

AUTHORS

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

November 24, 2022

Using the Database Contents

Contents

1	Introduction	1
2	Web Application Using the Database - 1/3 2.1 Search a client	1 1 2
	2.3 Create a new client	2 3
3		4 6 8 9 16 18
4	4.1 Add procedure in consultation	23 23 23
5	5.1 Trigger: Age Update	26 26 26 26 28 29 31 31
	5.5 Procedure, Salary Change	

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

```
if(!empty($VAT)||!empty($Name)||!empty($Street)||!empty($City)||!empty($Zip))
1
   {
2
       $sVAT = "%{$_REQUEST['VAT']}%";
3
       $sName = "%{$_REQUEST['Name']}%";
4
       $sStreet = "%{$_REQUEST['Street']}%";
5
       $sCity = "%{$_REQUEST['City']}%";
6
       $sZip = "%{$_REQUEST['Zip']}%";
       $result = $connection->prepare("SELECT VAT, name FROM client WHERE client.VAT
8
          LIKE ? and client.name LIKE ? and client.street LIKE ? and client.city LIKE
          ? and client.zip LIKE ?");
       $result->execute(array($sVAT, $sName, $sStreet, $sCity, $sZip));
9
10
11
       if ($result->rowCount() == 0)
13
14
       echo("There is no client with such info.");
15
16
       else
17
       {
           echo("");
19
           echo(" Name VAT  ");
20
           foreach($result->fetchAll(PD0::FETCH_ASSOC) as $row)
21
           {
22
               echo("<form action='newAppointment.php' method='post'>");
23
               echo("{$row['name']} <input type='hidden' name='VAT' value</pre>
24
                  ='{$row['VAT']}'>{$row['VAT']} ");
               echo(" <input type='submit' value='Add Appointment'/> ");
25
```

```
echo("</form>");
26
         echo("<form action='listAppointments.php' method='post'>");
27
               echo("<input type='hidden' name='VAT' value='{$row['VAT']}'>")
28
               echo("<input type='submit' value='List Appointements/Consultations</pre>
29
                   '/>");
               echo("</form> ");
30
31
32
           echo("");
33
34
       }
35
   }
```

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



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 newAppoint-ments.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

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


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).



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 newAppoint-ment.php.

Listing 3: Check Availability

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

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 Doctors Available Doctors 2019-12-17 10:00 Date: 17 / 12 / 2019 ♥ 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 66666666

TYPE	DAT	E	DOCTOR	
Appointment	2019-01-19	17:55:00	123456789	Show Details
Consultation	2019-01-19	17:55:00	123456789	Show Details
Appointment	2019-02-09	15:16:00	123456789	Show Details
Appointment	2019-02-12	12:02:00	123456789	Show Details
Appointment	2019-02-27	16:48:00	123456789	Show Details
Appointment	2019-03-02	13:53:00	123456789	Show Details

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

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

```
$VAT = $_REQUEST['VAT'];
1
2
   $result = $connection->prepare("SELECT * FROM appointment NATURAL LEFT OUTER JOIN
3
      consultation where VAT_client = ? order by date_timestamp");
   $result->execute(array($VAT));
4
5
   $nrows = $result->rowCount();
6
   if ($nrows == 0)
7
8
     echo("There are no records for this client.");
9
   }
10
   else
11
   {
12
     echo("");
13
     echo(" TYPE DATE DOCTOR ");
14
     foreach($result->fetchAll(PD0::FETCH_ASSOC) as $row)
15
16
     {
       $date = $row['date_timestamp'];
17
       $doctor = $row['VAT_doctor'];
18
       echo("");
19
       20
       echo("<form action='appointment.php' method='post'>");
21
       echo("<input type='hidden' name='date' value='{$date}'>");
22
       echo("<input type='hidden' name='doctor' value='{$doctor}'>");
23
       echo("<input type='hidden' name='description' value='{$row['description']}'>");
24
       echo("<input type='hidden' name='client' value='{$VAT}'>");
25
       echo("<input type='submit' value='Show Details'/>");
26
       echo("</form>");
27
       echo("");
28
       if (!is_null($row['SOAP_S']) or !is_null($row['SOAP_O']) or !is_null($row['
29
          SOAP_A']) or !is_null($row['SOAP_P']))
       {
30
         echo("");
31
         echo("Consultation {$row['date_timestamp']} {$row['
32
            VAT_doctor']}");
         echo("<form action='consultation.php' method='post'>");
33
         echo("<input type='hidden' name='date' value='{$date}'>");
34
         echo("<input type='hidden' name='doctor' value='{$doctor}'>");
35
         echo("<input type='hidden' name='soap_s' value='{$row['SOAP_S']}'>");
36
         echo("<input type='hidden' name='soap_o' value='{$row['SOAP_0']}'>");
37
         echo("<input type='hidden' name='soap_a' value='{$row['SOAP_A']}'>");
38
         echo("<input type='hidden' name='soap_a' value='{$row['SOAP_P']}'>");
39
         echo("<input type='submit' value='Show Details'/>");
40
         echo("</form>");
41
         echo("");
42
       }
43
```

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)

```
$\footnote{\text{sdate} = \sum_{REQUEST['doctor'];}}
$\footnote{\text{sdate} = \sum_{REQUEST['date'];}}
$\footnote{\text{sdate} = \sum_{REQUEST['date'];}}
$\footnote{\text{secho("VAT doctor: \sum_{doctor");}}}
$\footnote{\text{echo("VAT doctor: \sum_{doctor");}}}
$\footnote{\text{echo("SOAP_S: \sum_{REQUEST['date']}");}}
$\footnote{\text{echo("SOAP_O: \sum_{REQUEST['soap_o']}");}}
$\footnote{\text{secho("SOAP_O: \sum_{REQUEST['soap_o']}");}}
$\footnote{\text{sec
```

```
echo("SOAP_A: {$_REQUEST['soap_a']}");
10
   echo("SOAP_P: {$_REQUEST['soap_p']}");
11
12
13
   $result = $connection->prepare("select * from consultation_diagnostic where
14
      VAT_doctor = ? and date_timestamp = ? ");
   $result->execute(array($doctor, $date));
15
16
17
   $nrows = $result->rowCount();
18
   if (snrows == 0)
19
20
   echo("There is no Diagnosis.");
21
   }
22
   else
23
   {
24
       foreach($result->fetchAll(PD0::FETCH_ASSOC) as $row)
25
       {
26
           echo("");
27
           echo("Diagnosis ID : {$row['ID']}");
28
29
30
           $result2 = $connection—>prepare("select * from prescription as p where p.
              VAT_doctor=? and p.date_timestamp = ? group by p.name");
           $result2->execute(array($doctor, $date));
32
           $nrows2 = $result->rowCount();
33
           if($nrows2 == 0)
34
           {
35
              echo("");
36
              echo("No prescription");
37
              echo("");
38
39
           }
40
           else
41
           {
42
              foreach($result2—>fetchAll(PD0::FETCH_ASSOC) as $presc)
43
              {
44
                  echo("");
45
                  echo("name, lab: {$presc['name']}, {$presc['lab']} ");
46
                  echo("");
47
                  echo("");
48
                  echo(" dosage: {$presc['dosage']}");
49
                  echo("");
50
                  echo("");
51
                  echo(" description: {$presc['description']}");
52
                  echo("");
53
              }
54
```

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

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: 66666666

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)

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

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

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: SOAP O: SOAP A: SOAP P: Assistant Nurses: ☐ Jane Dentedoce 123746789 Assistant Nurses: ☐ Enfermeira Joana 154235485 Diagnostic Code: D000 Associated Prescriptions: medication lab 1 Dosage Description medication2 lab1 Dosage Description nalmada mae Dosage: Description: □D105 Diagnostic Code: Associated Prescriptions medication1 lab1 Dosage: Description: medication2 lab1 Dosage: Description palmada mae Description: Diagnostic Code: □ D106 Associated Prescriptions medication1 lab1 Dosage: Description: medication2 lab1 Dosage Description palmada mae Dosage: Description:

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)

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

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

```
echo("<form action='addConsultation.php' method='post'>");
69
     echo("<input type='hidden' name='doctor' value='{$doctor}'>");
70
     echo("<input type='hidden' name='date' value='{$date}'>");
71
     echo("SOAP_S: <input type='text' name='SOAP_S'/>");
72
     echo("SOAP_0: <input type='text' name='SOAP_0'/>");
73
     echo("SOAP_A:<input type='text' name='SOAP_A'/>");
74
     75
     // if there are nurses available
76
     if ($nrows3 != 0)
77
78
79
      foreach($result3->fetchAll(PD0::FETCH_ASSOC) as $row3)
      {
80
        echo("Assistant Nurses: <input type='checkbox' name='</pre>
81
           nurse_list[]' value='{$row3['VAT']}'><label>{$row3['name']} {$row3['VAT']}'
           ']}</label>");
      }
82
     }
83
     else
84
85
      echo("ATTENTION: No nurse/assistant available for this
86
         consultation");
     }
87
     foreach($result_diagcodes->fetchAll(PDO::FETCH_ASSOC) as $diagnostic)
89
      // check list of diagnostic codes
90
      $id = $diagnostic['ID'];
91
      echo("Diagnostic Code:  <input type='checkbox' name='codes[]'</pre>
92
         value='{$diagnostic['ID']}'><label>{$diagnostic['ID']}</label>");
      // check list of medication per diagnostic
93
      if ($nrows_medication != 0)
94
      {
95
96
        echo("Associated Prescriptions: ");
        foreach($med_array as $med)
97
        {
98
          $med_name = $med['name'];
99
          $med_lab = $med['lab'];
100
          $value = serialize($med);
101
          echo("
102
            meds_id[$id][]' value='$value'><label>{$med_name} {$med_lab}</label>
            >");
          103
            $med_name][$med_lab]'/>");
          echo("Description: <input type='text' name='description[$id][</pre>
104
            $med_name][$med_lab]'/>");
105
        }
      }
106
      else
107
```

```
{
108
         echo("No Medication on the DataBase");
109
        }
110
111
      echo("<input type='submit' value ='Submit Information'/>");
112
      echo("</form>");
113
    }
114
    // Consultation already exists
115
    else
116
    {
117
118
      foreach($result->fetchAll(PD0::FETCH_ASSOC) as $row)
119
      {
       echo("<form action='updateConsultation.php' method='post'>");
120
         echo("<input type='hidden' name='doctor' value='{$doctor}'>");
121
         echo("<input type='hidden' name='date' value='{$date}'>");
122
         echo("SOAP_S: <input type='text' value='{$row['SOAP_S']}'</pre>
             name='SOAP_S'/>");
         echo("SOAP_0: <input type='text' value='{$row['SOAP_0']}'</pre>
124
             name='SOAP_0'/>");
         echo("SOAP_A: <input type='text' value='{$row['SOAP_A']}'</pre>
125
             name='SOAP_A'/>");
         echo("SOAP_P: <input type='text' value='{$row['SOAP_P']}'</pre>
126
             name='SOAP_P'/>");
         //Nurses already assigned to this consultation
127
         if ($nrows2 != 0)
128
          {
129
           foreach($result2—>fetchAll(PD0::FETCH_ASSOC) as $row2)
130
131
             echo("<input type='hidden' name='nurses_preassigned[]' value='{$row2['VAT</pre>
132
                 ']}'>");
             echo("Assistant Nurses: <input type='checkbox' name='</pre>
133
                nurses_assigned[]' value='{$row2['VAT']}' checked><label>{$row2['name
                 ']} {$row2['VAT']}</label>");
           }
134
         }
135
         if ($nrows3 != 0)
136
137
           foreach($result3->fetchAll(PD0::FETCH_ASSOC) as $row3)
138
139
             echo("Assistant Nurses: <input type='checkbox' name='</pre>
140
                nurse_list[]' value='{$row3['VAT']}'><label>{$row3['name']} {$row3['
                VAT']}</label>");
           }
141
142
143
         elseif ($nrows2 == 0 and $nrows3 == 0)
144
          {
           echo("ATTENTION: No nurse/assistant available for this
145
```

```
consultation");
            }
146
            // already created diagnosis
147
            if($nrows_consult_diagnostic != 0)
148
149
              foreach($result_consult_diagnostic->fetchAll(PDO::FETCH_ASSOC) as
150
                  $diagnostic)
              {
151
                 // check list of diagnostic codes
152
                 $id = $diagnostic['ID'];
153
                 echo("<input type='hidden' name='preassigned_codes[]' value='{$diagnostic</pre>
154
                     ['ID']}'>");
                 echo("Diagnostic Code:  <input type='checkbox' name='</pre>
155
                     assigned_codes[]' value='{$diagnostic['ID']}' checked><label>{
                     $diagnostic['ID']}</label>");
                 echo("Associated Prescriptions: ");
                 // find medication already prescribed
157
                 $result_medication2 = $connection—>prepare("SELECT * FROM prescription
158
                     WHERE VAT_doctor = ? AND date_timestamp = ? AND ID = ?");
                 $result_medication2->execute(array($doctor,$date,$id));
159
                 $error6 = $result_medication2—>errorInfo();
160
                 if ($error6[1] != ''){
161
                   echo("Error: {$error6[2]}");
                 }
163
164
                 $nrows_medication2 = $result_medication2->rowCount();
165
                 if($nrows_medication2 != 0)
166
                 {
167
                   foreach($result_medication2—>fetchAll(PD0::FETCH_ASSOC) as $med)
                      $med_name = $med['name'];
170
171
                      med_lab = med['lab'];
                      $prevalue = array("name" => $med['name'], "lab" => $med['lab']);
172
                      $value = serialize($prevalue);
173
                     echo("<input type='hidden' name='meds_id_preassigned[$id][]' value='</pre>
174
                         $value'>");
                     echo("sinput type='checkbox' name
                         ='meds_id_assigned[$id][]' value='$value' checked><label>{
                         $med_name} {$med_lab}</label>");
                     echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo
176
                         $id][$med_name][$med_lab]' value='{$med['dosage']}'/>");
                     echo("Description: <input type='text' name='description[</pre>
177
                         $id][$med_name][$med_lab]' value='{$med['description']}'/>
                         tr>");
178
                   }
                 }
179
                 // find medication not prescribed
180
```

```
$result_medication2 = $connection->prepare("SELECT * FROM medication
181
                    WHERE (name, lab) NOT IN (SELECT name, lab FROM prescription WHERE
                    VAT_doctor = ? AND date_timestamp = ? AND ID = ?)");
                $result_medication2->execute(array($doctor,$date,$id));
182
                $error7 = $result_medication2->errorInfo();
183
                if ($error7[1] != ''){
184
                   echo("Error: {$error7[2]}");
185
                }
186
                $nrows_medication2 = $result_medication2->rowCount();
188
                if($nrows_medication2 != 0)
189
190
                   foreach($result_medication2—>fetchAll(PD0::FETCH_ASSOC) as $med)
191
                   {
192
                     $med_name = $med['name'];
193
                     $med_lab = $med['lab'];
194
                     $prevalue = array("name" => $med['name'], "lab" => $med['lab']);
195
                     $value = serialize($prevalue);
196
                     echo("sinput type='checkbox' name
197
                         ='meds_id[$id][]' value='$value'><label>{$med_name} {$med_lab}</
                         label>");
                     echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("echo("<t
198
                         $id][$med_name][$med_lab]'/>");
                     echo("Description: <input type='text' name='description[</pre>
199
                         $id][$med_name][$med_lab]'/>");
                   }
200
                }
201
              }
202
203
            }
            // not yet created diagnosis
204
            foreach($result_diagcodes->fetchAll(PDO::FETCH_ASSOC) as $diagnostic)
205
206
              // check list of diagnostic codes
207
              $id = $diagnostic['ID'];
208
              echo("Diagnostic Code:  <input type='checkbox' name='codes
209
                  []' value='{$diagnostic['ID']}'><label>{$diagnostic['ID']}</label>"
                  );
              // check list of medication per diagnostic
210
              if ($nrows_medication != 0)
211
              {
212
                echo("Associated Prescriptions: ");
213
                foreach($med_array as $med)
214
                {
215
                   $med_name = $med['name'];
216
217
                   $med_lab = $med['lab'];
                   $value = serialize($med);
218
                   219
```

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

```
$date = $_REQUEST['date'];
    $doctor = $_REQUEST['doctor'];
2
    s = \text{REQUEST['SOAP_S']};
3
   so = s_REQUEST['SOAP_0'];
4
    a = \text{REQUEST['SOAP_A']};
5
    p = \text{REQUEST['SOAP_P']};
6
7
8
   $result = $connection->prepare("INSERT INTO consultation VALUES (?, ?, ?, ?, ?)"
9
    $result->execute(array($doctor,$date,$s,$o,$a,$p));
10
11
    $error = $result->errorInfo();
12
   if ($error[1] != ''){
      echo("Error: {$error[2]}");
14
   }
15
   else
16
17
      echo("Consultation Created");
18
   }
19
20
21
22
   if(!empty($_REQUEST['nurse_list']))
23
      foreach($_REQUEST['nurse_list'] as $nurse)
24
      {
25
```

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

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

```
$date = $_REQUEST['date'];
1
    $doctor = $_REQUEST['doctor'];
2
    s = \text{REQUEST['SOAP_S']};
3
    so = s_REQUEST['SOAP_0'];
4
    a = \REQUEST['SOAP_A'];
5
    p = \text{REQUEST['SOAP_P']};
6
7
   $result = $connection->prepare("UPDATE consultation SET SOAP_S = ?, SOAP_O = ?,
8
       SOAP_A = ?, SOAP_P = ? WHERE VAT_doctor = ? and date_timestamp = ?");
    $result->execute(array($s,$o,$a,$p,$doctor,$date));
9
10
```

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

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

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

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

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

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)

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

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

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

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

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

    update nurse

12
   delimiter $$
13
    create trigger check_nurse_update before update on nurse
   for each row
15
   begin
16
```

```
if exists(select VAT
17
      from doctor where VAT = new.VAT) then
18
      signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the
19
         same VAT';
     end if;
20
   end $$
21
   delimiter;
22

    insert receptionist

23
   delimiter $$
24
    create trigger check_receptionist_insert before insert on receptionist
25
26
   begin
27
      if exists(select VAT
28
      from doctor where VAT = new.VAT) then
29
      signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the
30
         same VAT';
     end if;
32
   end $$
   delimiter;
33

    update receptionist

34
   delimiter $$
35
   create trigger check_receptionist_update before update on receptionist
36
   for each row
37
   begin
38
      if exists(select VAT
39
      from doctor where VAT = new.VAT) then
40
      signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the
41
         same VAT':
     end if;
42
   end $$
43
   delimiter ;
44
     - insert doctor
45
   delimiter $$
46
    create trigger check_doctor_insert before insert on doctor
47
   for each row
48
   begin
49
      if exists(select VAT
50
      from nurse where VAT = new.VAT) then
51
      signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a nurse with the
52
         same VAT';
     end if;
53
      if exists(select VAT
54
      from receptionist where VAT = new.VAT) then
      signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a receptionist with
56
         the same VAT';
      end if;
57
   end $$
58
```

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

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

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

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

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

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

```
if exists(select phone
10
      from phone_number_client where phone = new.phone and VAT != new.VAT) then
11
      signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
12
         belongs to a client';
      end if;
13
   end $$
14
   delimiter;
15

    on update phone

16
   delimiter $$
17
    create trigger check_phone_update before update on phone_number_employee
19
   begin
20
      if exists(select phone
21
      from phone_number_employee where phone = new.phone and VAT != new.VAT) then
22
      signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
23
         belongs to a employee';
     end if;
24
      if exists(select phone
25
      from phone_number_client where phone = new.phone and VAT != new.VAT) then
26
      signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
27
         belongs to a client';
      end if;
28
   end $$
29
   delimiter;
30

    on insert phone

31
   delimiter $$
32
    create trigger check_phone_insert_client before insert on phone_number_client
33
    for each row
34
   begin
35
      if exists(select phone
36
      from phone_number_employee where phone = new.phone and VAT != new.VAT) then
37
      signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
38
         belongs to a employee';
     end if;
39
      if exists(select phone
40
      from phone_number_client where phone = new.phone and VAT != new.VAT) then
41
      signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists,
42
         belongs to a client';
      end if;
43
   end $$
44
   delimiter;
45

    on update phone

46
   delimiter $$
47
    create trigger check_phone_update_client before update on phone_number_client
48
49
    for each row
   beain
50
      if exists(select phone
51
```

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

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

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

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