



Using the Database

Information Systems and Databases

AUTHORS

Jose Silva - 84109
Luís Simões - 81282
Pedro Ferreira - 84166

November 24, 2022

Contents

1	Introduction	1
2	Web Application Using the Database - 1/3	1
2.1	Search a client	1
2.2	Add Appointment	2
2.3	Create a new client	2
2.4	List available doctors	3
3	Web Application Using the Database - 2/3	4
3.1	List of Appointments/Consultations	4
3.2	Consultation Information	6
3.3	Appointment Information	8
3.4	Creating/Updating a Consultation	9
3.4.1	Creating a Consultation	16
3.4.2	Updating a Consultation	18
4	Web Application Using the Database - 3/3	23
4.1	Add procedure in consultation	23
4.2	Charting measures	23
5	Functions, Triggers and Stored Procedures	26
5.1	Trigger: Age Update	26
5.2	Trigger: Same individual, Multiple jobs	26
5.2.1	(a) Nurses, Receptionists and Doctors	26
5.2.2	(b) Permanent Doctors vs Trainees	28
5.3	Trigger: Repeated Phone Numbers	29
5.4	Function: Number of <i>No Shows</i>	31
5.5	Procedure: Salary Change	31

1 Introduction

This project concerns the development of an information system supporting the day-to-day operations of a dental clinic. This third part of the project concerns the creation of a web application to access the database created on the previous parts of the project. During this project, all interactions with the database that the values are selected by the user are protected against SQL injections through Prepared Statements.

The web application can be found [here](#) or go to the link <http://web.ist.utl.pt/ist425466/doctorsearch.php>.

2 Web Application Using the Database - 1/3

2.1 Search a client

We were asked to search for a client even if just part of the information about him was provided. To do so, it was used the code shown above.

Listing 1: Search client

```
1 if(!empty($VAT)||!empty($Name)||!empty($Street)||!empty($City)||!empty($Zip))
2 {
3     $sVAT = "%{$_REQUEST['VAT']}%";
4     $sName = "%{$_REQUEST['Name']}%";
5     $sStreet = "%{$_REQUEST['Street']}%";
6     $sCity = "%{$_REQUEST['City']}%";
7     $sZip = "%{$_REQUEST['Zip']}%";
8     $result = $connection->prepare("SELECT VAT, name FROM client WHERE client.VAT
        LIKE ? and client.name LIKE ? and client.street LIKE ? and client.city LIKE
        ? and client.zip LIKE ?");
9     $result->execute(array($sVAT, $sName, $sStreet, $sCity, $sZip));
10
11
12
13 if ($result->rowCount() == 0)
14 {
15     echo("<p>There is no client with such info.</p>");
16 }
17 else
18 {
19     echo("<table>");
20     echo("<tr> <th>Name</th> <th>VAT</th> <th></th> </tr>");
21     foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
22     {
23         echo("<tr><form action='newAppointment.php' method='post'>");
24         echo("<td>{$row['name']}</td> <td><input type='hidden' name='VAT' value
            ='{$row['VAT']}'>{$row['VAT']} </td>");
25         echo("<td> <input type='submit' value='Add Appointment' /> </td>");
```

```

26         echo("</form>");
27     echo("<form action='listAppointments.php' method='post'>");
28         echo("<td><input type='hidden' name='VAT' value='{<!--
29         ;
30         echo("<td><input type='submit' value='List Appointments/Consultations
31         ' /></td>");
32         echo("</form> </tr>");
33     }
34     echo("</table>");
35 }

```

The query won't run if there is no information about the client.

Search Client

VAT:

Name:

Address: Street: City: Zip:

(a) inputs

Name	VAT		
Pedro Oliveira	123454	<input type="button" value="Add Appointment"/>	<input type="button" value="List Appointments/Consultations"/>
Pedro Cebo	777777777	<input type="button" value="Add Appointment"/>	<input type="button" value="List Appointments/Consultations"/>
Hugo Burro	888888888	<input type="button" value="Add Appointment"/>	<input type="button" value="List Appointments/Consultations"/>

(b) result

Figure 1: Web page - search client

2.2 Add Appointment

As shown on listing 1 line 25, when the results are printed exists the possibility of registering a new appointment for that client. This submit form will change the window to the *newAppointments.php*, sending the *VAT_client* by post, where the query to insert a new appointment will be done.

2.3 Create a new client

There is also the possibility of insert a new client in the database. The code to do so is shown on the listing below and the web page is shown on fig 2.

Listing 2: New client

```

1 <form action="newClient.php" method="post">
2     <h3> New Client </h3>
3     <p>VAT: <input type="text" name="VAT"/></p>
4     <p>Name: <input type="text" name="Name"/></p>
5     <p>Birth date: <input type="date" name="birth_date"/></p>
6     <p>Gender: <input type="radio" name="gender" value="M" > Male
7         <input type="radio" name="gender" value="F"> Female
8         <input type="radio" name="gender" value="0"> Other</p>
9     <p>Address: Street: <input type="text" name="Street"/> City: <input type="text"
10         name="City"/> Zip: <input type="text" name="Zip"/></p>

```

```

11     <input type="submit" value = 'Add Client' />
12 </form>

```

New Client

VAT:

Name:

Birth date:

Gender: ☐ Male ☐ Female ☐ Other

Address: Street: City: Zip:

Figure 2: Web page - new client

When the user presses *Add Client*, he will be redirected to a page where the query that will populate the database will be done. On this page the user can also add phone numbers that will be linked to the new client (figure 3).

Phone Number:

Figure 3: Web page - add phone number

2.4 List available doctors

This function of the program appear two times, it is on the main page and on the *newAppointment.php*.

Listing 3: Check Availability

```

1 $app_timestamp = $_REQUEST['app_date'].' '.$_REQUEST['app_time'].':00';
2
3 $time1 = $_REQUEST['app_time'].':00';
4 $time2 = "01:00";
5
6 $time2 = date("H:i",strtotime($time1)+strtotime($time2));
7
8 $app_timestamp2 = $_REQUEST['app_date'].' '.$time2;
9 if(!empty($_REQUEST['app_date'])&&!empty($_REQUEST['app_time']))
10 {
11     echo("<h3>Available Doctors $app_timestamp</h3>");
12

```

```

13 $result = $connection->prepare("SELECT doc.name, doc.VAT from employee doc,
    doctor where doc.VAT = doctor.VAT and doc.VAT not in( select app.VAT_doctor
    from appointment app where app.VAT_doctor and app.date_timestamp BETWEEN ?
    AND ?) group by doc.VAT");
14 $result->execute(array($app_timestamp, $app_timestamp2));
15
16 ...
17 }

```

On the example above, the date has to be changed to the right format in order be possible to check the time interval on the database. the time slot is a number input because the minutes are irrelevant to the appointment.

List Available Doctor

Date:

Time (h):

(a) inputs

Available Doctors 2019-12-17 10:00

Name	VAT
Jane Sweettooth	123456789
Julia Sweettooth	987654321
Julio Isidro	987656789

(b) result

Figure 4: Web page - list available doctors

3 Web Application Using the Database - 2/3

3.1 List of Appointments/Consultations

To present a list of appointments/consultations it was simply used a natural join between appointments and consultations (since both have the same primary keys) ordered chronologically.

List of Appointments/Consultations for Client 666666666

TYPE	DATE	DOCTOR	
Appointment	2019-01-19 17:55:00	123456789	<input type="button" value="Show Details"/>
Consultation	2019-01-19 17:55:00	123456789	<input type="button" value="Show Details"/>
Appointment	2019-02-09 15:16:00	123456789	<input type="button" value="Show Details"/>
Appointment	2019-02-12 12:02:00	123456789	<input type="button" value="Show Details"/>
Appointment	2019-02-27 16:48:00	123456789	<input type="button" value="Show Details"/>
Appointment	2019-03-02 13:53:00	123456789	<input type="button" value="Show Details"/>

Figure 5: Web page - List of Appointments/Consultations - *listAppointments.php*

Listing 4: List of Appointments/Consultations (listAppointments.php)

```

1 $VAT = $_REQUEST['VAT'];
2
3 $result = $connection->prepare("SELECT * FROM appointment NATURAL LEFT OUTER JOIN
  consultation where VAT_client = ? order by date_timestamp");
4 $result->execute(array($VAT));
5
6 $nrows = $result->rowCount();
7 if ($nrows == 0)
8 {
9     echo("<p>There are no records for this client.</p>");
10 }
11 else
12 {
13     echo("<table>");
14     echo("<tr> <th>TYPE</th> <th>DATE</th> <th>DOCTOR</th> </tr>");
15     foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
16     {
17         $date = $row['date_timestamp'];
18         $doctor = $row['VAT_doctor'];
19         echo("<tr>");
20         echo("<td>Appointment</td> <td>{$date}</td> <td>{$doctor}</td>");
21         echo("<form action='appointment.php' method='post'>");
22         echo("<input type='hidden' name='date' value='{$date}'>");
23         echo("<input type='hidden' name='doctor' value='{$doctor}'>");
24         echo("<input type='hidden' name='description' value='{$row['description']}'>");
25         echo("<input type='hidden' name='client' value='{$VAT}'>");
26         echo("<td><input type='submit' value='Show Details' /></td>");
27         echo("</form>");
28         echo("</tr>");
29         if (!is_null($row['SOAP_S']) or !is_null($row['SOAP_O']) or !is_null($row['
          SOAP_A']) or !is_null($row['SOAP_P']))
30         {
31             echo("<tr>");
32             echo("<td>Consultation</td> <td>{$row['date_timestamp']}</td> <td>{$row['
              VAT_doctor']}</td>");
33             echo("<form action='consultation.php' method='post'>");
34             echo("<input type='hidden' name='date' value='{$date}'>");
35             echo("<input type='hidden' name='doctor' value='{$doctor}'>");
36             echo("<input type='hidden' name='soap_s' value='{$row['SOAP_S']}'>");
37             echo("<input type='hidden' name='soap_o' value='{$row['SOAP_O']}'>");
38             echo("<input type='hidden' name='soap_a' value='{$row['SOAP_A']}'>");
39             echo("<input type='hidden' name='soap_p' value='{$row['SOAP_P']}'>");
40             echo("<td><input type='submit' value='Show Details' /></td>");
41             echo("</form>");
42             echo("</tr>");
43         }

```

```

44 }
45 echo("</table>");
46 }

```

Every appointment and consultation has a *Show Details* button (Figure 5), which will lead the user to a page displaying the information associated to that specific consultation or appointment (Figure 7). All the information is sent as hidden variables so that there is no need to perform a query on the next web page.

3.2 Consultation Information

As far as the page displaying the information regarding an appointment is concerned, it is basically a table showing the appointment's information and a button to insert a new procedure in that consultation. The code of this page is shown below.

Consultation Details

VAT doctor: 987656789

date : 2019-11-17 17:00:00

SOAP_S: s

SOAP_O: o

SOAP_A: p

SOAP_P:

Diagnosis ID : D204

name, lab: medication2, lab1

dosage: 2 em 2 horas

description: nao esquecer

Figure 6: Web page - Consultation Details - *consultation.php*

Listing 5: Consultation Information (consultation.php)

```

1 $doctor = $_REQUEST['doctor'];
2 $date = $_REQUEST['date'];
3
4
5
6 echo("<p>VAT doctor: $doctor");
7 echo("<p>date : {$_REQUEST['date']}");
8 echo("<p>SOAP_S: {$_REQUEST['soap_s']}");
9 echo("<p>SOAP_O: {$_REQUEST['soap_o']}");

```



```

10 echo("<p>SOAP_A: {$_REQUEST['soap_a']}");
11 echo("<p>SOAP_P: {$_REQUEST['soap_p']}");
12
13
14 $result = $connection->prepare("select * from consultation_diagnostic where
    VAT_doctor = ? and date_timestamp = ? ");
15 $result->execute(array($doctor, $date));
16
17
18 $nrows = $result->rowCount();
19 if ($nrows == 0)
20 {
21     echo("<p>There is no Diagnosis.</p>");
22 }
23 else
24 {
25     foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
26     {
27         echo("<table>");
28         echo("<tr><td>Diagnosis ID : {$row['ID']}</td></tr>");
29
30
31         $result2 = $connection->prepare("select * from prescription as p where p.
            VAT_doctor=? and p.date_timestamp = ? group by p.name");
32         $result2->execute(array($doctor, $date));
33         $nrows2 = $result->rowCount();
34         if($nrows2 == 0)
35         {
36             echo("<tr><td></td>");
37             echo("<td>No prescription</td>");
38             echo("</tr>");
39
40         }
41         else
42         {
43             foreach($result2->fetchAll(PDO::FETCH_ASSOC) as $presc)
44             {
45                 echo("<tr><td></td>");
46                 echo("<td>name, lab: {$presc['name']}, {$presc['lab']} </td>");
47                 echo("</tr>");
48                 echo("<tr><td></td>");
49                 echo("<td> dosage: {$presc['dosage']}</td>");
50                 echo("</tr>");
51                 echo("<tr><td></td>");
52                 echo("<td> description: {$presc['description']}</td>");
53                 echo("</tr>");
54             }

```

```

55     }
56
57     echo("</table>");
58 }
59 }
60
61 echo("<form action='dental_charting.php' method='post'>");
62 echo("<input type='hidden' name = 'doctor' value = '$doctor'>");
63 echo("<input type='hidden' name = 'date' value = '$date'>");

```

3.3 Appointment Information

About the page displaying the information about a certain appointment there is no "magic" behind. Nevertheless, it has a button named *Consultation Info* which will send the user to a page where he can create the consultation that is directly related with the displayed appointment, or simply update information about that consultation if it was already created. The code for this page is on Listing 7.

Appointment Details

CLIENT: 6666666666

DOCTOR: 123456789

DATE: 2019-01-19 17:55:00

DESCRIPTION: tratamento

Consultation Info

Figure 7: Web page - Appointment Details - *appointment.php*

Listing 6: Appointment Information (appointment.php)

```

1 <html>
2   <body>
3     <h3>Appointment Details</h3>
4     <?php
5
6     $client = $_REQUEST['client'];
7     $date = $_REQUEST['date'];
8     $doctor = $_REQUEST['doctor'];
9     $description = $_REQUEST['description'];
10

```

```

11  echo("<p>CLIENT: {$client}</p>");
12  echo("<p>DOCTOR: {$doctor}</p>");
13  echo("<p>DATE: {$date}</p>");
14  echo("<p>DESCRIPTION: {$description}</p>");
15  echo("<p><form action='newConsultation.php' method='post'>");
16  echo("<input type='hidden' name='doctor' value='{$doctor}'>");
17  echo("<input type='hidden' name='date' value='{$date}'>");
18  echo("<input type='submit' value='Consultation Info'>");
19  echo("</form></p>");
20
21  ?>
22  </body>
23 </html>

```

3.4 Creating/Updating a Consultation

On this web page (Figure 8), the user can update whatever information he desires about the corresponding consultation. If this consultation was already created, the corresponding information will already appear on the appropriate spaces, but the user can always update that information.

Update Consultation Information

VAT_DOCTOR:	123456789	
DATE_TIMESTAMP:	2019-01-19 17:55:00	
SOAP_S:	<input type="text"/>	
SOAP_O:	<input type="text"/>	
SOAP_A:	<input type="text"/>	
SOAP_P:	<input type="text"/>	
Assistant Nurses:	<input type="checkbox"/> Jane Dentedoce 123746789	
Assistant Nurses:	<input type="checkbox"/> Enfermeira Joana 154235485	
Diagnostic Code:	<input type="checkbox"/> D000	Associated Prescriptions:
		<input type="checkbox"/> medication1 lab1 Dosage: <input type="text"/> Description: <input type="text"/>
		<input type="checkbox"/> medication2 lab1 Dosage: <input type="text"/> Description: <input type="text"/>
		<input type="checkbox"/> palmada mae Dosage: <input type="text"/> Description: <input type="text"/>
Diagnostic Code:	<input type="checkbox"/> D105	Associated Prescriptions:
		<input type="checkbox"/> medication1 lab1 Dosage: <input type="text"/> Description: <input type="text"/>
		<input type="checkbox"/> medication2 lab1 Dosage: <input type="text"/> Description: <input type="text"/>
		<input type="checkbox"/> palmada mae Dosage: <input type="text"/> Description: <input type="text"/>
Diagnostic Code:	<input type="checkbox"/> D106	Associated Prescriptions:
		<input type="checkbox"/> medication1 lab1 Dosage: <input type="text"/> Description: <input type="text"/>
		<input type="checkbox"/> medication2 lab1 Dosage: <input type="text"/> Description: <input type="text"/>
		<input type="checkbox"/> palmada mae Dosage: <input type="text"/> Description: <input type="text"/>

Figure 8: Web page - Creating/Updating a Consultation - *newConsultation.php*

The following code on Listing 7 is divided in two parts, when the consultation was not yet created (line 66 to 114) and when the consultation already existed on the database (line 114 to 227).

Most of the queries are re-utilized for both parts since when we use the *NOT IN* operator, if the consultation does not exist, it would simply list everything as it is supposed.

The first and second parts prepare the information input by the user to send to a page which will do the *INSERTS*, *UPDATES* and *DELETIONS* necessary. Since it is possible to have multiple nurses, prescriptions and medication assigned, the information is stored in arrays, and since the medication depends on the associated prescription, it has to be stored on a multidimensional array,

where the ID of the prescription is one of the keys, the same happens for the dosage and prescription which have to be indexed by the prescription ID and by the (name, lab) of the corresponding medication. Note: Since this project assumes that our database can have multiple nurses assigned to the same consultation, we changed the database created on Part2 adding the nurse VAT as a primary key of consultation_assistant.

The difference for the already created consultations is that we wish to present the information that is already on the database, this justifies the use of queries that distinguish nurses, diagnosis and prescriptions that are assigned to this consultations, from the available nurses and all of the diagnosis and prescriptions that can be assign to the consultation. The ones assigned will appear with the "checkbox" already "checked". There's also a need to store arrays corresponding to what was previously on the database, what the user wants to keep and what he wants to add, in order to do the deletions, updates and insertions. Note: Since the prescriptions are different for each diagnosis, an SQL query has to be executed for every already assigned diagnostic.

Also of notice that in order to simplify the indexing of the primary keys of medications (name, lab) we decided to serialize each entry of a medication into a string, which are then stored on an array, and will then be unserialized on the pages responsible for the inserts, updates and deletions.

Listing 7: Creating/Updating a Consultation (newConsultation.php)

```

1  $date = $_REQUEST['date'];
2  $doctor = $_REQUEST['doctor'];
3
4  $result = $connection->prepare("SELECT * FROM consultation WHERE VAT_doctor = ? and
    date_timestamp = ?");
5  $result->execute(array($doctor, $date));
6  $error = $result->errorInfo();
7  if ($error[1] != ''){
8      echo("<p>Error: {$error[2]}</p>");
9  }
10
11 $nrows = $result->rowCount();
12
13 // Get nurses already assigned to this consultation
14 $result2 = $connection->prepare("SELECT employee.name AS name, employee.VAT AS VAT
    FROM consultation_assistant AS ca, employee WHERE ca.VAT_doctor = ? and ca.
    date_timestamp = ? and ca.VAT_nurse = employee.VAT");
15 $result2->execute(array($doctor, $date));
16
17 $error2 = $result2->errorInfo();
18 if ($error2[1] != ''){
19     echo("<p>Error: {$error2[2]}</p>");
20 }
21 $nrows2 = $result2->rowCount();
22
23 // Get nurses available for the desired timestamp
24 $result3 = $connection->prepare("SELECT employee.name AS name, employee.VAT AS VAT
    FROM nurse, employee WHERE nurse.VAT NOT IN (SELECT VAT_nurse FROM
    consultation_assistant WHERE date_timestamp = ?) AND nurse.VAT = employee.VAT");

```

```

25 $result3->execute(array($date));
26
27 $error3 = $result3->errorInfo();
28 if ($error3[1] != ''){
29     echo("<p>Error: {$error3[2]}</p>");
30 }
31 $nrows3 = $result3->rowCount();
32
33 //Get diagnostic codes already assigned to this consultation
34 $result_consult_diagnostic = $connection->prepare("SELECT ID FROM
    consultation_diagnostic WHERE VAT_doctor = ? and date_timestamp = ?");
35 $result_consult_diagnostic->execute(array($doctor,$date));
36
37 $error4 = $result_consult_diagnostic->errorInfo();
38 if ($error4[1] != ''){
39     echo("<p>Error: {$error4[2]}</p>");
40 }
41
42 $nrows_consult_diagnostic = $result_consult_diagnostic->rowCount();
43
44 // Get diagnostic codes
45 $result_diagcodes = $connection->prepare("SELECT ID FROM diagnostic_code WHERE ID
    NOT IN (SELECT ID FROM consultation_diagnostic WHERE VAT_doctor = ? and
    date_timestamp = ?)");
46 $result_diagcodes->execute(array($doctor,$date));
47 $error5 = $result_diagcodes->errorInfo();
48 if ($error5[1] != ''){
49     echo("<p>Error: {$error5[2]}</p>");
50 }
51 $nrows_diagcodes = $result_diagcodes->rowCount();
52
53 // Get medication
54 $sql5 = "SELECT * FROM medication";
55 $result_medication = $connection->query($sql5);
56 $nrows_medication = $result_medication->rowCount();
57 foreach($result_medication as $meds)
58 {
59     $med_array[] = array("name" => $meds['name'], "lab" => $meds['lab']);
60 }
61
62 echo("<table>");
63 echo("<tr><td>VAT_DOCTOR:</td> <td>{$doctor}</td></tr>");
64 echo("<tr><td>DATE_TIMESTAMP:</td> <td>{$date}</td></tr>");
65
66 // if the consultation was not yet created
67 if ($nrows == 0)
68 {

```

```

69 echo("<form action='addConsultation.php' method='post'>");
70 echo("<input type='hidden' name='doctor' value='{&#123;doctor}&#123;'>");
71 echo("<input type='hidden' name='date' value='{&#123;date}&#123;'>");
72 echo("<tr><td>SOAP_S:</td> <td><input type='text' name='SOAP_S' /></td></tr>");
73 echo("<tr><td>SOAP_O:</td> <td><input type='text' name='SOAP_O' /></td></tr>");
74 echo("<tr><td>SOAP_A:</td> <td><input type='text' name='SOAP_A' /></td></tr>");
75 echo("<tr><td>SOAP_P:</td> <td><input type='text' name='SOAP_P' /></td></tr>");
76 // if there are nurses available
77 if ($nrows3 != 0)
78 {
79     foreach($result3->fetchAll(PDO::FETCH_ASSOC) as $row3)
80     {
81         echo("<tr><td>Assistant Nurses: </td> <td><input type='checkbox' name='
            nurse_list[]' value='{&#123;row3['VAT']}&#123;'><label>{&#123;row3['name']}&#123; {&#123;row3['VAT
            ']}</label></td></tr>");
82     }
83 }
84 else
85 {
86     echo("<tr><td>ATTENTION: </td><td>No nurse/assistant available for this
        consultation</td></tr>");
87 }
88 foreach($result_diagcodes->fetchAll(PDO::FETCH_ASSOC) as $diagnostic)
89 {
90     // check list of diagnostic codes
91     $id = $diagnostic['ID'];
92     echo("<tr><td>Diagnostic Code: </td> <td><input type='checkbox' name='codes[]'
        value='{&#123;diagnostic['ID']}&#123;'><label>{&#123;diagnostic['ID']}&#123;</label></td>");
93     // check list of medication per diagnostic
94     if ($nrows_medication != 0)
95     {
96         echo("<td>Associated Prescriptions: </td></tr>");
97         foreach($med_array as $med)
98         {
99             $med_name = $med['name'];
100             $med_lab = $med['lab'];
101             $value = serialize($med);
102             echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name='
                meds_id[$id][]' value='$value'><label>{&#123;med_name}&#123; {&#123;med_lab}&#123;</label></td
                >");
103             echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[$id][
                $med_name][$med_lab]' /></td>");
104             echo("<td>Description:</td> <td><input type='text' name='description[$id][
                $med_name][$med_lab]' /></td></tr>");
105         }
106     }
107     else

```

```

108     {
109         echo("<td>No Medication on the DataBase</td></tr>");
110     }
111 }
112 echo("<tr><td><input type='submit' value ='Submit Information'/></td></tr>");
113 echo("</form></table>");
114 }
115 // Consultation already exists
116 else
117 {
118     foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
119     {
120         echo("<form action='updateConsultation.php' method='post'>");
121         echo("<input type='hidden' name='doctor' value='{&#123;doctor&#125;}'>");
122         echo("<input type='hidden' name='date' value='{&#123;date&#125;}'>");
123         echo("<tr><td>SOAP_S:</td> <td><input type='text' value='{&#123;row['SOAP_S']&#125;}'<br>
124             name='SOAP_S'/></td></tr>");
125         echo("<tr><td>SOAP_O:</td> <td><input type='text' value='{&#123;row['SOAP_O']&#125;}'<br>
126             name='SOAP_O'/></td></tr>");
127         echo("<tr><td>SOAP_A:</td> <td><input type='text' value='{&#123;row['SOAP_A']&#125;}'<br>
128             name='SOAP_A'/></td></tr>");
129         echo("<tr><td>SOAP_P:</td> <td><input type='text' value='{&#123;row['SOAP_P']&#125;}'<br>
130             name='SOAP_P'/></td></tr>");
131         //Nurses already assigned to this consultation
132         if ($nrows2 != 0)
133         {
134             foreach($result2->fetchAll(PDO::FETCH_ASSOC) as $row2)
135             {
136                 echo("<input type='hidden' name='nurses_preassigned[]' value='{&#123;row2['VAT']&#125;}'>");
137                 echo("<tr><td>Assistant Nurses: </td> <td><input type='checkbox' name='<br>
138                     nurses_assigned[]' value='{&#123;row2['VAT']&#125;}' checked=<label>{&#123;row2['name']&#125;} {&#123;row2['VAT']&#125;}</label></td></tr>");
139             }
140         }
141         if ($nrows3 != 0)
142         {
143             foreach($result3->fetchAll(PDO::FETCH_ASSOC) as $row3)
144             {
145                 echo("<tr><td>Assistant Nurses: </td> <td><input type='checkbox' name='<br>
146                     nurse_list[]' value='{&#123;row3['VAT']&#125;}'><label>{&#123;row3['name']&#125;} {&#123;row3['VAT']&#125;}</label></td></tr>");
147             }
148         }
149         elseif ($nrows2 == 0 and $nrows3 == 0)
150         {
151             echo("<tr><td>ATTENTION: </td><td>No nurse/assistant available for this

```

```

        consultation</td></tr>");
146     }
147     // already created diagnosis
148     if($nrows_consult_diagnostic != 0)
149     {
150         foreach($result_consult_diagnostic->fetchAll(PDO::FETCH_ASSOC) as
            $diagnostic)
151         {
152             // check list of diagnostic codes
153             $id = $diagnostic['ID'];
154             echo("<input type='hidden' name='preassigned_codes[]' value='{${diagnostic
                ['ID']}'}>");
155             echo("<tr><td>Diagnostic Code: </td> <td><input type='checkbox' name='
                assigned_codes[]' value='{${diagnostic['ID']}' checked><label>{
                ${diagnostic['ID']}</label></td>");
156             echo("<td>Associated Prescriptions: </td></tr>");
157             // find medication already prescribed
158             $result_medication2 = $connection->prepare("SELECT * FROM prescription
                WHERE VAT_doctor = ? AND date_timestamp = ? AND ID = ?");
159             $result_medication2->execute(array($doctor,$date,$id));
160             $error6 = $result_medication2->errorInfo();
161             if ($error6[1] != ''){
162                 echo("<p>Error: {$error6[2]}</p>");
163             }
164
165             $nrows_medication2 = $result_medication2->rowCount();
166             if($nrows_medication2 != 0)
167             {
168                 foreach($result_medication2->fetchAll(PDO::FETCH_ASSOC) as $med)
169                 {
170                     $med_name = $med['name'];
171                     $med_lab = $med['lab'];
172                     $prevalue = array("name" => $med['name'], "lab" => $med['lab']);
173                     $value = serialize($prevalue);
174                     echo("<input type='hidden' name='meds_id_preassigned[$id][]' value='
                        $value'>");
175                     echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name
                        ='meds_id_assigned[$id][]' value='$value' checked><label>{
                        $med_name} {$med_lab}</label></td>");
176                     echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[
                        $id][$med_name][$med_lab]' value='{${med['dosage']}'}/></td>");
177                     echo("<td>Description:</td> <td><input type='text' name='description[
                        $id][$med_name][$med_lab]' value='{${med['description']}'}/></td></
                        tr>");
178                 }
179             }
180             // find medication not prescribed

```



```

181 $result_medication2 = $connection->prepare("SELECT * FROM medication
      WHERE (name,lab) NOT IN (SELECT name, lab FROM prescription WHERE
      VAT_doctor = ? AND date_timestamp = ? AND ID = ?)");
182 $result_medication2->execute(array($doctor,$date,$id));
183 $error7 = $result_medication2->errorInfo();
184 if ($error7[1] != ''){
185     echo("<p>Error: {$error7[2]}</p>");
186 }
187
188 $nrows_medication2 = $result_medication2->rowCount();
189 if($nrows_medication2 != 0)
190 {
191     foreach($result_medication2->fetchAll(PDO::FETCH_ASSOC) as $med)
192     {
193         $med_name = $med['name'];
194         $med_lab = $med['lab'];
195         $prevalue = array("name" => $med['name'], "lab" => $med['lab']);
196         $value = serialize($prevalue);
197         echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name
              ='meds_id[$id][']' value='$value'><label>{$med_name} {$med_lab}</
              label></td>");
198         echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[
              $id][\$med_name][\$med_lab]'/></td>");
199         echo("<td>Description:</td> <td><input type='text' name='description[
              $id][\$med_name][\$med_lab]'/></td></tr>");
200     }
201 }
202 }
203 }
204 // not yet created diagnosis
205 foreach($result_diagcodes->fetchAll(PDO::FETCH_ASSOC) as $diagnostic)
206 {
207     // check list of diagnostic codes
208     $id = $diagnostic['ID'];
209     echo("<tr><td>Diagnostic Code: </td> <td><input type='checkbox' name='codes
          [']' value='{\$diagnostic['ID']}'><label>{\$diagnostic['ID']}'</label></td>"
        );
210     // check list of medication per diagnostic
211     if ($nrows_medication != 0)
212     {
213         echo("<td>Associated Prescriptions: </td></tr>");
214         foreach($med_array as $med)
215         {
216             $med_name = $med['name'];
217             $med_lab = $med['lab'];
218             $value = serialize($med);
219             echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name='

```

```

220         meds_id[$id][] ' value='$value'><label>{$med_name} {$med_lab}</label>
            ></td>");
221     echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[$id]
        ][$med_name][$med_lab]'/></td>");
222     echo("<td>Description:</td> <td><input type='text' name='description[
        $id][$med_name][$med_lab]'/></td></tr>");
223     }
224     }
225     echo("<tr><td><input type='submit' value = 'Submit Information'></td></tr>");
226     echo("</form></table>");
227 }
228 }

```

3.4.1 Creating a Consultation

If the Consultation was not yet created the page created by the following code (Listing 8) will be used, this page will inform us how many rows were created.

Listing 8: Creating a Consultation (addConsultation.php)

```

1  $date = $_REQUEST['date'];
2  $doctor = $_REQUEST['doctor'];
3  $s = $_REQUEST['SOAP_S'];
4  $o = $_REQUEST['SOAP_O'];
5  $a = $_REQUEST['SOAP_A'];
6  $p = $_REQUEST['SOAP_P'];
7
8
9  $result = $connection->prepare("INSERT INTO consultation VALUES (?, ?, ?, ?, ?, ?)"
    );
10 $result->execute(array($doctor,$date,$s,$o,$a,$p));
11
12 $error = $result->errorInfo();
13 if ($error[1] != ''){
14     echo("<p>Error: {$error[2]}</p>");
15 }
16 else
17 {
18     echo("<p>Consultation Created</p>");
19 }
20
21
22 if(!empty($_REQUEST['nurse_list']))
23 {
24     foreach($_REQUEST['nurse_list'] as $nurse)
25     {

```

```

26     $result = $connection->prepare("INSERT INTO consultation_assistant VALUES (?,
27         ?, ?)");
28     $result->execute(array($doctor,$date,$nurse));
29
30     $error = $result->errorInfo();
31     if ($error[1] != ''){
32         echo("<p>Error: {$error[2]}</p>");
33     }
34     else
35     {
36         echo("<p>Nurses/Assistants Assigned</p>");
37     }
38 }
39
40 if(!empty($_REQUEST['codes']))
41 {
42     foreach($_REQUEST['codes'] as $code)
43     {
44
45         $result = $connection->prepare("INSERT INTO consultation_diagnostic VALUES (?,
46             ?, ?)");
47         $result->execute(array($doctor,$date,$code));
48
49         $error = $result->errorInfo();
50         if ($error[1] != ''){
51             echo("<p>Error: {$error[2]}</p>");
52         }
53         else
54         {
55             echo("<p>Diagnosis Created</p>");
56         }
57
58         //insert medication for each diagnosis
59         $med_id = $_REQUEST['meds_id'];
60         if(!empty($med_id[$code]))
61         {
62             foreach($med_id[$code] as $med)
63             {
64                 $med = unserialize($med);
65                 $name = $med['name'];
66                 $lab = $med['lab'];
67                 $dosage = $_REQUEST['dosage'];
68                 $dosage = $dosage[$code][$name][$lab];
69                 $description = $_REQUEST['description'];
70                 $description = $description[$code][$name][$lab];

```

```

71     $result = $connection->prepare("INSERT INTO prescription VALUES (?, ?, ?,?,
72         ?,?,?)");
73     $result->execute(array($name,$lab,$doctor,$date,$code,$dosage,$description)
74         );
75     $error = $result->errorInfo();
76     if ($error[1] != ''){
77         echo("<p>Error: {$error[2]}</p>");
78     }
79     else
80     {
81         echo("<p>Prescription Created</p>");
82     }
83 }
84 }
85 }
86 }

```

3.4.2 Updating a Consultation

When updating a consultation, there's more details then when simply creating it, here we have to consider what was already assigned to this consultation, thus what will be deleted, inserted or simply updated.

The check lists on Listing 7 always send to this page three arrays for each one of the attributes (nurses, prescriptions, medication), one with what was previously on the database, a second with what the user wanted to keep on the database, and even another with new things that the user wants to insert. Taking the difference of the first two arrays, it's possible to figure out what should be deleted, and what should be untouched (except for the case of medication where we have to do an update). The other array contains the items to be inserted.

For the medication is a little bit different since it is possible for the dosage and/or description to have been updated. For this case, instead of ignoring the prescriptions that the user wanted to keep, we do an update with the information corresponding to all of those prescriptions.

Listing 9: Updating a Consultation (updateConsultation.php)

```

1  $date = $_REQUEST['date'];
2  $doctor = $_REQUEST['doctor'];
3  $s = $_REQUEST['SOAP_S'];
4  $o = $_REQUEST['SOAP_O'];
5  $a = $_REQUEST['SOAP_A'];
6  $p = $_REQUEST['SOAP_P'];
7
8  $result = $connection->prepare("UPDATE consultation SET SOAP_S = ?, SOAP_O = ?,
9      SOAP_A = ?, SOAP_P = ? WHERE VAT_doctor = ? and date_timestamp = ?");
10 $result->execute(array($s,$o,$a,$p,$doctor,$date));

```

```
11 $error = $result->errorInfo();
12 if ($error[1] != ''){
13     echo("<p>Error: {$error[2]}</p>");
14 }
15 else{
16     echo("<p>Consultation updated</p>");
17 }
18
19 //nurses
20 if(!empty($_REQUEST['nurses_preassigned']))
21 {
22     //removed nurses from consultation
23     if(!empty($_REQUEST['nurses_assigned']))
24     {
25         $deletions = array_diff($_REQUEST['nurses_preassigned'], $_REQUEST['
            nurses_assigned']);
26     }
27     else
28     {
29         $deletions = $_REQUEST['nurses_preassigned'];
30     }
31     foreach($deletions as $deletion)
32     {
33         $result = $connection->prepare("DELETE FROM consultation_assistant WHERE
            VAT_doctor = ? and date_timestamp = ? and VAT_nurse = ?");
34         $result->execute(array($doctor,$date,$deletion));
35         $error = $result->errorInfo();
36         if ($error[1] != ''){
37             echo("<p>Error: {$error[2]}</p>");
38         }
39         else{
40             echo("<p>Assistants Removed</p>");
41         }
42     }
43 }
44 if(!empty($_REQUEST['nurse_list']))
45 {
46     //nurses added to consultation
47     foreach($_REQUEST['nurse_list'] as $nurse)
48     {
49         $result = $connection->prepare("INSERT INTO consultation_assistant VALUES (?,
            ?, ?)");
50         $result->execute(array($doctor,$date,$nurse));
51         $error = $result->errorInfo();
52         if ($error[1] != ''){
53             echo("<p>Error: {$error[2]}</p>");
54         }
55     }
56 }
```

```
55     else{
56         echo("<p>New Nurses/Assistants Assigned</p>");
57     }
58 }
59 }
60
61 //new diagnosis
62 if(!empty($_REQUEST['codes']))
63 {
64     foreach($_REQUEST['codes'] as $code)
65     {
66         $result = $connection->prepare("INSERT INTO consultation_diagnostic VALUES (?,
67             ?, ?)");
68         $result->execute(array($doctor,$date,$code));
69         $error = $result->errorInfo();
70         if ($error[1] != ''){
71             echo("<p>Error: {$error[2]}</p>");
72         }
73         else{
74             echo("<p>Diagnosis Created</p>");
75         }
76
77         //insert medication for each diagnosis
78         $med_id = $_REQUEST['meds_id'];
79         if(!empty($med_id[$code]))
80         {
81             foreach($med_id[$code] as $med)
82             {
83                 $med = unserialize($med);
84                 $name = $med['name'];
85                 $lab = $med['lab'];
86                 $dosage = $_REQUEST['dosage'];
87                 $dosage = $dosage[$code][$name][$lab];
88                 $description = $_REQUEST['description'];
89                 $description = $description[$code][$name][$lab];
90
91                 $result = $connection->prepare("INSERT INTO prescription VALUES (?, ?, ?,
92                     ?, ?, ?, ?)");
93                 $result->execute(array($name,$lab,$doctor,$date,$code,$dosage,$description)
94                     );
95                 $error = $result->errorInfo();
96                 if ($error[1] != ''){
97                     echo("<p>Error: {$error[2]}</p>");
98                 }
99                 else{
100                     echo("<p>Prescriptions added</p>");
101                 }
102             }
103         }
104     }
105 }
```

```

99     }
100   }
101 }
102 }
103
104 //already assigned diagnosis
105 if(!empty($_REQUEST['preassigned_codes']))
106 {
107     //check changes on the already assigned diagnosis
108     if(!empty($_REQUEST['assigned_codes'])) {
109         $deletions = array_diff($_REQUEST['preassigned_codes'], $_REQUEST['
            assigned_codes']);
110     } else {
111         $deletions = $_REQUEST['preassigned_codes'];
112     }
113     //deletions
114     foreach($deletions as $code)
115     {
116         //delete all prescriptions for this consultation_diagnostic
117         $result = $connection->prepare("DELETE FROM prescription WHERE VAT_doctor = ?
            AND date_timestamp = ? AND ID = ?");
118         $result->execute(array($doctor,$date,$code));
119         $error = $result->errorInfo();
120         if ($error[1] != '') {
121             echo("<p>Error: {$error[2]}</p>");
122         }
123         else {
124             echo("<p>Prescriptions Deleted</p>");
125         }
126
127         //delete consultation_diagnostic
128         $result = $connection->prepare("DELETE FROM consultation_diagnostic WHERE
            VAT_doctor = ? AND date_timestamp = ? AND ID = ?");
129         $result->execute(array($doctor,$date,$code));
130         $error = $result->errorInfo();
131         if ($error[1] != '') {
132             echo("<p>Error: {$error[2]}</p>");
133         }
134         else {
135             echo("<p>Diagnosis Deleted</p>");
136         }
137     }
138     //updates on preassigned diagnosis
139     $meds_assigned = $_REQUEST['meds_id_assigned'];
140     $meds_preassigned = $_REQUEST['meds_id_preassigned'];
141     foreach($_REQUEST['assigned_codes'] as $code)
142     {
143         if(!empty($meds_preassigned[$code]))

```

```

143 {
144     if(!empty($meds_assigned[$code])){
145         //updates to be made
146         foreach($meds_assigned[$code] as $update)
147         {
148             $med = unserialize($update);
149             $name = $med['name'];
150             $lab = $med['lab'];
151             $dosage = $_REQUEST['dosage'];
152             $dosage = $dosage[$code][$name][$lab];
153             $description = $_REQUEST['description'];
154             $description = $description[$code][$name][$lab];
155
156             $result = $connection->prepare("UPDATE prescription SET dosage = ?,
                description = ? WHERE name = ? AND lab = ? AND VAT_doctor = ? AND
                date_timestamp = ? AND ID = ?");
157             $result->execute(array($dosage,$description,$name,$lab,$doctor,$date,
                $code));
158             $error = $result->errorInfo();
159             if ($error[1] != ''){
160                 echo("<p>Error: {$error[2]}</p>");
161             }
162             else{
163                 echo("<p>Updates Completed</p>");
164             }
165         }
166         $deletions = array_diff($meds_preassigned[$code],$meds_assigned[$code]);}
167     else{
168         $deletions = $meds_preassigned[$code];}
169     //deletions to be made
170     foreach($deletions as $deletion)
171     {
172         $med = unserialize($deletion);
173         $name = $med['name'];
174         $lab = $med['lab'];
175
176
177         $result = $connection->prepare("DELETE FROM prescription WHERE name = ? AND
                lab = ? AND VAT_doctor = ? AND date_timestamp = ? AND ID = ?");
178         $result->execute(array($name,$lab,$doctor,$date,$code));
179         $error = $result->errorInfo();
180         if ($error[1] != ''){
181             echo("<p>Error: {$error[2]}</p>");
182         }
183         else{
184             echo("<p>Prescriptions Deleted</p>");
185         }

```



```
186     }
187   }
188   //insertions of meds for this diagnostic
189   $med_id = $_REQUEST['meds_id'];
190   if(!empty($med_id[$code]))
191   {
192     foreach($med_id[$code] as $med)
193     {
194       $med = unserialize($med);
195       $name = $med['name'];
196       $lab = $med['lab'];
197       $dosage = $_REQUEST['dosage'];
198       $dosage = $dosage[$code][$name][$lab];
199       $description = $_REQUEST['description'];
200       $description = $description[$code][$name][$lab];
201
202       $result = $connection->prepare("INSERT INTO prescription VALUES (?, ?, ?,
203                                     ?, ?, ?, ?)");
204       $result->execute(array($name,$lab,$doctor,$date,$code,$dosage,$description)
205                        );
206       $error = $result->errorInfo();
207       if ($error[1] != ''){
208         echo("<p>Error: {$error[2]}</p>");
209       }
210       else{
211         echo("<p>Prescriptions added</p>");
212       }
213     }
214   }
215 }
```

4 Web Application Using the Database - 3/3

4.1 Add procedure in consultation

On the previous section, it was created a page where the information of a consultation would be displayed. To this page, it was added a button where the user can add a particular type of procedure (in this case a dental chart), this button will redirect the user to a new page where the query will be done (*dental_charting.php*).

4.2 Charting measures

On this page, *dental_charting.php*, the user can insert the results of the dental charting, i.e. per-teeth measurements for the gap between the tooth and the gums and a description. For each dental chart, all the measures will be made in the context of the same transaction (listing 4.2)

```

1  $sql = "SELECT * from teeth";
2  $result = $connection->query($sql);
3  $nrows = $result->rowCount();
4  if ($nrows == 0)
5  {
6  echo("<p>erro!</p>");
7  }
8  else
9  {
10     echo("<form action='dental_charting.php' method='post'> <table>");
11     echo("<tr> <th>Name</th> <th>Quadrant</th> <th>Number</th> <th>Description</th>
        <th>Measure</th> <th></th> </tr>");
12     foreach($result as $row)
13     {
14         //insert description array
15         echo("<tr> ");
16         echo("<td>{$row['name']}</td> <td>{$row['quadrant']} </td> <td>{$row['
            number']} </td>");
17         echo("<td> <input type='text' name='description[{$row['quadrant']}][{$row
            ['number']}]' /> </td>");
18         echo("<td> <input type='number' step = '0.1' name='measure[{$row['quadrant
            ']}][{$row['number']}]' min = '0' /> </td>");
19         echo("<input type='hidden' value='$doctor' name='doctor' >");
20         echo("<input type='hidden' value='$name' name='name' >");
21         echo("<input type='hidden' value='$date' name='date' >");
22         echo("</tr>");
23     }
24     echo("</table> <p><input type='submit' value='new measure'> </form>");
25 }
26
27 $description = $_REQUEST['description'];
28 $measure = $_REQUEST['measure'];
29 $desc = 'default';
30
31
32
33 $result = $connection->prepare("insert into procedure_in_consultation values
    (?, ?, ?, ?)");
34 $result->execute(array($name, $doctor, $date, $desc));
35 $error = $result->errorInfo();
36 if ($error[1] == ''){
37     echo("<p>procedure in consultation added</p>");
38 }
39
40 try{
41     $connection->beginTransaction();

```

```

42
43     for($i=1;$i<=2;$i++){
44         for($j=1;$j<=3;$j++){
45             if(!empty($measure[$i][$j])){
46
47                 $result = $connection->prepare("insert into procedure_charting
48                     values (?, ?, ?,?,?,?,?)");
49                 $result2 = $connection->prepare("UPDATE procedure_charting SET
50                     description = ?, measure = ? WHERE name = ? and VAT = ? and
51                     date_timestamp = ? and quadrant=? and number = ?");
52
53                 if($result->execute(array($name, $doctor, $date,$i,$j,$description[
54                     $i][$j],$measure[$i][$j]))){
55                     echo("<p>New measure added</p>");
56                 }
57                 else if ($result2->execute(array($description[$i][$j],$measure[$i][
58                     $j],$name, $doctor, $date,$i,$j))){
59                     echo("<p>New measure updated</p>");
60                 }
61                 else{
62                     $error = $result->errorInfo();
63                     if ($error[1] != ''){
64                         echo("<p>Error: {$error[2]}</p>");
65                     }
66                 }
67             }
68         }
69     }
70
71     $connection->commit();
72 } catch(Exception $e) {
73     $connection->rollback();
74     throw $e;
75 }
76
77 $connection = null;

```

First of all, a query is done to print on the page all the tooth and make it possible to add information assigned to each teeth. The information regarding each teeth is stored in a 2D array according to its quadrant and number.

After that, the procedure in consultation is inserted in the database.

At last, a transaction is started and all the measures inserted will be committed at the same time. If by any chance, already exists a measure for that teeth, instead of a insert into the database it will be done an update

5 Functions, Triggers and Stored Procedures

5.1 Trigger: Age Update

For the age update, the *timestampdiff()* function does the trick, subtracting the birth date from the current date and returning the value in years. So every time there is an insert on appointment for a certain client, this calculation is made, updating the age of that specific client.

Listing 10: Age Update

```
1 delimiter $$
2 create trigger update_age after insert on appointment
3 for each row
4 begin
5     update client
6     set age = timestampdiff(year, birth_date, now())
7     where VAT = new.VAT_client;
8 end $$
9 delimiter ;
```

5.2 Trigger: Same individual, Multiple jobs

5.2.1 (a) Nurses, Receptionists and Doctors

In other to prevent nurses and receptionists of also being doctors, and vice-versa, there were created triggers that fire whenever there are an update or an insert on one of this tables, that's why there are two triggers for each table. The triggers simply signal an error that aborts the query whenever one of this prohibited actions occur.

Listing 11: (a) Nurses and Receptionists can't be Doctors

```
1 — insert nurse
2 delimiter $$
3 create trigger check_nurse_insert before insert on nurse
4 for each row
5 begin
6     if exists(select VAT
7         from doctor where VAT = new.VAT) then
8         signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the
9             same VAT';
10    end if;
11 end $$
12 — update nurse
13 delimiter $$
14 create trigger check_nurse_update before update on nurse
15 for each row
16 begin
```

```
17  if exists(select VAT
18  from doctor where VAT = new.VAT) then
19  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the
    same VAT';
20  end if;
21 end $$
22 delimiter ;
23 — insert receptionist
24 delimiter $$
25 create trigger check_receptionist_insert before insert on receptionist
26 for each row
27 begin
28  if exists(select VAT
29  from doctor where VAT = new.VAT) then
30  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the
    same VAT';
31  end if;
32 end $$
33 delimiter ;
34 — update receptionist
35 delimiter $$
36 create trigger check_receptionist_update before update on receptionist
37 for each row
38 begin
39  if exists(select VAT
40  from doctor where VAT = new.VAT) then
41  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the
    same VAT';
42  end if;
43 end $$
44 delimiter ;
45 — insert doctor
46 delimiter $$
47 create trigger check_doctor_insert before insert on doctor
48 for each row
49 begin
50  if exists(select VAT
51  from nurse where VAT = new.VAT) then
52  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a nurse with the
    same VAT';
53  end if;
54  if exists(select VAT
55  from receptionist where VAT = new.VAT) then
56  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a receptionist with
    the same VAT';
57  end if;
58 end $$
```

```

59 delimiter ;
60 — update doctor
61 delimiter $$
62 create trigger check_doctor_update before update on doctor
63 for each row
64 begin
65     if exists(select VAT
66         from nurse where VAT = new.VAT) then
67         signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a nurse with the
68             same VAT';
69     end if;
70     if exists(select VAT
71         from receptionist where VAT = new.VAT) then
72         signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a receptionist with
73             the same VAT';
74     end if;
75 end $$
76 delimiter ;

```

5.2.2 (b) Permanent Doctors vs Trainees

What was explained on (a) happens here, except for the fact that the considered tables are the one for permanent doctors, and the one for trainees.

Listing 12: (b) Doctors can't be both Permanent and Trainees

```

1 — insert permanent_doctor
2 delimiter $$
3 create trigger check_permanent_insert before insert on permanent_doctor
4 for each row
5 begin
6     if exists(select VAT
7         from trainee_doctor where VAT = new.VAT) then
8         signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a trainee with the
9             same VAT';
10    end if;
11 end $$
12 delimiter ;
13 — update permanent_doctor
14 delimiter $$
15 create trigger check_permanent_update before update on permanent_doctor
16 for each row
17 begin
18     if exists(select VAT
19         from trainee_doctor where VAT = new.VAT) then
20         signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a trainee with the
21             same VAT';

```

```

20     end if;
21 end $$
22 delimiter ;
23 — insert trainee_doctor
24 delimiter $$
25 create trigger check_trainee_insert before insert on trainee_doctor
26 for each row
27 begin
28     if exists(select VAT
29         from permanent_doctor where VAT = new.VAT) then
30         signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a permanent doctor
31             with the same VAT';
32     end if;
33 end $$
34 delimiter ;
35 — update trainee_doctor
36 delimiter $$
37 create trigger check_trainee_update before update on trainee_doctor
38 for each row
39 begin
40     if exists(select VAT
41         from permanent_doctor where VAT = new.VAT) then
42         signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a permanent doctor
43             with the same VAT';
44     end if;
45 end $$
46 delimiter ;

```

5.3 Trigger: Repeated Phone Numbers

Since we can't allow repeated phone numbers, it is necessary to fire an error signal whenever there are an insertion or an update on the table for employees' phone numbers and on the table for clients' phone numbers. That being said, the rationale behind the trigger is the same as in the previous section.

Listing 13: Triggers for Repeated Phone Numbers

```

1 — on insert phone
2 delimiter $$
3 create trigger check_phone_insert before insert on phone_number_employee
4 for each row
5 begin
6     if exists(select phone
7         from phone_number_employee where phone = new.phone and VAT != new.VAT) then
8         signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
9             belongs to a employee';
10    end if;

```

```
10  if exists(select phone
11  from phone_number_client where phone = new.phone and VAT != new.VAT) then
12  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
    belongs to a client';
13  end if;
14  end $$
15  delimiter ;
16  — on update phone
17  delimiter $$
18  create trigger check_phone_update before update on phone_number_employee
19  for each row
20  begin
21  if exists(select phone
22  from phone_number_employee where phone = new.phone and VAT != new.VAT) then
23  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
    belongs to a employee';
24  end if;
25  if exists(select phone
26  from phone_number_client where phone = new.phone and VAT != new.VAT) then
27  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
    belongs to a client';
28  end if;
29  end $$
30  delimiter ;
31  — on insert phone
32  delimiter $$
33  create trigger check_phone_insert_client before insert on phone_number_client
34  for each row
35  begin
36  if exists(select phone
37  from phone_number_employee where phone = new.phone and VAT != new.VAT) then
38  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
    belongs to a employee';
39  end if;
40  if exists(select phone
41  from phone_number_client where phone = new.phone and VAT != new.VAT) then
42  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
    belongs to a client';
43  end if;
44  end $$
45  delimiter ;
46  — on update phone
47  delimiter $$
48  create trigger check_phone_update_client before update on phone_number_client
49  for each row
50  begin
51  if exists(select phone
```



```

52  from phone_number_employee where phone = new.phone and VAT != new.VAT) then
53  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
    belongs to a employee';
54  end if;
55  if exists(select phone
56  from phone_number_client where phone = new.phone and VAT != new.VAT) then
57  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
    belongs to a client';
58  end if;
59  end $$
60  delimiter ;

```

5.4 Function: Number of *No Shows*

This function aims to compute the number of appointments that did not result in a consultation given some parameters, in order to do that it's necessary to first search every appointment that fulfils those requisitions. After that just count the number of appointments whose primary keys do not exist on the consultations table.

Listing 14: No Shows

```

1  delimiter $$
2  create function no_shows(c_gender char(2), a_year varchar(4), max_age int, min_age
    int)
3  returns integer
4  begin
5      declare ns_count integer;
6      select count(*) into ns_count
7      from appointment
8      where VAT_client in (select VAT
9      from client where
10     age between max_age and min_age
11     and gender = c_gender)
12     and (VAT_doctor, date_timestamp) not in
13     (select VAT_doctor, date_timestamp from consultation)
14     and extract(year from date_timestamp) = a_year;
15     return ns_count;
16 end$$
17 delimiter ;

```

5.5 Procedure: Salary Change

On this procedure we do two updates, one for doctors that had more than 100 consultations, and the other for the opposite case, besides that we just have to check the number of years of practice for each permanent doctor and consider only the consultations for the current year.

Listing 15: Doctors Salary Change

```
1 delimiter $$
2 create procedure salary_raise(in x_years integer)
3 begin
4     update employee
5     set salary = salary + salary*0.05
6     where VAT in (select VAT
7     from permanent_doctor
8     where years > x_years)
9     and VAT not in (select VAT_doctor
10    from consultation
11    where extract(year from date_timestamp) = year(curdate()))
12    group by VAT_doctor
13    having count(*) > 100);
14
15    update employee
16    set salary = salary + salary*0.1
17    where VAT in (select VAT
18    from permanent_doctor
19    where years > x_years)
20    and VAT in (select VAT_doctor
21    from consultation
22    where extract(year from date_timestamp) = year(curdate()))
23    group by VAT_doctor
24    having count(*) > 100);
25 end$$
26 delimiter ;
```