

INSTITUTO SUPERIOR TÉCNICO

MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE
COMPUTADORES

SISTEMAS DE INFORMAÇÃO E BASES DE DADOS

Using the Database

Work by:
Bernardo Gonilho
Diogo Moura
Diogo Alves

Number:
84016
86976
86980

Turn L09 - Grupo 11

2019/2020

Introduction

The goal of this part of the project is to use and manipulate the database previously created in the part 2 using PHP and HTML, by developing a Web-based application. This Web-application must execute the tasks specified in the assignment, which are detailed below.

The Web-application can be accessed through this link:

http://web.tecnico.ulisboa.pt/ist186980/sibdprojeto/insert_client_info.php

SQL Triggers, Functions and Procedures

In this section we create triggers to ensure cardinality and disjointness. We also create a function that returns a count of the appointments where the client missed the consultation and a procedure that automates the process of updating the doctors' salary. These triggers must not only precede and check the insertion process, but also the update process. In SQL, the functions may be useful if we constantly repeat a task, in order to reduce the amount of code, and the procedures if one wants to automate changes in the database, such as updates and insertions, since the functions do not allow those.

Trigger to update the age of the clients according to the birth date and the current date. The trigger fires whenever a new appointment for a client is inserted into the database.

```
delimiter $$
create trigger update_clients_age after insert on appointment
for each row
begin
    update client set age=floor(datediff(now(),client.birth_date)/365)
    where client.VAT= new.VAT_client;
end$$
delimiter ;
```

Triggers to ensure that an individual that is a receptionist or a nurse at the clinic cannot simultaneously be a doctor.

```
delimiter $$
create trigger ensure_nurse_i before insert on nurse
for each row
begin
    if new.VAT in (select VAT from doctor)
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Already exists a doctor
        with the same VAT!';
    end if;
end$$
delimiter ;

delimiter $$
create trigger ensure_nurse_u before update on nurse
for each row
begin
```

```
        if new.VAT in (select VAT from doctor)
        then
            signal sqlstate '45000' set MESSAGE_TEXT = 'Already exists a doctor
            with the same VAT!';
        end if;
    end$$
delimiter ;

delimiter $$
create trigger ensure_receptionist_i before insert on receptionist
for each row
begin
    if new.VAT in (select VAT from doctor)
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Already exists a doctor
        with the same VAT!';
    end if;
end$$
delimiter ;

delimiter $$
create trigger ensure_receptionist_u before update on receptionist
for each row
begin
    if new.VAT in (select VAT from doctor)
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Already exists a doctor
        with the same VAT!';
    end if;
end$$
delimiter ;

delimiter $$
create trigger ensure_doctor_i before insert on doctor
for each row
begin
    if new.VAT in (select VAT from receptionist) or new.VAT in (select VAT from nurse)
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Already exists a nurse
        or a receptionist with the same VAT!';
    end if;
end$$
delimiter ;

delimiter $$
create trigger ensure_doctor_u before update on doctor
for each row
```

```
begin
    if new.VAT in (select VAT from receptionist) or new.VAT in (select VAT from nurse)
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Already exists a nurse
        or a receptionist with the same VAT!';
    end if;
end$$
delimiter ;
```

Triggers to ensure that doctors cannot simultaneously be trainees and permanent staff.

```
delimiter $$
create trigger ensure_permanent_doctor_i before insert on permanent_doctor
for each row
begin
    if new.VAT in (select VAT from trainee_doctor)
    then
        signal sqlstate '44000' set MESSAGE_TEXT = 'Already exists a
        Trainee Doctor with the same VAT!';
    end if;
end$$
delimiter ;
```

```
delimiter $$
create trigger ensure_permanent_doctor_u before update on permanent_doctor
for each row
begin
    if new.VAT in (select VAT from trainee_doctor)
    then
        signal sqlstate '44000' set MESSAGE_TEXT = 'Already exists a
        Trainee Doctor with the same VAT!';
    end if;
end$$
delimiter ;
```

```
delimiter $$
create trigger ensure_trainee_doctor_i before insert on trainee_doctor
for each row
begin
    if new.VAT in (select VAT from permanent_doctor)
    then
        signal sqlstate '44001' set MESSAGE_TEXT = 'Already exists a
        Permanent Doctor with the same VAT!';
    elseif new.supervisor in (select VAT from trainee_doctor)
    then
        signal sqlstate '44001' set MESSAGE_TEXT = 'A supervisor cannot
        be a trainee!';
    end if;
end$$
```

```
delimiter ;

delimiter $$
create trigger ensure_trainee_doctor_u before update on trainee_doctor
for each row
begin
    if new.VAT in (select VAT from permanent_doctor)
    then
        signal sqlstate '44001' set MESSAGE_TEXT = 'Already exists a
Permanent Doctor with the same VAT!';
    elseif new.supervisor in (select VAT from trainee_doctor)
    then
        signal sqlstate '44001' set MESSAGE_TEXT = 'A supervisor cannot
be a trainee!';
    end if;
end$$
delimiter ;
```

Triggers to ensure that different individuals (doctors or clients) cannot have the same phone number.

```
delimiter $$
create trigger ensure_phone_number_client_i before insert on phone_number_client
for each row
begin
    if new.phone in (select phone from phone_number_employee natural join doctor)
    then
        signal sqlstate '44003' set MESSAGE_TEXT = 'Already exists a
doctor with the same phone number!';
    elseif new.phone in (select phone from phone_number_client)
    then
        signal sqlstate '44003' set MESSAGE_TEXT = 'Already exists a
client with the same phone number!';
    end if;
end$$
delimiter ;
```

```
delimiter $$
create trigger ensure_phone_number_client_u before update on phone_number_client
for each row
begin
    if new.phone in (select phone from phone_number_employee natural join doctor)
    then
        signal sqlstate '44003' set MESSAGE_TEXT = 'Already exists a
doctor with the same phone number!';
    elseif new.phone in (select phone from phone_number_client)
    then
        signal sqlstate '44003' set MESSAGE_TEXT = 'Already exists a
client with the same phone number!';
    end if;
end$$
delimiter ;
```

```
        end if;
end$$
delimiter ;

delimiter $$
create trigger ensure_phone_number_doctor_i before insert on phone_number_employee
for each row
begin
    if new.VAT in (select VAT from doctor) then
        if new.phone in (select phone from phone_number_client) then
            signal sqlstate '44004' set MESSAGE_TEXT = 'Already exists a
            client with the same phone number!';
        elseif new.VAT in (select VAT from phone_number_employee natural join doctor) then
            signal sqlstate '44004' set MESSAGE_TEXT = 'Already exists
            another doctor with the same phone number!';
        end if;
    end if;
end if;
end$$
delimiter ;

delimiter $$
create trigger ensure_phone_number_doctor_u before update on phone_number_employee
for each row
begin
    if new.VAT in (select VAT from doctor) then
        if new.phone in (select phone from phone_number_client) then
            signal sqlstate '44004' set MESSAGE_TEXT = 'Already exists
            a client with the same phone number!';
        elseif new.VAT in (select VAT from phone_number_employee natural join doctor) then
            signal sqlstate '44004' set MESSAGE_TEXT = 'Already exists
            another doctor with the same phone number!';
        end if;
    end if;
end if;
end$$
delimiter ;
```

Function to compute the total number of no-shows (i.e., appointments where the client missed the consult) for clients of a given gender, within a given age group, and within a given year.

```
delimiter $$
create function no_shows(gender char(20),age_lower int,age_upper int,appointment_year int)
returns integer
return (select count(*) from appointment natural left join consultation
inner join client on appointment.VAT_client=client.VAT
where client.gender=gender and client.age between age_lower and age_upper
and extract(year from appointment.date_timestamp)=appointment_year
and consultation.VAT_doctor is null)
$$
delimiter ;
```

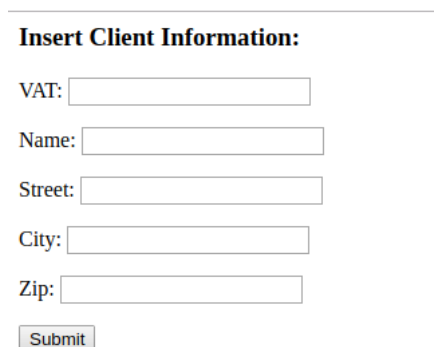
Stored procedure to change the salary of all doctors that have been practicing for more than x years, where x is an input parameter. The new salary should correspond to a raise of 10 percent over the original salary, in the case of doctors with more than 100 consults in the current year, and to a raise of 5 percent otherwise.

```
delimiter $$  
create procedure doctors_salary ( in practice_years int )  
begin  
  
update employee set salary =1.1*salary where employee.VAT in(  
select VAT from permanent_doctor inner join  
consultation on permanent_doctor.VAT = consultation.VAT_doctor  
where permanent_doctor.years > practice_years  
and year(date(consultation.date_timestamp)) = year(now())  
group by permanent_doctor.VAT  
having count(date_timestamp)>100);  
  
update employee set salary =1.05*salary where employee.VAT in(  
select VAT from permanent_doctor inner join  
consultation on permanent_doctor.VAT = consultation.VAT_doctor  
where permanent_doctor.years > practice_years  
and year(date(consultation.date_timestamp)) = year(now())  
group by permanent_doctor.VAT  
having count(date_timestamp)<=100);  
  
end $$  
delimiter ;
```

PHP/HTML

First Part

When a client comes to the clinic asking for an appointment, the page displayed is the page below:



Insert Client Information:

VAT:

Name:

Street:

City:

Zip:

Figura 1: *insert_client_info.php*

As we can see, we can fill in some information (VAT, Name, Street, City or Zip-Code) in order to find if the client is already in the database. If we leave a filter in blank, it will display all the clients in the database

with every value of the field we left blank. The process of searching a client is done through a SQL query.

The code is:

```
<html>
  <body>
    <form action="client_list.php" method="post">
      <h3>Insert Client Information:</h3>
      <p>VAT: <input type="text" name="VAT"/></p>
      <p>Name: <input type="text" name="name"/></p>
      <p>Street: <input type="text" name="street"/></p>
      <p>City: <input type="text" name="city"/></p>
      <p>Zip: <input type="text" name="zip"/></p>
      <p><input type="submit" value="Submit"/></p>
    </form>
  </body>
</html>
```


When we submit the search for a client, we are directed to a page where the client list is displayed, and we can observe each client by its VAT, Name, Street, City and Zip-Code. We can observe the following in the page:

[Home](#)

Clients

VAT	Name	Street	City	Zip	
101010101	Paul McCartney	Lower Street	McAllen	5346-134	Schedule appointment
110110110	Bryan Adams	Edmund Street	Rolla	3542-133	Schedule appointment
121212121	Celine Dion	Third Street	Joplin	3455-245	Schedule appointment
123456789	John Mayer	Bulkeley Street	Los Angeles	1384-142	Schedule appointment
131313131	Marvin Gaye	St Peters Street	Fillmore	5743-532	Schedule appointment
141414141	Bonnie Tyler	Avondale Street	Bloomington	3456-245	Schedule appointment
151515151	Stevie Wonder	Old Mill Street	Lynchburg	4562-623	Schedule appointment
161616161	John Lennon	Homefield Street	Northampton	4653-645	Schedule appointment
171717171	Freddie Mercury	Rosewood Street	Towson	3243-453	Schedule appointment
181818181	Elton John	Smith Street	Tallahassee	1246-654	Schedule appointment
191919191	Mick Jagger	Warren Street	San Marino	5674-765	Schedule appointment
202020202	Morgan Bush	Moorside Street	Northampton	3442-765	Schedule appointment
212121212	Abraham Walters	Teal Street	Monett	3432-123	Schedule appointment
220220220	Sara Wolf	Bouverie Street	Baton Rouge	4328-134	Schedule appointment
232323232	Felicity Morgan	Newport Street	Thaxted	5234-765	Schedule appointment
242424242	Sabrina Adams	Ashleigh Street	Beckenham	1344-765	Schedule appointment
252525252	Bruce Reese	Addison Street	Westfield	4328-531	Schedule appointment
262626262	Harriet Daniels	Ingles Street	Needles	4134-765	Schedule appointment
272727272	May Torres	Moseley Street	Norris	5325-765	Schedule appointment
282828282	Shane Reyes	Belward Street	Uxbridge	4134-123	Schedule appointment
292929292	Evangeline Chapman	Moorside Street	Silssaltquay	1343-132	Schedule appointment
888888888	George Michael	Brooklyn Street	Winslow	9123-123	Schedule appointment
999999999	David Bowie	Broomhill Street	Hastings	2245-442	Schedule appointment

[New Client](#)

Figura 2: *client_list.php*

The code for this page is:

```
<html>
  <body>
    <a href="insert_client_info.php">Home</a>
    <h3>Clients</h3>
  </body>
</html>
```

```
$host = "db.tecnico.ulisboa.pt";
$user = "ist186980";
$pass = "tcsa6810";
$dsn = "mysql:host=$host;dbname=$user";
try
{
    $connection = new PDO($dsn, $user, $pass);
}
catch(PDOException $exception)
{
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
}

$VAT = $_REQUEST['VAT'];
$name = $_REQUEST['name'];
$street = $_REQUEST['street'];
$city = $_REQUEST['city'];
$zip = $_REQUEST['zip'];

$sql = $connection->prepare("SELECT * FROM client WHERE VAT LIKE :VAT AND
(name LIKE :name AND street LIKE :street AND city LIKE :city AND zip LIKE :zip)");
$sql->bindValue(':VAT', "%{$VAT}%");
$sql->bindValue(':name', "%{$name}%");
$sql->bindValue(':street', "%{$street}%");
$sql->bindValue(':city', "%{$city}%");
$sql->bindValue(':zip', "%{$zip}%");
$sql->execute();
$result = $sql->fetchAll(PDO::FETCH_ASSOC);

if(count($result) == 0)
{
    echo("<p>No clients found</p>");
    echo("<p><a href=\"create_new_client.php\">New Client</a></p>");
    $connection = null;
    exit();
}

echo("<table border=\"0\" cellspacing=\"5\">\n");
echo("<tr>\n<td>VAT</td>\n<td>Name</td>\n<td>Street</td>\n
<td>City</td>\n<td>Zip</td>\n</tr>");
foreach($result as $row)
{
    echo("<tr>\n");
    echo("<td><a href=\"appointments_list.php?VAT=\"");
```

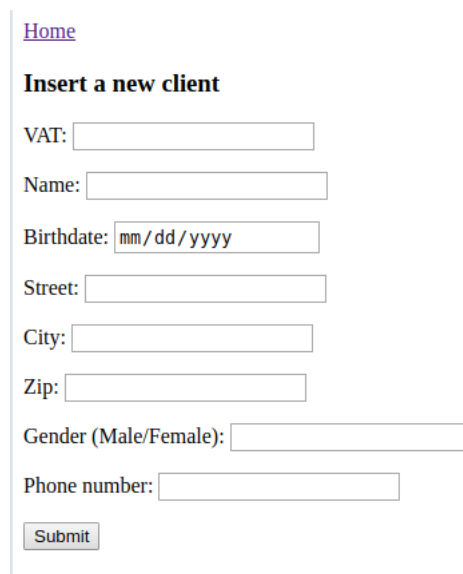
```

        echo($row['VAT']);
        echo("<a href=\"schedule_appointment.php?VAT=\">Schedule appointment</a></td>\n");
        echo("<td>{$row['name']}</td>\n");
        echo("<td>{$row['street']}</td>\n");
        echo("<td>{$row['city']}</td>\n");
        echo("<td>{$row['zip']}</td>\n");
        echo("<td><a href=\"schedule_appointment.php?VAT=\">Schedule appointment</a></td>\n");
        echo($row['VAT']);
        echo("<p><a href=\"create_new_client.php\">New Client</a></p>");
    }
    echo("</table>\n");
    echo("<p><a href=\"create_new_client.php\">New Client</a></p>");

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

```

Then, if the client wasn't in the database already, we can click the "New Client" button in the left bottom corner of the page to proceed to create a new client and add it to the database, and the below page is presented:



The screenshot shows a web page with a header link "Home". Below it is a section titled "Insert a new client". This section contains several input fields: "VAT:", "Name:", "Birthdate:" (with a placeholder "mm/dd/yyyy"), "Street:", "City:", "Zip:", "Gender (Male/Female):", and "Phone number:". At the bottom of this section is a "Submit" button.

Figura 3: *create_new_client.php*

The code to create a new client is:

```
<html>
  <body>
    <a href="insert_client_info.php">Home</a>
    <form action="insert_new_client.php" method="post">
      <h3>Insert a new client</h3>
      <p>VAT: <input type="text" name="VAT"/></p>
      <p>Name: <input type="text" name="name"/></p>
      <p>Birthdate: <input type="date" name="birth_date" /></p>
      <p>Street: <input type="text" name="street"/></p>
      <p>City: <input type="text" name="city"/></p>
      <p>Zip: <input type="text" name="zip"/></p>
      <p>Gender (Male/Female): <input type="text" name="gender"/></p>
      <p>Phone number: <input type="text" name="phone_number"/></p>
      <p><input type="submit" value="Submit"/></p>
    </form>
  </body>
</html>
```

This code prompts the user to insert the new client's VAT, name, birthdate, street, city, zipcode, gender and phone number in order to add it to the database.

If the client already exists in the database, we are able to check if he/she has appointments/consultations scheduled or we can proceed to schedule an appointment. If we want to see the client's current appointments/consultations, then the user must click the VAT of the client he wants to check, which leads to this page (the page below is an example page, varying according to each client):

[Home](#)

Appointments/Consultations

Date/Time	Doctor VAT	Doctor Name
2019-01-01 14:00:00	333333333	Sally Fernandez
2019-01-02 10:00:00	333333333	Sally Fernandez
2019-01-07 15:00:00	333333333	Sally Fernandez New Consultation
2019-12-19 13:00:00	333333333	Sally Fernandez New Consultation

Figura 4: *appointments_list.php*

The corresponding php code is listed below:

```
<html>
  <body>
    <a href="insert_client_info.php">Home</a>
    <h3>Appointments/Consultations</h3>
  <?php
    $host = "db.tecnico.ulisboa.pt";
    $user = "ist186980";
    $pass = "tcsa6810";
    $dsn = "mysql:host=$host;dbname=$user";
    try
    {
        $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error: ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }

    $VAT = $_GET['VAT'];

    $sql = $connection->prepare("SELECT appointment.date_timestamp as date,
    appointment.VAT_doctor as VAT_d, employee.name as doc_name, consultation.VAT_doctor
    as consultation_VAT_doc FROM appointment NATURAL LEFT JOIN consultation INNER JOIN
    employee on employee.VAT = appointment.VAT_doctor
    WHERE VAT_client = :VAT order by appointment.date_timestamp");
    $sql->bindValue(':VAT', "$VAT");
    $sql->execute();
    $result = $sql->fetchAll(PDO::FETCH_ASSOC);
    if(count($result) == 0)
    {
        echo("<p>No appointments/consultations found for selected client</p>");
        $connection = null;
        exit();
    }
    echo("<table border=\"0\" cellspacing=\"5\">\n");
    echo("<tr>\n<td>Date/Time</td>\n<td>Doctor VAT</td>\n<td>Doctor Name<td>\n</tr>");
    foreach($result as $row)
    {
        echo("<tr>\n");

        if($row['consultation_VAT_doc'] != NULL)
        {
            echo("<td><a href=\"consultation_list.php?date_timestamp=");
            echo($row['date']);
```

```

        echo("&VAT=");
        echo($row['VAT_d']);
        echo(">{$row['date']}</a></td>\n");
    }
    else
    {
        echo("<td>{$row['date']}</td>\n");
    }
    echo("<td>{$row['VAT_d']}</td>\n");
    echo("<td>{$row['doc_name']}</td>\n");
    if($row['consultation_VAT_doc'] == NULL)
    {
        echo("<td><a href=\"create_consultation.php?date_timestamp=");
        echo($row['date']);
        echo("&VAT_doctor=");
        echo($row['VAT_d']);
        echo(">New Consultation</a></td>\n");
    }
    echo("</tr>\n");
}
echo("</table>");

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

```

In this page, the user can see the client's consultations by pressing each one of the consultations in the Date/Time column. By pressing one of them, it is then possible to see the details regarding the consultation.

If the client wants to schedule an appointment, the user presses the "Schedule Appointment" button on the right of the desired client which leads to this page:

[Home](#)

Schedule Appointment:

Date/Time:

Figura 5: *Schedule Appointment*

[Home](#)

Schedule Appointment:

Date/Time:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Figura 6: *Schedule Appointment*

The user then proceeds to enter the desired date for an appointment and then checks for the available doctors at that time through a SQL query. If the user doesn't introduce an accurate time between 9am and 5pm, then the next page will present an error message. If the user introduces a valid time, then it will display the available doctors for the appointment and their corresponding VAT, as well as an optional description regarding the appointment.

[Home](#)

Select Doctor

Doctor:

Description:

Figura 7: *check_available_doctors.php*

The corresponding code is:

```
<html>
  <body>
    <a href="insert_client_info.php">Home</a>
    <form action="insert_appointment.php" method="post">
      <h3>Select Doctor</h3>
```

<?php

```
$host = "db.tecnico.ulisboa.pt";
$user = "ist186980";
$pass = "tcsa6810";
$dsn = "mysql:host=$host;dbname=$user";
try
{
    $connection = new PDO($dsn, $user, $pass);
}
catch(PDOException $exception)
{
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
}

$VAT_client = $_REQUEST['VAT'];
$date_timestamp = date('Y-m-d H:i:s',strtotime($_REQUEST['date_timestamp']));
$dateTimestamp = strtotime($_REQUEST['date_timestamp']);
if($VAT_client == '' || $VAT_client == NULL)
{
    echo("<p>Error: VAT must not be empty</p>");
    exit();
}
if($date_timestamp == '' || $date_timestamp == NULL)
{
    echo("<p>Error: Date/Time must not be empty</p>");
    exit();
}
if(date('i', $dateTimestamp) != 0 || date('s', $dateTimestamp) != 0 || date('H',
$dateTimestamp) < 9 || date('H', $dateTimestamp) > 17)
{
    echo("<p>Error: Time must be an exact hour between 9am and 5pm</p>");
    exit();
}
if($dateTimestamp < time())
{
    echo("<p>Error: Date/Time must be in the future</p>");
    exit();
}

echo("<p>Doctor:<select name=\"VAT_doctor\">");
$sql = $connection->prepare("SELECT * FROM doctor NATURAL JOIN employee d
WHERE NOT EXISTS(SELECT * FROM appointment WHERE d.VAT = appointment.VAT_doctor
AND appointment.date_timestamp = :date_timestamp)");
$sql->bindValue(':date_timestamp', "$date_timestamp");
```



```

$sql->execute();
$result = $sql->fetchAll(PDO::FETCH_ASSOC);
if(count($result) == 0)
{
    echo("<p>No doctors available for selected date and time</p>");
    exit();
}
foreach($result as $row)
{
    $VAT_doctor = $row['VAT'];
    $doctor_name = $row['name'];
    echo("<option value=$VAT_doctor>$doctor_name (VAT: $VAT_doctor)</option>");
}
echo("</select>");
echo("</p>");
echo("<input type=\"hidden\" name=\"VAT_client\" value=\"$VAT_client\">");
echo("<input type=\"hidden\" name=\"date_timestamp\" value=\"$date_timestamp\">");
echo("<p>Description: <input type=\"text\" name=\"description\"/></p>");
echo("<p><input type=\"submit\" value=\"Create Appointment\"/></p>");
$connection = null;

```

?>

```

        </form>
    </body>
</html>

```

If all goes according to plan, then its should be displayed a message stating that the appointment was inserted successfully in the database.

[Home](#)

Appointment inserted successfully

Figura 8: *insert_appointment.php*

The code corresponding to the insertion of an appointment is:

```

<html>
    <body>
        <a href="insert_client_info.php">Home</a>
    <?php
        $host = "db.tecnico.ulisboa.pt";
        $user = "ist186980";
        $pass = "tcsa6810";
        $dsn = "mysql:host=$host;dbname=$user";
        try
        {

```

```
        $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error: ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }

    $VAT_doctor = $_REQUEST['VAT_doctor'];
    $date_timestamp = $_REQUEST['date_timestamp'];
    $description = $_REQUEST['description'];
    $VAT_client = $_REQUEST['VAT_client'];

    $sql = $connection->prepare("INSERT INTO appointment VALUES (:VAT_doctor, :date_timestamp,
    $sql->bindValue(':VAT_doctor', "$VAT_doctor");
    $sql->bindValue(':date_timestamp', "$date_timestamp");
    $sql->bindValue(':description', "$description");
    $sql->bindValue(':VAT_client', "$VAT_client");
    $sql->execute();

    $nrows=$sql->rowCount();
    if($nrows == 1)
        echo("<p>Appointment inserted successfully</p>");
    else if($nrows == 0)
    {
        echo("<p>Error: Appointment not inserted</p>");
        exit();
    }
    $connection = null;
?>
</body>
</html>
```

Second Part

While in the Appointments/Consultations web page, the user can create a new consultation by pressing the "New Consultation" button next to the scheduled appointment. The user is then redirected to the following page:

[Home](#)

New Consultation

Subjective observation:

Objective observation:

Assessment:

Plan:

Assistant nurses:

☐ Samuel Stevenson (VAT: 555555555)

☐ Otis Burns (VAT: 666666666)

Diagnostics:

☐ Bacterial Oral Infection (ID: 101050D)

☐ Disorder of Hard Tissues of Tooth (ID: 111975D)

☐ Dental Cavities (ID: 118065D)

☐ Sensitive Dentin (ID: 135048D)

☐ Gingivitis (ID: 138192D)

☐ Infectious Disease (ID: 140880D)

Prescriptions:

☐ Ibuprofen (Lab: Boots Group)

Dosage:

Description:

Diagnostic:

Bacterial Oral Infection (ID: 101050D) ▼

☐ Orajel (Lab: Norwich Warner Pharmaceuticals)

Dosage:

Description:

Diagnostic:

Bacterial Oral Infection (ID: 101050D) ▼

☐ Tylenol (Lab: McNeil)

Dosage:

Description:

Diagnostic:

Bacterial Oral Infection (ID: 101050D) ▼

Create

Figura 9: *create_consultation.php*

The corresponding code is:

```
<html>
  <body>
    <a href="insert_client_info.php">Home</a>
    <form action="insert_consultation.php" method="post">
      <h3>New Consultation</h3>
    </form>
  </body>
</html>

<?php
    $host = "db.tecnico.ulisboa.pt";
    $user = "ist186980";
    $pass = "tcsa6810";
    $dsn = "mysql:host=$host;dbname=$user";

    try
    {
        $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error: ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }

    $VAT_doctor = $_GET['VAT_doctor'];
    $date_timestamp = $_GET['date_timestamp'];
    if($VAT_doctor == '' || $VAT_doctor == NULL)
    {
        echo("<p>Error: VAT_doctor must not be empty</p>");
        exit();
    }
    if($date_timestamp == '' || $date_timestamp == NULL)
    {
        echo("<p>Error: date_timestamp must not be empty</p>");
        exit();
    }

    echo("<input type=\"hidden\" name=\"VAT_doctor\" value=\"$VAT_doctor\">");
    echo("<input type=\"hidden\" name=\"date_timestamp\" value=\"$date_timestamp\">");

    $sql = $connection->prepare("SELECT * from appointment WHERE VAT_doctor = :VAT_doctor
    AND date_timestamp = :date_timestamp");
    $sql->bindValue(':VAT_doctor', "$VAT_doctor");
    $sql->bindValue(':date_timestamp', "$date_timestamp");
    $sql->execute();
    $result = $sql->fetchAll(PDO::FETCH_ASSOC);
    if(count($result) == 0)
    {
        echo("<p>Error: There is no appointment with the parameters specified</p>");
        $connection = null;
    }
}
```

```
        exit();
    }

    echo("<p>Subjective observation: <input type=\"text\" name=\"SOAP_S\"/></p>");
    echo("<p>Objective observation: <input type=\"text\" name=\"SOAP_O\"/></p>");
    echo("<p>Assessment: <input type=\"text\" name=\"SOAP_A\"/></p>");
    echo("<p>Plan: <input type=\"text\" name=\"SOAP_P\"/></p>");

    echo("<p>Assistant nurses:</p>");
    $sql = $connection->prepare("SELECT * from nurse NATURAL JOIN employee");
    $sql->execute();
    $result = $sql->fetchAll(PDO::FETCH_ASSOC);
    if(count($result) == 0)
    {
        echo("<p>Error: There are no nurses in the database</p>");
    }
    else
    {
        foreach($result as $row)
        {
            echo("<p><input type=\"checkbox\" name=\"VATs_nurses[]\" "
                value=\"{$row['VAT']}\">{$row['name']} (VAT: {$row['VAT']})</p>");
        }
    }

    echo("<p>Diagnostics:</p>");
    $sql = $connection->prepare("SELECT * from diagnostic_code");
    $sql->execute();
    $result = $sql->fetchAll(PDO::FETCH_ASSOC);
    if(count($result) == 0)
    {
        echo("<p>Error: There are no diagnostic codes in the database</p>");
    }
    else
    {
        foreach($result as $row)
        {
            echo("<p><input type=\"checkbox\" name=\"diagnostic_codes[]\" "
                value=\"{$row['ID']}\">{$row['description']} (ID: {$row['ID']})</p>");
        }
    }

    echo("<p>Prescriptions:</p>");
    $sql2 = $connection->prepare("SELECT * from medication");
```

```

$sql2->execute();
$result2 = $sql2->fetchAll(PDO::FETCH_ASSOC);
if(count($result2) == 0)
{
    echo("<p>Error: There are no medications in the database</p>");
}
else
{
    echo("<table border=\"0\" cellpadding=\"5\">\n");
    foreach($result2 as $row)
    {
        echo("<tr>\n");
        echo("<td><input type=\"checkbox\" name=\"prescription_medication[]\" value=\"{"$row['name']}|{"$row['lab']}\" >{"$row['name']} (Lab: {"$row['lab']}\" </td>\n");
        echo("<td>Dosage: <input type=\"text\" name=\"prescription_dosage[{"$row['name']}|{"$row['lab']}\" > </td>\n");
        echo("<td>Description: <input type=\"text\" name=\"prescription_description[{"$row['name']}|{"$row['lab']}\" > </td>\n");
        echo("<td>Diagnostic: <select name=\"prescription_diagnostic[{"$row['name']}|{"$row['lab']}\" >");
        $sql->execute();
        $result = $sql->fetchAll(PDO::FETCH_ASSOC);
        foreach($result as $row2)
        {
            echo("<option value=\"{"$row2['ID']}\">{"$row2['description']} (ID: {"$row2['ID']}\"</option>");
        }
        echo("</select>");
        echo("</td>\n");
        echo("</tr>\n");
    }

    echo("</table>");
    echo("<p><input type=\"submit\" value=\"Create\"/></p>");
}

$connection = null;

?>

</form>

</body>

</html>

```

In this page, its possible to see all the details regarding the creation of a new consultation (SOAP notes, the assistant nurses, diagnostics and prescriptions). When a new consultation is created we are directed to the following web page:

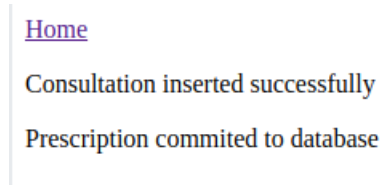


Figura 10: *insert_consultation.php*

The code regarding the insertion of new consultations is the following:

```
<html>
    <body>
        <a href="insert_client_info.php">Home</a>
    </body>
</html>

<?php
    $host = "db.tecnico.ulisboa.pt";
    $user = "ist186980";
    $pass = "tcsa6810";
    $dsn = "mysql:host=$host;dbname=$user";

    try
    {
        $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error: ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }

    $SOAP_S = $_REQUEST['SOAP_S'];
    $SOAP_O = $_REQUEST['SOAP_O'];
    $SOAP_A = $_REQUEST['SOAP_A'];
    $SOAP_P = $_REQUEST['SOAP_P'];

    $VAT_doctor = $_REQUEST['VAT_doctor'];
    $date_timestamp = $_REQUEST['date_timestamp'];
    if($VAT_doctor == '' || $VAT_doctor == NULL)
    {
        echo("<p>Error: VAT_doctor must not be empty</p>");
        exit();
    }
}
```

```
if($date_timestamp == '' || $date_timestamp == NULL)
{
    echo("<p>Error: date_timestamp must not be empty</p>");
    exit();
}

$connection->beginTransaction();
$sql = $connection->prepare("INSERT INTO consultation VALUES (:VAT_doctor,
:date_timestamp, :SOAP_S, :SOAP_O, :SOAP_A, :SOAP_P)");
$sql->bindValue(':VAT_doctor', "$VAT_doctor");
$sql->bindValue(':date_timestamp', "$date_timestamp");
$sql->bindValue(':SOAP_S', "$SOAP_S");
$sql->bindValue(':SOAP_O', "$SOAP_O");
$sql->bindValue(':SOAP_A', "$SOAP_A");
$sql->bindValue(':SOAP_P', "$SOAP_P");
$sql->execute();

$nrows=$sql->rowCount();
if($nrows == 1)
    echo("<p>Consultation inserted successfully</p>");
else if($nrows == 0)
{
    echo("<p>Error: Consultation not inserted</p>");
    echo("<p>Rolling Back...</p>");
    $connection->rollBack();
    exit();
}

foreach($_REQUEST['VATs_nurses'] as $assistant_VAT)
{
    $sql = $connection->prepare("INSERT INTO consultation_assistant VALUES
(:VAT_doctor, :date_timestamp, :assistant_VAT)");
    $sql->bindValue(':VAT_doctor', "$VAT_doctor");
    $sql->bindValue(':date_timestamp', "$date_timestamp");
    $sql->bindValue(':assistant_VAT', "$assistant_VAT");
    $sql->execute();

    $nrows=$sql->rowCount();
    if($nrows == 1)
        echo("<p>Consultation assistant inserted successfully</p>");
    else if($nrows == 0)
    {
        echo("<p>Error: Consultation assistant not inserted</p>");
        echo("<p>Rolling Back...</p>");
        $connection->rollBack();
        exit();
    }
}
```



```
}

foreach($_REQUEST['diagnostic_codes'] as $diagnostic_ID)
{
    $sql = $connection->prepare("INSERT INTO consultation_diagnostic VALUES
    (:VAT_doctor, :date_timestamp, :diagnostic_ID)");
    $sql->bindValue(':VAT_doctor', "$VAT_doctor");
    $sql->bindValue(':date_timestamp', "$date_timestamp");
    $sql->bindValue(':diagnostic_ID', "$diagnostic_ID");
    $sql->execute();
    $nrows=$sql->rowCount();
    if($nrows == 1)
        echo("<p>Diagnostic code inserted successfully</p>");
    else if($nrows == 0)
    {
        echo("<p>Error: Diagnostic code not inserted</p>");
        echo("<p>Rolling Back...</p>");
        $connection->rollBack();
        exit();
    }
}

foreach($_REQUEST['prescription_medication'] as $medication)
{
    list($medication_name,$medication_lab) = explode('|', $medication);
    $dosage = $_REQUEST['prescription_dosage'][$medication];
    $description = $_REQUEST['prescription_description'][$medication];
    $diagnostic_ID = $_REQUEST['prescription_diagnostic'][$medication];
    $sql = $connection->prepare("INSERT INTO prescription VALUES
    (:medication_name, :medication_lab, :VAT_doctor, :date_timestamp,
    :diagnostic_ID, :dosage, :description)");
    $sql->bindValue(':medication_name', "$medication_name");
    $sql->bindValue(':medication_lab', "$medication_lab");
    $sql->bindValue(':VAT_doctor', "$VAT_doctor");
    $sql->bindValue(':date_timestamp', "$date_timestamp");
    $sql->bindValue(':diagnostic_ID', "$diagnostic_ID");
    $sql->bindValue(':dosage', "$dosage");
    $sql->bindValue(':description', "$description");
    $sql->execute();
    $nrows=$sql->rowCount();
    if($nrows == 1)
        echo("<p>Prescription inserted successfully</p>");
    else if($nrows == 0)
    {
        echo("<p>Error: Prescription not inserted</p>");
        echo("<p>Rolling Back...</p>");
        $connection->rollBack();
        exit();
    }
}
```

```

    }
  }

  $connection->commit();
  echo("<p>Prescription committed to database</p>");
  $connection = null;
?>
</body>
</html>

```

If there was already a consultation, it is possible to see all the consultation details by pressing the Date/Time of the consultation (example below):

[Home](#)

Consultation

Subjective observation: xxxxxxxx

Objective observation: xxxxxxxx

Assessment: xxxxxxxx

Plan: xxxxxxxx

Assistant nurses:

Otis Burns (VAT: 666666666)

Diagnostic ID	Diagnostic Description	Medication Name	Medication Lab	Medication Dosage	Medication Description
140880D	Infectious Disease	Ibuprofen	Boots Group	100mg	Every night

[Create Dental Charting](#)

Figura 11: *consultation_list.php*

The corresponding code to the consultation list is:

```

<html>
  <body>
    <a href="insert_client_info.php">Home</a>
    <h3>Consultation</h3>
  <?php
    $host = "db.tecnico.ulisboa.pt";
    $user = "ist186980";
    $pass = "tcsa6810";
    $dsn = "mysql:host=$host;dbname=$user";
    try
    {
      $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)

```

```
{
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
}

$VAT_doctor = $_GET['VAT'];
$date_timestamp = $_GET['date_timestamp'];

$sql = $connection->prepare("SELECT * from consultation WHERE VAT_doctor
= :VAT_doctor AND date_timestamp = :date_timestamp");
$sql->bindValue(':VAT_doctor', "$VAT_doctor");
$sql->bindValue(':date_timestamp', "$date_timestamp");
$sql->execute();
$result = $sql->fetch(PDO::FETCH_ASSOC);
if(count($result) == 0)
{
    echo("<p>No consultation found for selected parameters</p>");
    $connection = null;
    exit();
}

echo("<p>Subjective observation: {$result['SOAP_S']}</p>");
echo("<p>Objective observation: {$result['SOAP_O']}</p>");
echo("<p>Assessment: {$result['SOAP_A']}</p>");
echo("<p>Plan: {$result['SOAP_P']}</p>");

$sql = $connection->prepare("SELECT * from consultation_assistant
NATURAL JOIN employee WHERE VAT_nurse = VAT AND VAT_doctor
= :VAT_doctor AND date_timestamp = :date_timestamp");
$sql->bindValue(':VAT_doctor', "$VAT_doctor");
$sql->bindValue(':date_timestamp', "$date_timestamp");
$sql->execute();
$result = $sql->fetchAll(PDO::FETCH_ASSOC);
if(count($result) == 0)
{
    echo("<p><br>There are no assistant nurses for
this consultation</p>");
}
else
{
    echo("<p><br>Assistant nurses:</p>");
    foreach($result as $row)
    {
```

```
        echo("<p>{$row['name']} (VAT: {$row['VAT']})</p>");
    }
}

$sql = $connection->prepare("SELECT consultation_diagnostic.ID,
prescription.description, name, lab, dosage, diagnostic_code.description
as ddescription from consultation NATURAL JOIN consultation_diagnostic
NATURAL LEFT OUTER JOIN prescription LEFT JOIN diagnostic_code on
diagnostic_code.ID = consultation_diagnostic.ID WHERE VAT_doctor
= :VAT_doctor AND date_timestamp = :date_timestamp");
$sql->bindValue(':VAT_doctor', "$VAT_doctor");
$sql->bindValue(':date_timestamp', "$date_timestamp");
$sql->execute();
$result = $sql->fetchAll(PDO::FETCH_ASSOC);
if(count($result) == 0)
{
    echo("<p><br>There are no diagnostics/prescriptions found for
this consultation</p>");
}
else
{
    echo("<table border=\"0\" cellspacing=\"5\">\n");
    echo("<tr>\n<td>Diagnostic ID</td>\n<td>Diagnostic
Description</td>\n<td>Medication Name</td>\n<td>Medication
Lab</td>\n<td>Medication Dosage</td>\n<td>Medication
Description</td>\n");
    foreach($result as $row)
    {
        echo("<tr>\n");
        if($row['ID'] != NULL)
            echo("<td>{$row['ID']}</td>\n");
        else
            echo("<td>---</td>\n");
        if($row['ddescription'] != NULL)
            echo("<td>{$row['ddescription']}</td>\n");
        else
            echo("<td>---</td>\n");
        if($row['name'] != NULL)
            echo("<td>{$row['name']}</td>\n");
        else
            echo("<td>---</td>\n");
        if($row['lab'] != NULL)
            echo("<td>{$row['lab']}</td>\n");
        else
            echo("<td>---</td>\n");
    }
}
```

```
                echo("<td>---</td>\n");
            if($row['dosage'] != NULL)
                echo("<td>{$row['dosage']}</td>\n");
            else
                echo("<td>---</td>\n");
            if($row['description'] != NULL)
                echo("<td>{$row['description']}</td>\n");
            else
                echo("<td>---</td>\n");

            echo("</tr>\n");
        }
        echo("</table>\n");
    }

    echo("<a href=\"create_dental_charting.php?
    VAT_doctor=$VAT_doctor&date_timestamp=$date_timestamp\
    \">Create Dental Charting</a>");
    $connection = null;
?>

</body>
</html>
```

Each consultation is searched through a SQL query.

Third Part

When checking a consultation, it is possible for the user to create a dental charting by creating in the "Create Dental Charting" in the bottom left corner. By clicking there, the user is redirected to the following web page:

[Home](#)

New Dental Charting

Procedure name:

Procedure type:

Procedure description:

Tooth	Measure (mm)	Description
<input type="checkbox"/> Molar (quadrant 1, number 1)	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> Pre-Molar (quadrant 1, number 2)	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> Incisor (quadrant 2, number 1)	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> Molar (quadrant 2, number 2)	<input type="text"/>	<input type="text"/>

Figura 12: *create_dental_charting*

The corresponding code is:

```
<html>
  <body>
    <a href="insert_client_info.php">Home</a>
    <form action="insert_dental_charting.php" method="post">
      <h3>New Dental Charting</h3>
</html>

<?php
    $host = "db.tecnico.ulisboa.pt";
    $user = "ist186980";
    $pass = "tcsa6810";
    $dsn = "mysql:host=$host;dbname=$user";
    try
    {
        $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error: ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }

    $VAT_doctor = $_GET['VAT_doctor'];
    $date_timestamp = $_GET['date_timestamp'];
    echo("<input type=\"hidden\" name=\"VAT_doctor\" value=\"\$VAT_doctor\">");
    echo("<input type=\"hidden\" name=\"date_timestamp\" value=\"\$date_timestamp\">");
```

```

echo("<p>Procedure name: <input type=\"text\" name=\"procedure_name\"/></p>");
echo("<p>Procedure type: <input type=\"text\" name=\"procedure_type\"/></p>");
echo("<p>Procedure description: <input type=\"text\"
name=\"prodedure_description\"/></p>");

$sql = $connection->prepare("SELECT * from teeth");
$sql->execute();
$result = $sql->fetchAll(PDO::FETCH_ASSOC);
if(count($result) == 0)
{
    echo("<p>No teeth found in the database</p>");
    $connection = null;
    exit();
}

echo("<table border=\"0\" cellspacing=\"5\">\n");
echo("<tr>\n<td>Tooth</td>\n<td>Measure (mm)</td>\n<td>Description<td>\n</tr>\n");
foreach($result as $row)
{
    echo("<tr>\n");
    echo("<td><input type=\"checkbox\" name=\"teeth[]\"
    value=\"{ $row['quadrant']}| { $row['number_']}\" >{ $row['name']}
    (quadrant { $row['quadrant']}, number { $row['number_']})</td>\n");
    echo("<td><input type=\"text\"
    name=\"measure_tooth[{ $row['quadrant']}|{ $row['number_']}]\"/></td>\n");
    echo("<td><input type=\"text\"
    name=\"description_tooth[{ $row['quadrant']}|{ $row['number_']}]\"/></td>\n");
    echo("</tr>\n");
}
echo("</table>");
echo("<p><input type=\"submit\" value=\"Create\"/></p>");

$connection = null;
?>

</form>

</body>

</html>

```

If the procedure is not specified enough, then it is impossible for the user to create the dental charting and the page displays an error message after clicking the "Create" button. However, if the dental charting has all the mandatory fields filled then the page will display all the data inserted into the dental charting with a message for each field stating that the data was inserted successfully.

[Home](#)

Procedure inserted successfully

Procedure in consultation inserted successfully

Tooth in dental charting inserted successfully

Tooth in dental charting inserted successfully

Tooth in dental charting inserted successfully

Tooth in dental charting inserted successfully

Dental charting committed to database

Figura 13: *insert_dental_charting.php*

```
<html>
  <body>
    <a href="insert_client_info.php">Home</a>
  <?php
    $host = "db.tecnico.ulisboa.pt";
    $user = "ist186980";
    $pass = "tcsa6810";
    $dsn = "mysql:host=$host;dbname=$user";
    try
    {
        $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error: ");
        echo($exception->getMessage());
        echo("</p>");
        exit();
    }

    $VAT_doctor = $_REQUEST['VAT_doctor'];
    $date_timestamp = $_REQUEST['date_timestamp'];
    $procedure_name = $_REQUEST['procedure_name'];
    $procedure_type = $_REQUEST['procedure_type'];
    $procedure_description = $_REQUEST['procedure_description'];

    if($VAT_doctor == '' || $VAT_doctor == NULL)
    {
        echo("<p>Error: VAT_doctor must not be empty</p>");
        exit();
    }
    if($date_timestamp == '' || $date_timestamp == NULL)
```



```
{
    echo("<p>Error: date_timestamp must not be empty</p>");
    exit();
}
if($procedure_name == '' || $procedure_name == NULL)
{
    echo("<p>Error: Procedure Name must not be empty</p>");
    exit();
}

$connection->beginTransaction();
$sql = $connection->prepare("INSERT INTO procedure_ VALUES (:procedure_name,
:procedure_type)");
$sql->bindValue(':procedure_name', "$procedure_name");
$sql->bindValue(':procedure_type', "$procedure_type");
$sql->execute();

$nrows=$sql->rowCount();
if($nrows == 1)
    echo("<p>Procedure inserted successfully</p>");
else if($nrows == 0)
    echo("<p>Procedure not inserted</p>");

$sql = $connection->prepare("INSERT INTO procedure_in_consultation VALUES
(:procedure_name, :VAT_doctor, :date_timestamp, :procedure_description)");
$sql->bindValue(':procedure_name', "$procedure_name");
$sql->bindValue(':VAT_doctor', "$VAT_doctor");
$sql->bindValue(':date_timestamp', "$date_timestamp");
$sql->bindValue(':procedure_description', "$procedure_description");
$sql->execute();
$nrows=$sql->rowCount();
if($nrows == 1)
    echo("<p>Procedure in consultation inserted successfully</p>");
else if($nrows == 0)
{
    echo("<p>Error: Procedure in consultation not inserted</p>");
    echo("<p>Rolling Back...</p>");
    $connection->rollBack();
    exit();
}

foreach($_REQUEST['teeth'] as $tooth)
{
    list($tooth_quadrant,$tooth_number) = explode('|', $tooth);
    $measure_tooth = $_REQUEST['measure_tooth'][$tooth];
    if($measure_tooth == '' || $measure_tooth == NULL)
    {
```

```
        echo("<p>Error: Tooth gap is not valid</p>");
        echo("<p>Rolling Back...</p>");
        $connection->rollBack();
        exit();
    }
    $description_tooth = $_REQUEST['description_tooth'][$tooth];
    $sql = $connection->prepare("INSERT INTO procedure_charting VALUES
    (:procedure_name, :VAT_doctor, :date_timestamp, :tooth_quadrant,
    :tooth_number, :description_tooth, :measure_tooth)");
    $sql->bindValue(':procedure_name', "$procedure_name");
    $sql->bindValue(':VAT_doctor', "$VAT_doctor");
    $sql->bindValue(':date_timestamp', "$date_timestamp");
    $sql->bindValue(':tooth_quadrant', "$tooth_quadrant");
    $sql->bindValue(':tooth_number', "$tooth_number");
    $sql->bindValue(':description_tooth', "$description_tooth");
    $sql->bindValue(':measure_tooth', $measure_tooth);
    $sql->execute();
    $nrows=$sql->rowCount();
    if($nrows == 1)
        echo("<p>Tooth in dental charting inserted successfully</p>");
    else if($nrows == 0)
    {
        echo("<p>Error: Tooth in dental charting not inserted</p>");
        echo("<p>Rolling Back...</p>");
        $connection->rollBack();
        exit();
    }
}

$connection->commit();
echo("<p>Dental charting committed to database</p>");
$connection = null;
?>

</body>
</html>
```