owner as vatt,  
        c.date_timestamp as datee, c.VAT_client as vc, c.VAT_vet as vv, c.s as s, c.o  
        as o, c.aa as aa, c.p as p, c.weight as wg, a.gender as gd, a.age as ag from  
        animal a left outer join consult c on a.VAT = c.VAT_owner and a.name = c.name  
        left outer join consult_diagnosis cd on c.name = cd.name and c.VAT_owner =  
        cd.VAT_owner and c.date_timestamp = cd.date_timestamp left outer join  
        prescription p on p.code = cd.code and p.name = cd.name and p.VAT_owner =  
        cd.VAT_owner and p.date_timestamp = cd.date_timestamp where c.name = :aname and
```

```

c.VAT_owner = :vat and c.date_timestamp = :date");
// -----
$sql->bindParam(':aname', $aname, PDO::PARAM_STR);
$sql->bindParam(':vat', $vat, PDO::PARAM_STR);
$sql->bindParam(':date', $date, PDO::PARAM_STR);
$sql->execute();
$num = $sql->rowCount();
//echo("<p>$num records retrieved:</p>\n");
if($num >= 1){
    echo("<h3>Consult information</h3>");
    echo("<table border=\"1\">\n");

    echo("<tr><td>animal</td><td>vat</td><td>date</td><td>client</td><td>vet
</td><td>weight</td><td>gender</td><td>age</td></tr>\n");

    foreach($sql1 as $row)
    {
        echo("<tr><td>");
        echo($row["name"]);
        echo("</td><td>");
        echo($row["vatt"]);
        echo("</td><td>");
        echo($row["datee"]);
        echo("</td><td>");
        echo($row["vc"]);
        echo("</td><td>");
        echo($row["vv"]);
        echo("</td><td>");
        echo($row["wg"]);
        echo("</td><td>");
        echo($row["gd"]);
        echo("</td><td>");
        echo($row["ag"]);
        echo("</td>");
        echo("</tr>\n");
    }

    echo("</table>\n");
    $sql1->execute();
    echo("<h3>SOAP notes</h3>");
    echo("<table border=\"1\">\n");
    echo("<tr><td>subjective obs</td><td>objective
obs</td><td>assessment</td><td>plan</td></tr>\n");
    foreach($sql1 as $row){
        echo("<tr><td>");
        echo($row["s"]);

```

```

        echo("</td><td>");
        echo($row["o"]);
        echo("</td><td>");
        echo($row["aa"]);
        echo("</td><td>");
        echo($row["p"]);
        echo("</td>");
        echo("</tr>\n");
    }
    echo("</table>\n");
    //
    /*$sql11 = $connection->prepare("SELECT cd.code as cde ,
p.name_med as nm, p.lab as lb, p.dosage as dsg from consult_diagnosis cd,
prescription p where cd.code = p.code and cd.name = p.name and cd.VAT_owner =
p.VAT_owner and cd.date_timestamp = p.date_timestamp and cd.name = :aname and
cd.VAT_owner = :vat and cd.date_timestamp = :date");*/
    $sql11 = $connection->prepare("SELECT cd.code as cde from
consult_diagnosis cd where cd.name = :aname and cd.VAT_owner = :vat and
cd.date_timestamp = :date");
    $sql11->bindParam(':aname', $aname, PDO::PARAM_STR);
    $sql11->bindParam(':vat', $vat, PDO::PARAM_STR);
    $sql11->bindParam(':date', $date, PDO::PARAM_STR);
    $sql11->execute();
    $num_d = $sql11->rowCount();
    //echo("num_diagnosiss: $num_d\n");
    if($num_d >= 1){
        $sql11->bindParam(':aname', $aname, PDO::PARAM_STR);
        $sql11->bindParam(':vat', $vat, PDO::PARAM_STR);
        $sql11->bindParam(':date', $date, PDO::PARAM_STR);
        $sql11->execute();
        echo("<h3>Diagnosis</h3>");
        echo("<table border=\"1\">\n");
        echo("<tr><td>diagnosis</td></tr>\n");
        foreach($sql11 as $row){
            echo("<tr><td>");
            echo($row["cde"]);
            echo("</td>");
            echo("</tr>\n");
        }
        echo("</table>\n");
    }
    $sql111 = $connection->prepare("SELECT p.name_med as nm, p.lab
as lb, p.dosage as dsg from prescription p where p.name = :aname and
p.VAT_owner = :vat and p.date_timestamp = :date");
    $sql111->bindParam(':aname', $aname, PDO::PARAM_STR);

```

```

    $sql111->bindParam(':vat', $vat, PDO::PARAM_STR);
    $sql111->bindParam(':date', $date, PDO::PARAM_STR);
    $sql111->execute();
    $num_pres = $sql111->rowCount();
    //echo("num_prescriptions: $num_pres\n");

    if($num_pres >= 1){
        $sql111->bindParam(':aname', $aname, PDO::PARAM_STR);
        $sql111->bindParam(':vat', $vat, PDO::PARAM_STR);
        $sql111->bindParam(':date', $date, PDO::PARAM_STR);
        $sql111->execute();
        echo("<h3>Prescriptions</h3>");
        echo("<table border=\"1\">\n");

        echo("<tr><td>Medication</td><td>Lab</td><td>Dosage</td></tr>\n");
        foreach($sql111 as $row){
            echo("<tr><td>");
            echo($row["nm"]);
            echo("</td><td>");
            echo($row["lb"]);
            echo("</td><td>");
            echo($row["dsg"]);
            echo("</td>");
            echo("</tr>\n");
        }
        echo("</table>\n");
    }

    //print blood test result
    $search_num = $connection->prepare("SELECT num FROM
test_procedure WHERE name = :aname AND VAT_owner = :vat AND date_timestamp =
:date AND type = 'blood'");
    $search_num->bindParam(':aname', $aname, PDO::PARAM_STR);
    $search_num->bindParam(':vat', $vat, PDO::PARAM_STR);
    $search_num->bindParam(':date', $date, PDO::PARAM_STR);
    $search_num->execute();
    $num_p = $search_num->rowCount();
    $numm = $search_num->fetchColumn();
    if($num_p >= 1){
        $sql_results = $connection->prepare("SELECT
indicator_name, value FROM produced_indicator WHERE name = :aname AND VAT_owner
= :vat AND date_timestamp = :date AND num = :num");
        $sql_results->bindParam(':aname', $aname,
PDO::PARAM_STR);
        $sql_results->bindParam(':vat', $vat, PDO::PARAM_STR);

```

```

        $sql_results->bindParam(':date', $date, PDO::PARAM_STR);
        $sql_results->bindParam(':num', $numm, PDO::PARAM_STR);
        $sql_results->execute();
        echo("<h3>Blood test results of procedure with num =
$numm</h3>");
        echo("<table border=\"1\">\n");
        echo("<tr><td>indicator</td><td>indicator
count</td></tr>\n");
        foreach($sql_results as $row){
            echo("<tr><td>");
            echo($row["indicator_name"]);
            echo("</td><td>");
            echo($row["value"]);
            echo("</td>");
            echo("</tr>\n");
        }
        echo("</table>\n");
    }

} else {
    echo("<h3> This animal does not have any consult registered.
</h3>\n");
}
/*echo("<h3>Check the existence of another animal</h3>");
echo("<form action='checkanimal.php' method='post'>\n <p>\n<input type='submit' value='Go to homepage' /></p>\n </form>");*/
echo("<h3>Check again the consults</h3>");
echo("<form action='check_consults.php?aname=$aname&vat=$vat'
method='post'>\n <p>\n <p><input type='submit' value='Check
consults' /></p>\n </form>");

$connection = null;
?>
</body>
</html>
```

Nesta *web page*, através da *primary key* da consult, apresenta-se as informações de uma consulta em diversas tabelas. Na primeira, constam as informações que identificam a consulta, o VAT do cliente e do veterinário, o género, idade do animal e peso associado a essa consulta. Na tabela seguinte apresentam-se as *SOAP notes*. Caso a consulta tenha associada um diagnóstico e/ou uma prescrição de um medicamento, imprime-se o código do diagnóstico e informações sobre o medicamento prescrito. Se a consulta tiver associada um *blood test*, apresentam-se também os valores dos indicadores obtidos. No final encontra-se um botão que permite voltar à lista de consultas do animal.

Na figura seguinte apresenta-se a informação da última consulta do Marley que não teve nenhuma prescrição associada nem nenhum *blood test*.

Consult information							
animal	vat	date	client	vet	weight	gender	age
Marley	ZJ 37 07 20	2018-03-14 10:00:00	ZJ 37 07 20	EA 24 06 86	40.20	M	6

subjective obs	objective obs	assessment	plan
Looks fat and is unable to jump	Obesity	high weight	Medication, meal cut and excercise

Diagnosis	
diagnosis	
DIBT	

Check again the consults	
<input type="button" value="Check consults"/>	

Figura 8 - Informação da última consulta do Marley

No caso da 2^a consulta da lista, foi feito um teste ao sangue cujos indicadores foram *Creatinine level* e *Kidney Enzime* e houve uma prescrição associada, pelo que a informação apresentada é maior à anterior:

Consult information							
animal	vat	date	client	vet	weight	gender	age
Marley	ZJ 37 07 20	2017-09-08 08:00:00	ZJ 37 07 20	CY 10 60 11 A	30.80	M	6

subjective obs	objective obs	assessment	plan
Doesnt want to eat or play	High level of enzymes	Kidney failure	Medication and admission

Diagnosis	
diagnosis	
KDNFLR	

Prescriptions		
Medication	Lab	Dosage
Adaytidah polipstiso	Dayer	50mg

Blood test results of procedure with num = 1	
indicator	indicator count
Creatinine level	2
Kidney Enzime	2

Check again the consults	
<input type="button" value="Check consults"/>	

Figura 9 - Apresentação da informação de uma consulta com todos os parâmetros a apresentar

1.3. Task 3

enter_results.php

```
<html>
    <body>
        <?php
            $host = "db.tecnico.ulisboa.pt";
            $user = "ist425480";
            $pass = "ipnr6905";
            $dsn = "mysql:host=$host;dbname=$user";
            try
            {
                $connection = new PDO($dsn, $user, $pass);
            }
            catch(PDOException $exception)
            {
                echo("<p>Error:      ");
                echo($exception->getMessage());
                echo("</p>");
                exit();
            }
            $aname = $_GET['aname'];
            //echo("<p> $aname</p>\n");
            $vat = $_GET['vat'];
            //echo("<p> $vat</p>\n");
            $date = $_GET['date'];
            //echo("<p> $date</p>\n");
            $sql = "SELECT VAT from assistant";
            $result = $connection->query($sql);
            //Enter results of a blood test
            //VAT_assistant, values of indicators: white blood cell count, number of neutrophils, number of lymphocites, number of monocytes
            echo("<form action='addrresults.php?aname=$aname&vat=$vat&date=$date' method='post'>\n <h2>Enter results</h2> \n<p>number of neutrophils: <input type='text' name='neutrophils'/></p> <p>number of lymphocits: <input type='text' name='lymphocits'/></p>\n <p>number of monocytes: <input type='text' name='monocytes'/></p>\n<p>number of white blood cells: <input type='text' name='white_blood_cell'/></p>\n<p>assistant VAT: <select name='VAT_assistant'> <option value=\"$VAT_assistant\">$VAT_assistant</option>");");
            foreach($result as $row)
            {
                $VAT_assistant = $row['VAT'];
                echo("<option value=\"$VAT_assistant\">$VAT_assistant</option>");
```

```

        }
echo("</select></p>\n<p>num: <input type='text'
      name='num' /></p>\n<p><input type='submit' value='Enter
results' /></p>\n  </form>");

$connection    =      null;
?>
</body>
</html>

```

Nesta web page, o utilizador pode inserir resultados de um blood test associados a uma consulta. Para isso tem de inserir o número das células obtidas, selecionar um dos VATs disponíveis no *drop down*, ou selecionar a opção em branco, caso não tenha sido realizado por nenhum assistente e introduzir o *num* correspondente. Como visto na figura 8, a última consulta do Marley não teve associado nenhum teste ao sangue. A seguinte figura ilustra a introdução de um teste ao sangue nessa consulta.

Enter results

number of neutrophils: 2000

number of lymphocits: 2000

number of monocytes: 2500

number of white blood cells: 6500

assistant VAT: CB 46 16 95 B ▾

num: 1

Enter results

Figura 10 - Adição de um blood test à última consulta do Marley

Ao clicar no botão para introduzir os resultados, o utilizador é redirecionado para a web page presente na figura 11, associada ao ficheiro addresults.php, que se descreve de seguida.

Results entered successfully!

Check the consults

[Check consults](#)

Check the existence of another animal

[Go to homepage](#)

Figura 11 - Confirmação da inserção dos resultados do blood test na base de dados

addresults.php

```
<html>
<body>
<?php
    $host    =      "db.tecnico.ulisboa.pt";
    $user    =      "ist425480";
    $pass    =      "ipnr6905";
    $dsn     =      "mysql:host=$host;dbname=$user";
    try
    {
        $connection = new PDO($dsn, $user,
$pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error:      ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }
    //Request parameters about new consult
    $aname   =      $_GET['aname'];
    $vat     =      $_GET['vat'];
    $date    =      $_GET['date'];
    $neutrophils= $_REQUEST['neutrophils'];
    $lymphocits = $_REQUEST['lymphocits'];
    $monocytes = $_REQUEST['monocytes'];
    $white_blood_cells = $_REQUEST['white_blood_cell'];
    $VAT_assistant = $_REQUEST['VAT_assistant'];
    $num     =      $_REQUEST['num'];
    //-----
    try{
        $connection->beginTransaction();
        $add_participation = $connection->prepare("INSERT Into
participation values (:aname, :vat, :datee, :VAT_assistant)");

        $add_participation->bindParam(':aname', $aname,
PDO::PARAM_STR);
        $add_participation->bindParam(':vat', $vat,
PDO::PARAM_STR);
        $add_participation->bindParam(':datee', $date,
PDO::PARAM_STR);
        $add_participation->bindParam(':VAT_assistant',
$VAT_assistant, PDO::PARAM_STR);
    }
```

```

    $add_participation->execute();

    //--
    $add_procedure = $connection->prepare("INSERT into
procedure_ values (:aname, :vat,:datee, :num, 'blood test')");
    $add_procedure->bindParam(':aname', $aname,
PDO::PARAM_STR);
    $add_procedure->bindParam(':vat', $vat,
PDO::PARAM_STR);
    $add_procedure->bindParam(':datee', $date,
PDO::PARAM_STR);
    $add_procedure->bindParam(':num', $num,
PDO::PARAM_INT);
    $add_procedure->execute();

    $add_performed = $connection->prepare("INSERT into
performed values (:aname, :vat,:datee, :num, :VAT_assistant)");
    $add_performed->bindParam(':aname', $aname,
PDO::PARAM_STR);
    $add_performed->bindParam(':vat', $vat,
PDO::PARAM_STR);
    $add_performed->bindParam(':datee', $date,
PDO::PARAM_STR);
    $add_performed->bindParam(':num', $num,
PDO::PARAM_INT);
    $add_performed->bindParam(':VAT_assistant',
$VAT_assistant, PDO::PARAM_STR);
    $add_performed->execute();

    $add_testprocedure = $connection->prepare("INSERT into
test_procedure values (:aname, :vat,:datee, :num, 'blood')");
    $add_testprocedure->bindParam(':aname', $aname,
PDO::PARAM_STR);
    $add_testprocedure->bindParam(':vat', $vat,
PDO::PARAM_STR);
    $add_testprocedure->bindParam(':datee', $date,
PDO::PARAM_STR);
    $add_testprocedure->bindParam(':num', $num,
PDO::PARAM_STR);

    $add_testprocedure->execute();
    $add_neutrophils = $connection->prepare("INSERT into
produced_indicator values (:aname, :vat,:datee,
:num,'neutrophil',:neutrophils')");
    $add_neutrophils->bindParam(':aname', $aname,
PDO::PARAM_STR);

```

```

                $add_neutrophils->bindParam(':vat', $vat,
PDO::PARAM_STR);
                $add_neutrophils->bindParam(':datee', $date,
PDO::PARAM_STR);
                $add_neutrophils->bindParam(':num', $num,
PDO::PARAM_INT);
                $add_neutrophils->bindParam(':neutrophils',
$num, PDO::PARAM_STR);
                $add_neutrophils->execute();
                //--
                $add_lymphocits = $connection->prepare("INSERT into
produced_indicator values (:aname, :vat,:datee,
:num,'lymphocit',:lymphocits)");
                $add_lymphocits->bindParam(':aname', $aname,
PDO::PARAM_STR);
                $add_lymphocits->bindParam(':vat', $vat,
PDO::PARAM_STR);
                $add_lymphocits->bindParam(':datee', $date,
PDO::PARAM_STR);
                $add_lymphocits->bindParam(':num', $num,
PDO::PARAM_INT);
                $add_lymphocits->bindParam(':lymphocits', $lymphocits,
PDO::PARAM_STR);
                $add_lymphocits->execute();
                //--
                $add_monocytes = $connection->prepare("INSERT into
produced_indicator values (:aname, :vat,:datee,
:num,'monocyt',:monocytes)");
                $add_monocytes->bindParam(':aname', $aname,
PDO::PARAM_STR);
                $add_monocytes->bindParam(':vat', $vat,
PDO::PARAM_STR);
                $add_monocytes->bindParam(':datee', $date,
PDO::PARAM_STR);
                $add_monocytes->bindParam(':num', $num,
PDO::PARAM_INT);
                $add_monocytes->bindParam(':monocytes', $monocytes,
PDO::PARAM_STR);
                $add_monocytes->execute();
                //--
                $add_white_blood_cells = $connection->prepare("INSERT
into produced_indicator values (:aname, :vat,:datee, :num,'white blood
cell',:white_blood_cells')");
                $add_white_blood_cells->bindParam(':aname', $aname,
PDO::PARAM_STR);
                $add_white_blood_cells->bindParam(':vat', $vat,

```

```

    PDO::PARAM_STR);
        $add_white_blood_cells->bindParam(':datee', $date,
    PDO::PARAM_STR);
        $add_white_blood_cells->bindParam(':num', $num,
    PDO::PARAM_INT);
        $add_white_blood_cells->bindParam(':white_blood_cells',
$white_blood_cells, PDO::PARAM_STR);
        $add_white_blood_cells->execute();
        $connection->commit();
        echo("<h3>Results entered successfully!</h3>");
    } catch(Exception $e){
        echo $e->getMessage();
        $connection->rollBack();
    }
    echo("<h3>Check the consults</h3>");
    echo("<form
        action='check_consults.php?aname=$aname&vat=$vat' method='post'>\n
    </p>\n    <p><input type='submit' value='Check consults' /></p>\n
    </form>");  

        echo("<h3>Check the existence of another animal</h3>");  

        echo("<form action='checkanimal.php' method='post'>\n
    </p>\n    <p><input type='submit' value='Go to homepage' /></p>\n
    </form>");  

        $connection = null;
    ?>
</body>
</html>

```

Este ficheiro é responsável por adicionar na base de dados, os dados relacionados ao *blood test*. A inserção dos dados é feita no contexto de uma única transação dentro de um bloco `try{catch{}}`. Se ocorrer algum erro na inserção, esta fica sem efeito através de um *rollBack*. Além da inserção dos indicadores resultantes do teste na tabela *produced_indicator*, adiciona-se um registo para as seguintes tabelas: *participation*, *procedure_*, *performed* e *test_procedure*. É de referir que se teve de alterar o ficheiro de população da base de dados entregue na parte 2 para adicionar os indicadores especificados no enunciado na tabela *indicator* de modo a evitar problemas de *foreign keys*.

Ao voltar a verificar a informação sobre a última consulta do Marley, verifica-se que já é apresentada informação sobre um teste ao sangue, como se demonstra na seguinte figura.

Consult information

animal	vat	date	client	vet	weight	gender	age
Marley	ZJ 37 07 20	2018-03-14 10:00:00	ZJ 37 07 20	EA 24 06 86	40.20	M	6

SOAP notes

subjective obs	objective obs	assessment	plan
Looks fat and is unable to jump	Obesity	high weight	Medication, meal cut and excercise

Diagnosis

diagnosis
DIBT

Blood test results of procedure with num = 1

indicator	indicator count
lymphocit	2000
monocyt	2500
neutrophil	2000
white blood cell	6500

Check again the consults

[Check consults](#)

Figura 12 - Informação da última consulta do Marley após inserção dos resultados do blood test

2. Functions, Triggers and Stored Procedures

Antes de testar os códigos ou executá-los novamente, usamos a função DROP para garantir que problemas não esperados ocorram:

```
DROP TRIGGER update_age;
DROP TRIGGER already_vet_insert;
DROP TRIGGER already_assistant_insert;
DROP TRIGGER already_assistant_update;
DROP TRIGGER already_veterinary_update;
DROP TRIGGER cell_n;
DROP TRIGGER cell_n_update;
DROP FUNCTION n_consults;
DROP PROCEDURE ref_values;
```

2.1. Task 1

O seguinte código faz a atualização da idade do animal quando uma nova consulta do animal é inserida na base de dados. Usamos o ano da atual data [YEAR(NOW())] para o cálculo.

```
delimiter //
CREATE TRIGGER update_age AFTER INSERT ON consult
FOR EACH ROW
BEGIN
update animal
set age = YEAR(NOW()) - animal.birth_year;
END; //
delimiter ;
```

2.2. Task 2

Para verificar se um indivíduo não pode ser simultaneamente um médico veterinário e um assistente no hospital em questão, foi necessário procurar esses na atual base de dados dos veterinários e assistentes existentes. Em caso positivo o resultado dessa procura, o código *SIGNAL SQLSTATE VALUE '45000'* informa uma ocorrência de erro de condição, e então imprime a mensagem a informar que o indivíduo já é um veterinário ou um assistente, de acordo com o caso, conforme o código abaixo. Essas verificações são feitas nas tentativas de inserção ou atualização de dados na base de dados.

```
delimiter //
CREATE TRIGGER already_vet_insert BEFORE INSERT ON assistant
FOR EACH ROW
```

```

BEGIN
IF NEW.VAT IN (
SELECT v.VAT FROM veterinary v)
THEN SIGNAL SQLSTATE VALUE '45000'
SET MESSAGE_TEXT = 'This veterinary already exist';
END IF;
END;
// 
delimiter ;

delimiter //
CREATE TRIGGER already_assistant_insert BEFORE INSERT ON veterinary
FOR EACH ROW
BEGIN
IF NEW.VAT IN (
SELECT a.VAT FROM assistant a)
THEN SIGNAL SQLSTATE VALUE '45000'
SET MESSAGE_TEXT = 'This assistant already exist';
END IF;
END ;
// 
delimiter ;

delimiter //
CREATE TRIGGER already_assistant_update BEFORE UPDATE ON veterinary
FOR EACH ROW
BEGIN
IF NEW.VAT IN (
SELECT a.VAT FROM assistant a)
THEN SIGNAL SQLSTATE VALUE '45000'
SET MESSAGE_TEXT = 'This assistant already exist';
END IF;
END ;
// 
delimiter ;

delimiter //
CREATE TRIGGER already_veterinary_update BEFORE UPDATE ON assistant
FOR EACH ROW
BEGIN
IF NEW.VAT IN (
SELECT v.VAT FROM veterinary v)
THEN SIGNAL SQLSTATE VALUE '45000'
SET MESSAGE_TEXT = 'This veterinary already exist';
END IF;
END ;

```

```
//  
delimiter ;
```

A tabela abaixo mostra a atual lista de veterinário registrados na base de dados do hospital.

VAT	specialization	bio
EA 24 06 86	General Practicioneer	Pop culture geek. Vet. Unapologetic social media lover. Travel fanatic.
CY 10 60 11 A	Surgery	Coffee buff. Web enthusiast. Surgery Vet. Gamer. Avid organizer.

Na tentativa de inserir o veterinário associado ao VAT 'CY 10 60 11 A' na tabela de assistentes, o trigger impede informando que o veterinário já existe, como mostra o print abaixo:

```
MySQL [ist191410]> insert into assistant values('CY 10 60 11 A');  
ERROR 1644 (45000): This veterinary already exist
```

O inverso também é valido. No print abaixo tem-se uma tentativa de inserir um 'assistant', por exemplo o 'SM 62 10 29 D', na tabela 'veterinary':

```
MySQL [ist191410]> insert into veterinary values('SM 62 10 29 D', 'Surgery', 'Coffee buff. Web enthusiast. Surgery Vet. Gamer. Avid organizer.');//  
ERROR 1644 (45000): This assistant already exist
```

2.3. Task 3

Para ter certeza que diferentes indivíduos não podem ter o mesmo número do telemóvel, é necessário verificar, sempre que ocorrer uma tentativa de atualização ou inserção de um novo número, se este já existe na base de dados, na tabela *phone_number*. Nesse caso, o código informa uma ocorrência de erro de condição, e imprime uma mensagem.

```
delimiter //  
CREATE TRIGGER cell_n BEFORE INSERT ON phone_number  
FOR EACH ROW  
BEGIN  
IF NEW.phone IN (  
SELECT pn.phone FROM phone_number pn)  
THEN SIGNAL SQLSTATE VALUE '45000'  
SET MESSAGE_TEXT = 'This phone number already exist';  
END IF;  
END ;  
//  
delimiter ;
```

```
delimiter //  
CREATE TRIGGER cell_n_update BEFORE UPDATE ON phone_number  
FOR EACH ROW  
BEGIN  
IF NEW.phone IN (  
SELECT pn.phone FROM phone_number pn)
```

```

THEN SIGNAL SQLSTATE VALUE '45000'
SET MESSAGE_TEXT = 'This phone number already exists';
END IF;
END ;
//
delimiter ;

```

Na tabela a seguir, tem-se todos os números de telemóvel cadastrado na base de dados:

```

MySQL [ist191410]> select * from phone_number;
+-----+-----+
| VAT      | phone    |
+-----+-----+
| CB 46 16 95 B | 070 2149 6062 |
| CB 46 92 95 B | 070 2179 6060 |
| CY 10 60 11 A | 079 4132 9210 |
| EA 24 06 86   | 070 2699 7334 |
| EM 97 43 84   | 077 7080 5592 |
| GS 14 83 38 A | 079 7140 4420 |
| NH 08 51 10   | 078 6008 1591 |
| SM 62 10 29 D | 070 7514 9424 |
| WT 72 72 72 A | 077 7408 9030 |
| YL 71 92 45 B | 079 1080 0169 |
| ZA 16 10 97   | 078 6911 1109 |
| ZJ 17 07 20   | 079 7422 0004 |
| ZJ 37 07 20   | 079 7422 0554 |
+-----+-----+

```

No print a seguir, mostra-se uma tentativa de inserir um '*phone number*' já existente. O mesmo acontece se tentar atualizar o número de uma '*person*' já existente.

```

MySQL [ist191410]> insert into phone_number values('CB 46 16 95 B','070 2149 6062');
ERROR 1644 (45000): This phone number already exists

```

2.4. Task 4

Um animal é unicamente identificado com as *keys* '*animal_name*' e '*animal_vat*', isso quer dizer que para computar o número total de consultas de um animal específico dentro de um ano dado, são necessários 3 parâmetros de entradas (*animal_name*, *animal_vat* e *year*). A função criada para este efeito é a seguinte:

```

delimiter //
create function n_consults(animal_name varchar(255), animal_vat varchar(255), c_year year )
returns integer
begin
declare num integer;
select count(date_timestamp) into num
from consult c
where c.name = animal_name
and c.VAT_owner = animal_vat
and year(c.date_timestamp) = c_year;
return num;
end
//
```

```
delimiter ;
```

Para testar a função, pode-se usar como exemplo o desejo de ter a tabela de todos os animais que tiveram mais de 2 consultas no ano de 2018. Então, tem-se o print a baixo:

```
MySQL [ist191410]> select distinct name,VAT_owner from consult where n_consults(name,VAT_owner, 2018) >2;
+-----+-----+
| name | VAT_owner |
+-----+-----+
| Luna | YL 71 92 45 B |
| Tom  | ZJ 17 07 20 |
+-----+-----+
```

Dado essa tabela (consultas em 2018), para confirmação de resultado:

```
Database changed
MySQL [ist191410]> select name, date_timestamp from consult where year(date_timestamp) = 2018;
+-----+-----+
| name      | date_timestamp      |
+-----+-----+
| Odie      | 2018-01-30 17:45:00 |
| Jerry     | 2018-01-05 10:00:00 |
| Jerry     | 2018-01-25 16:00:00 |
| Shiro     | 2018-01-30 17:45:00 |
| Oreo      | 2018-06-10 17:00:00 |
| Garfield  | 2018-01-31 18:00:00 |
| Oscar     | 2018-04-14 15:00:00 |
| Oscar     | 2018-08-14 15:00:00 |
| Francis   | 2018-06-08 10:00:00 |
| Minnie    | 2018-06-08 10:30:00 |
| Luna      | 2018-05-12 14:00:00 |
| Luna      | 2018-08-26 10:30:00 |
| Bob       | 2018-02-25 17:00:00 |
| Bob       | 2018-05-30 14:00:00 |
| Luna      | 2018-10-07 17:00:00 |
| Luna      | 2018-11-07 17:00:00 |
| Tom       | 2018-01-05 09:00:00 |
| Tom       | 2018-02-24 14:00:00 |
| Tom       | 2018-12-05 09:00:00 |
| Marley    | 2018-03-14 10:00:00 |
+-----+-----+
```

2.5. Task 5

Mudando o valor de referência associado a todos os indicadores que são medidos em miligramas modifica duas tabelas na base de dados. A primeira é a ‘produced_indicator’ onde a coluna ‘value’ medida em ‘milligrams’ deve ser modificada dividindo o valor por 10, o que resulta em ‘centigrams’. Por outro lado, a segunda tabela a alterar é a ‘indicator’, onde temos que ajustar o ‘reference_value’ para 10x menor e trocar as unidades para ‘centigrams’. Isto é aplicado apenas para os casos em que o ‘indicator’ é medido em miligramas.

```
delimiter //
create procedure ref_values()
begin

update produced_indicator pi
set pi.value = pi.value / 10,
pi.name = pi.name,
```

```

pi.VAT_owner = pi.VAT_owner,
pi.date_timestamp = pi.date_timestamp,
pi.num = pi.num
where pi.indicator_name in (
select i.name from indicator i
where i.units = 'milligrams');

update indicator i
set i.reference_value = i.reference_value / 10, i.units = 'centigrams'
where i.units = 'milligrams';
end

delimeter ;

```

Aqui tem-se a tabela '*indicator*' com alguns deles medido em miligramas. E na sequência a tabela '*produced_indicator*' na qual tem referência com a primeira.

```

MySQL [ist191410]> select * from indicator;
+-----+-----+-----+
| name | reference_value | units | description |
+-----+-----+-----+
| Carnine Enzime | 250.00 | picograms | Level of enzimes of Carnine |
| Kidney Enzime | 115.00 | milligrams | Level of enzimes of kidney function |
| Kidney Enzime 2 | 150.00 | milligrams | Level of enzimes of kidney function 2 |
| Kidney Enzime 3 | 50.00 | milligrams | Level of enzimes of kidney function 3 |
| Poison | 0.00 | picograms | Amount of poison |
+-----+-----+-----+
+-----+-----+-----+-----+
| name | VAT_owner | date_timestamp | num | indicator_name | value |
+-----+-----+-----+-----+
| Jerry | CB 46 92 95 B | 2018-01-25 16:00:00 | 4 | Poison | 9.00 |
| Marley | ZJ 37 07 20 | 2017-09-08 08:00:00 | 1 | Kidney Enzime | 2.00 |
+-----+-----+-----+-----+

```

Utilizando o comando que invoca o *procedure* (*call ref_values;*), tem-se as seguintes alterações:

```

+-----+-----+-----+-----+
| name | reference_value | units | description |
+-----+-----+-----+
| Carnine Enzime | 250.00 | picograms | Level of enzimes of Carnine |
| Kidney Enzime | 11.50 | centigrams | Level of enzimes of kidney function |
| Kidney Enzime 2 | 15.00 | centigrams | Level of enzimes of kidney function 2 |
| Kidney Enzime 3 | 5.00 | centigrams | Level of enzimes of kidney function 3 |
| Poison | 0.00 | picograms | Amount of poison |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| name | VAT_owner | date_timestamp | num | indicator_name | value |
+-----+-----+-----+-----+
| Jerry | CB 46 92 95 B | 2018-01-25 16:00:00 | 4 | Poison | 9.00 |
| Marley | ZJ 37 07 20 | 2017-09-08 08:00:00 | 1 | Kidney Enzime | 0.20 |
+-----+-----+-----+-----+

```