

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ II

Θέμα: Υλοποίηση ΒΔ και εφαρμογής διαχείρισης
ασφαλιστικών προϊόντων



**Εργαστήριο 2020-2021, Τμήμα Δευτέρας
Κος Ε. Κοσμάτος**

“Υλοποίηση της διαδικτυακής εφαρμογής (portal)”

Εργασία 3
Ημ. Παράδοσης: 31 Δεκ. 2020

ΒΕΛΑΣΚΟ ΠΑΟΛΑ
ΚΟΥΜΠΟΥΡΗ ΜΑΡΓΑΡΙΤΑ

161020
18390109

Πίνακας περιεχομένων

| | |
|---|-----------|
| ΣΚΟΠΟΣ ΕΡΓΑΣΙΑΣ | 3 |
| ΕΙΣΑΓΩΓΗ | 3 |
| ΥΛΟΠΟΙΗΣΗ ΕΡΓΑΣΙΑΣ | 4 |
| 1. Δημιουργία νέου χρήστη | 4 |
| 2. Τροποποίηση κώδικα – procedure | 5 |
| 3. Ερωτήματα εργασίας | 5 |
| Εντολές..... | 5 |
| Το αρχείο index.html | 5 |
| Αποτέλεσμα | 6 |
| http://localhost/General%20Insurance/..... | 6 |
| Βήμα 9 | 7 |
| Εντολές..... | 7 |
| Το αρχείο view_products.html | 7 |
| Τεκμηρίωση..... | 9 |
| Αποτελέσματα..... | 9 |
| http://localhost/General%20Insurance/view_products.php..... | 9 |
| Βήμα 10 | 10 |
| Εντολές..... | 10 |
| Το αρχείο menuproducts.html | 10 |
| Το αρχείο prod_per_client.php | 11 |
| Τεκμηρίωση..... | 12 |
| Αποτελέσματα..... | 13 |
| http://localhost/General%20Insurance/menuproducts.html..... | 13 |
| http://localhost/General%20Insurance/prod_per_client.php..... | 13 |
| Βήμα 11 | 14 |
| Εντολές..... | 14 |
| Το αρχείο clientoption.html | 14 |
| Το αρχείο clientpayment.php..... | 15 |
| Τεκμηρίωση..... | 16 |

| | |
|---|-----------|
| Αποτελέσματα..... | 17 |
| http://localhost/General%20Insurance/clientoption.html | 17 |
| http://localhost/General%20Insurance/client_payment.php | 17 |
| ΣΥΜΠΕΡΑΣΜΑΤΑ..... | 18 |
| ΠΑΡΑΡΤΗΜΑ | 19 |
| Κώδικας/ Εντολές – Εργ. Άσκηση 2..... | 19 |

ΣΚΟΠΟΣ ΕΡΓΑΣΙΑΣ

Ο σκοπός της τρίτης τελικής εργαστηριακής άσκησης είναι να μας φέρει σε πρώτη επαφή με τη δημιουργία μιας διαδικτυακής εφαρμογής σε συνδυασμό με τις ήδη υπάρχουσες γνώσεις που έχουμε πάνω στις Βάσεις Δεδομένων. Συγκεκριμένα, μας ζητείται να δημιουργήσουμε μια ιστοσελίδα για τη διαχείριση των ασφαλιστικών προϊόντων της ΒΔ, με χρήση των τεχνολογιών PHP και HTML.

ΕΙΣΑΓΩΓΗ

Αρχικά θα δημιουργήσουμε 3 σελίδες .html

1. Index.html
2. Menuproducts.html
3. Clientoption.html

εκ των οποίων η μια θα αποτελεί έναν 'πίνακα περιεχομένων' ο οποίος θα οδηγεί με ένα κουμπί σε μια νέα σελίδα, η κάθε μια από τις οποίες θα αποτελεί ένα από τα βήματα της εργασίας. Η κάθε μία από τις σελίδες που ανοίγει έχει ως επιλογή είτε να επιστρέψει στη σελίδα περιεχομένων είτε να ξαναεπαναλάβει την ενέργεια, την οποία είναι καθορισμένη να κάνει.

Επίσης έχουμε και 3 κώδικες php

1. Client_payment.php
2. Prod_per_client.php
3. View_products.php

οι οποίοι εκτελούν τα queries και δημιουργούν τους πίνακες που περιγράφονται από την εκφώνηση.

Αξίζει να σημειώσουμε πως για να 'τρέξουμε' τα αρχεία χρειάζεται να έχουμε κάνει start στον **Apache** και να έχουμε τα αρχεία σε ένα φάκελο ενιαίο, που θα βρίσκεται στον φάκελο htdocs του xampp.

ΥΛΟΠΟΙΗΣΗ ΕΡΓΑΣΙΑΣ

1. Δημιουργία νέου χρήστη

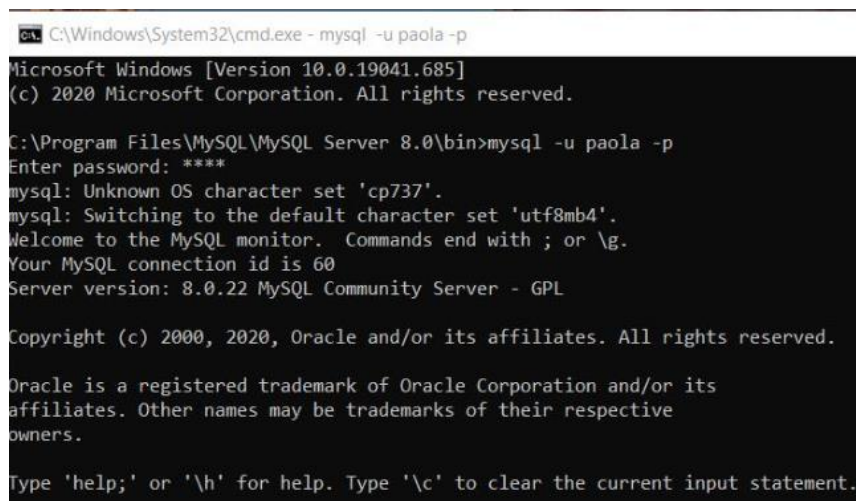
Για να πραγματοποιήσουμε τη σύνδεση μεταξύ της βάσης και του server, χρειάζεται να δημιουργήσουμε έναν καινούργιο χρήστη, στον οποίο θα αποδώσουμε όλα τα δικαιώματα έτσι ώστε να μπορεί να διαχειριστεί τη βάση, χωρίς περιορισμούς.

Η δημιουργία του χρήστη γίνεται με την εξής εντολή:

```
create user '<username>'@'localhost' identified by '1234';  
grant all privileges on general_insurance.* to  
'<username>'@'localhost';  
alter user <username>@localhost identified with  
mysql_native_password by '1234';
```

Στη συνέχεια για να συνδεθούμε ως χρήστης στη βάση δεδομένων απλά εκτελούμε την εξής εντολή στο cmd:

```
mysql -u <username> -p
```



2. Τροποποίηση κώδικα – procedure

Για το τρίτο βήμα της εργασίας θα χρειαστούμε ένα κομμάτι της εργασίας 2 στο οποίο υπολογίζεται το μηνιαίο ποσό των συμβολαίων που έχει υπογράψει ο πελάτης, καθώς και ο αριθμός των συμβολαίων που κατέχει εκείνο τον μήνα. Ο κώδικας είναι ο παρακάτω για το procedure που αναφέραμε:

```
drop procedure getPayment;
delimiter $$
create procedure getPayment(in afmc int, in datie date)
begin
select count(*) as number_of_contracts, ifnull(sum(annual_cost/12),0)
as monthly_cost
from insurance inner join contract
on insurance.insurance_code = contract.insurance_code
where datie between start_date and end_date
and AFM = afmc;
end $$
delimiter ;
```

3. Ερωτήματα εργασίας

Εντολές

Η hostpage της ασφαλιστικής εταιρείας «General Insurance» υλοποιείται ως εξής:

Το αρχείο index.html

```
<html>
  <head>
    <title>General Insurance</title>
    <center></center>
  </head>

  <body style='background-color:rgb(230, 210, 238) '>
    <h1>
      <center>Welcome to General Insurance</center>
    </h1>
    <center>
      <form name="insert" action="view_products.php" method="POST">
        <pre>

          </pre>
          <pre><input type="submit" name="submit" value="View products"
/></pre>
          <pre><p><b>To see the products "General Insurance"
provides</b></p>
```

```

    </pre>
  </form>

  <form name="insert" action="menuproducts.html" method="POST">
    <pre><input type="submit" name="submit" value="View products"
/></pre>
    <pre><p><b>See customers info by product</b></p>

    </pre>
  </form>

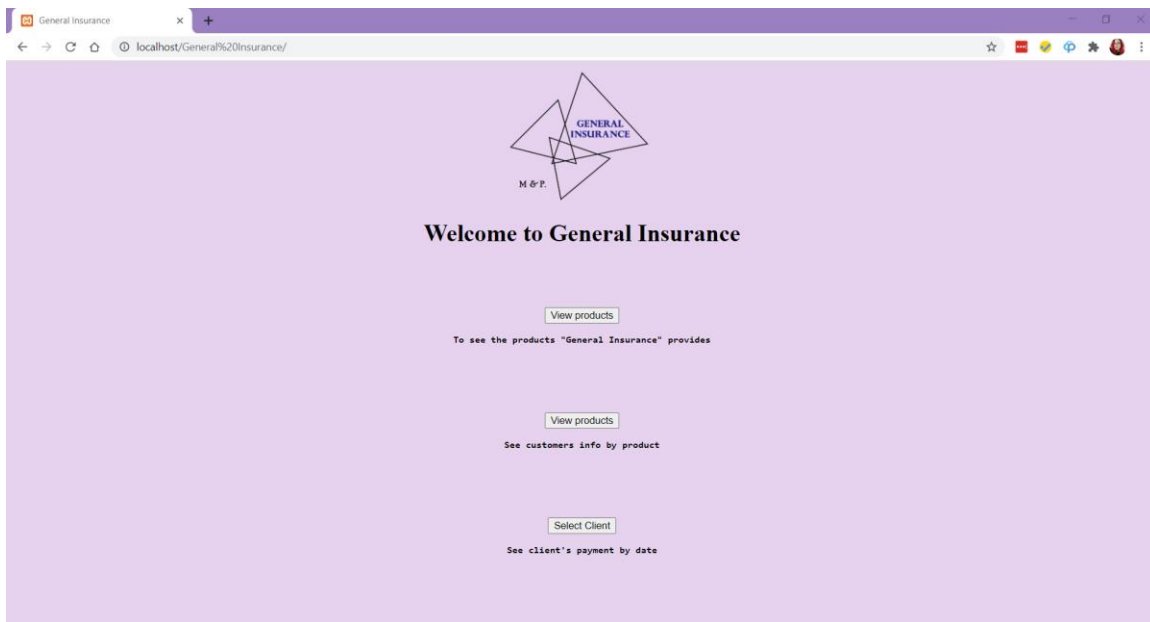
  <form name="insert" action="clientoption.html" method="POST">
    <pre><input type="submit" name="submit" value="Select Client"
/></pre>
    <pre><p><b>See client's payment by date</b></p>

    </pre>
  </form>
</center>
</body>
</html>

```

Αποτέλεσμα

<http://localhost/General%20Insurance/>



Βήμα 9

Δημιουργήστε μια σελίδα στην οποία θα παρουσιάζονται όλα τα ασφαλιστικά προϊόντα.

Εντολές

Το αρχείο view_products.html

```
<html>

<head>
    <title>General Insurance</title>
    <center></center>
</head>

<body>
<h1>
    <center>General Insurance</center>
</h1>
<?php
echo "<body style='background-color:rgb(241, 229, 247)'">";
$mysqli = mysqli_connect("localhost", "paola", "1234",
"general_insurance");
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_errno());
    exit();
} else {

    $sql = "SELECT * FROM insurance";
    //echo $sql;

    $res = mysqli_query($mysqli, $sql);
    if ($res == TRUE) {
        echo "<center><table border = '1'">";
        echo "<tr>
<th>Insurance Code</th>
<th>Insurance name</th>
<th>Annual cost</th>
<th>Minimum duration</th>
</tr>";
        while ($newArray = mysqli_fetch_array($res, MYSQLI_BOTH)) {
            $insurance_code = $newArray['insurance_code'];
            $insurance_name = $newArray['insurance_name'];
            $annual_cost = $newArray['annual_cost'];
            $min_duration = $newArray['min_duration'];

            echo "<tr><td><center>" . $insurance_code .
"</center></td><td><center>" . $insurance_name .
"</center></td><td><center>" . $annual_cost .
"</center></td><td><center>" . $min_duration . "</center></td></tr>";
```



```
        }

        echo "</table></center>";
    } else {
        echo "Error: Cannot retrieve insurance table data";
    }

    mysqli_close($mysqli);
}

?>
<p>

</p>
<a href="http://localhost/General%20Insurance/"><center>Go to main
page</center></a>

</body>

</html>
```

Τεκμηρίωση

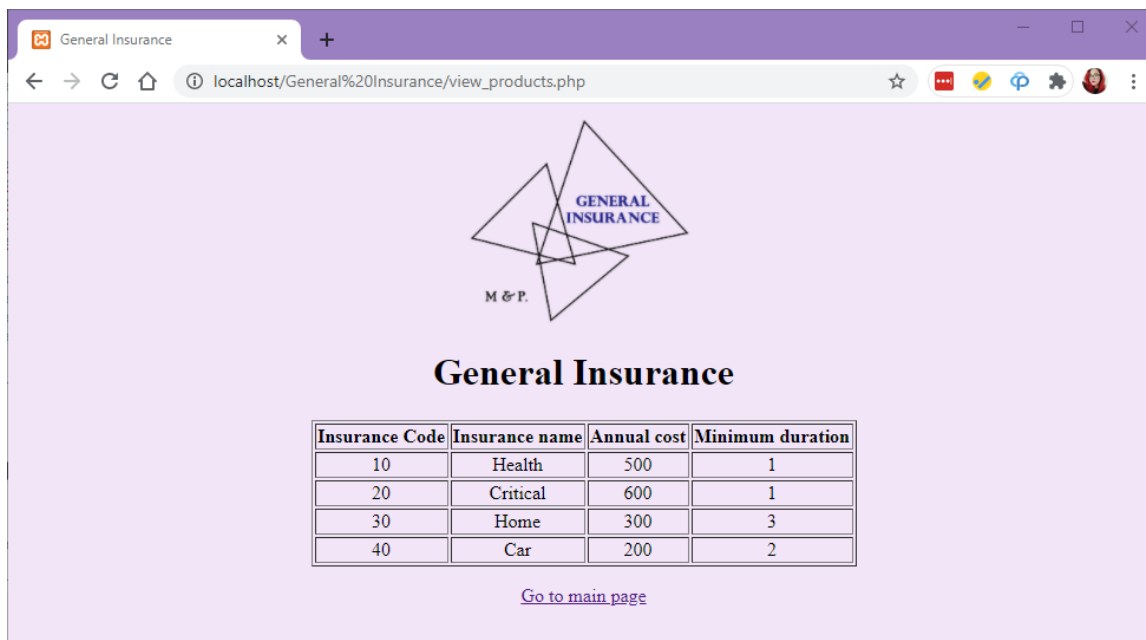
Το αρχείο **index.html** στο πρώτο πεδίο form, οδηγεί στην εκτέλεση του αρχείου **view_products.php**. Το τελευταίο, παρουσιάζει τα προϊόντα τα οποία προσφέρει η ασφαλιστική εταιρία εκτελώντας το query:

```
SELECT * FROM insurance
```

Στην καινούργια σελίδα, παρατίθεται ο πίνακας με τα προϊόντα της ασφαλιστικής εταιρίας.

Αποτελέσματα

http://localhost/General%20Insurance/view_products.php



| Insurance Code | Insurance name | Annual cost | Minimum duration |
|----------------|----------------|-------------|------------------|
| 10 | Health | 500 | 1 |
| 20 | Critical | 600 | 1 |
| 30 | Home | 300 | 3 |
| 40 | Car | 200 | 2 |

[Go to main page](#)

Βήμα 10

Δημιουργήστε μια σελίδα στην οποία, ο χρήστης μπορεί να επιλέγει ένα ασφαλιστικό προϊόν από μια *drop down list*. Αφού επιλέξει προϊόν θα του παρουσιάζονται οι πελάτες (στοιχεία τους) που έχουν υπογράψει ένα τουλάχιστον συμβόλαιο στο επιλεγμένο ασφαλιστικό προϊόν.

Εντολές

Το αρχείο `menuproducts.html`

```
<html>
  <head>
    <title>General Insurance</title>
    <center></center>
  </head>

  <body style="background-color: rgb(192, 154, 207)">
    <h1>
      <center>General Insurance</center>
    </h1>

    <form action="prod_per_client.php" method="POST">
      <pre>

      </pre>
      <center>
        <label for="products">Choose a product:</label>
        <select name="products" id="products">
          <option value="Health">Health</option>
          <option value="Critical">Critical</option>
          <option value="Home">Home</option>
          <option value="Car">Car</option>
        </select>
        <input type="submit" name="submit" value="View client" />
      </center>
    </form>

    <p>

    </p>

    <a
href="http://localhost/General%20Insurance/"><center>Back</center></a>
  </body>
</html>
```

Το αρχείο prod_per_client.php

```
<html>

<head>
    <title>General Insurance</title>
    <center></center>
</head>

<body>
    <h1>
        <center>General Insurance</center>
    </h1>

    <?php

        echo "<body style='background-color:rgb(241, 229, 247)'>";
        $mysqli = mysqli_connect("localhost", "paola", "1234",
"general_insurance");
        if (mysqli_connect_errno()) {
            printf("Connect failed: %s\n", mysqli_connect_errno());
            exit();
        } else {

            echo "<pre><h3><center>" . $_POST['products'] .
"</center></h3></pre>";

            $sql = "SELECT customer.name,
customer.surname, customer.phone, customer.address, customer.doy FROM
contract
            INNER JOIN insurance ON insurance.insurance_code =
contract.insurance_code
            INNER JOIN customer ON customer.AFM = contract.AFM
            where insurance_name = '" . $_POST['products'] . "'";

            // echo $sql;

            $res = mysqli_query($mysqli, $sql);
            if ($res == TRUE) {

                echo "<center><table border = '1' width='700'>";
                echo "<tr>
                <th>Name</th>
                <th>Surname</th>
                <th>Phone</th>
                <th>Address</th>
                <th>DOY</th>
                </tr>";

                while ($newArray = mysqli_fetch_array($res, MYSQLI_BOTH)) {
                    $name = $newArray['name'];
```

```

        $sur = $newArray['surname'];
        $pho = $newArray['phone'];
        $add = $newArray['address'];
        $dou = $newArray['doy'];
        echo "<tr><td><center>" . $name .
"</center></td><td><center>" . $sur . "</center></td><td><center>" .
$pho . "</center></td><td><center>" . $add .
"</center></td><td><center>" . $dou . "</center></td></tr>";
    }

    echo "</table></center>";
} else {
    echo "Error: Cannot retrieve client's data";
}

mysqli_close($mysqli);
}

?>
<p>

</p>
<a href="http://localhost/General%20Insurance/menuproducts.html">
    <center>Choose another product</center>
</a>
<p><a href="http://localhost/General%20Insurance/">
    <center>Go to main page</center>
</a></p>

</body>

</html>

```

Τεκμηρίωση

Το αρχείο **index.html** στο δεύτερο πεδίο form, οδηγεί στην εκτέλεση του αρχείου **menuproducts.html** το οποίο υλοποιεί τον κώδικα **prod_per_client.php**. Το τελευταίο, παρουσιάζει τους πελάτες που έχουν τουλάχιστον ένα συμβόλαιο από το πεδίο που επιλέχθηκε από το drop down menu εκτελώντας το query:

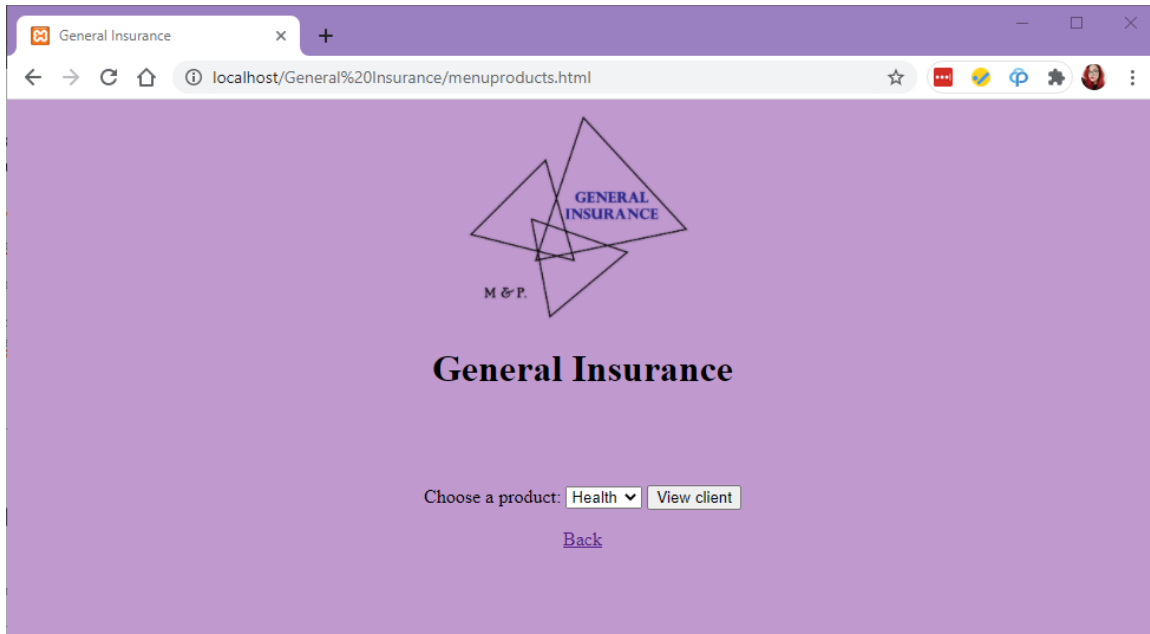
```

"SELECT customer.name,
customer.surname, customer.phone, customer.address, customer.doy FROM
contract
    INNER JOIN insurance ON insurance.insurance_code =
contract.insurance_code
    INNER JOIN customer ON customer.AFM = contract.AFM
where insurance_name = '". $_POST['products'] . "'";

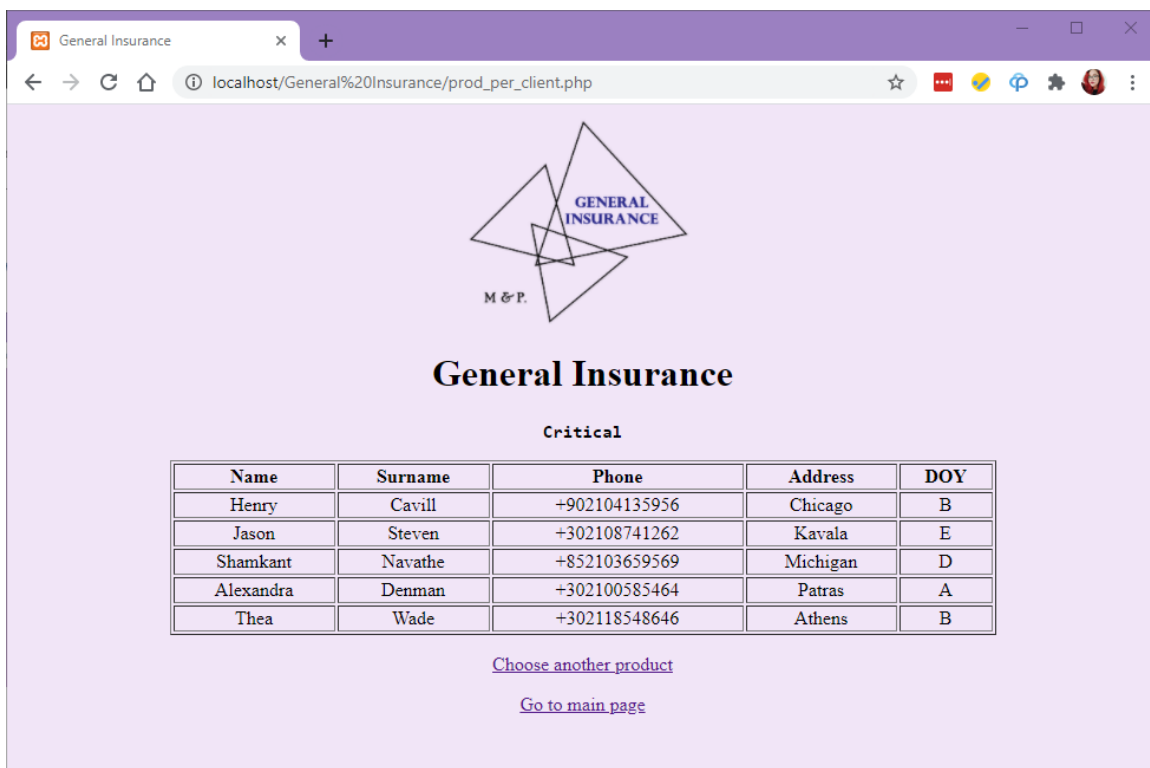
```

Αποτελέσματα

<http://localhost/General%20Insurance/menuproducts.html>



http://localhost/General%20Insurance/prod_per_client.php



Βήμα 11

Δημιουργήστε μια σελίδα στην οποία, ο χρήστης θα εισάγει το ΑΦΜ ενός πελάτη και μια ημερομηνία και θα επιστρέφει τον αριθμό των συμβολαίων που κατέχει εκείνο το μήνα, καθώς επίσης και το συνολικό ποσό που πρέπει να πληρώσει το συγκεκριμένο μήνα.

Εντολές

Οι εντολές που πραγματοποιήσαμε είναι οι εξής

Το αρχείο clientoption.html

```
<html>

<head>
  <title>General Insurance</title>
  <center></center>
</head>

<body style="background-color: rgb(192, 154, 207)">
  <h1>
    <center>General Insurance</center>
  </h1>

  <form name="insert" action="client_payment.php" onsubmit="return
validateForm()" method="POST" required>
    <center>
      <p>
        Insert Client's AFM:
        <input type="text" name="input_afm" size="10" />
        Insert date:
        <input type="text" name="input_date" size="10" />
        <input type="submit" name="submit" value="View client" />
      </p>
    </center>
  </form>

  <p></p>
  <a href="http://localhost/General%20Insurance/">
    <center>Back</center>
  </a>
</body>

</html>
```

Το αρχείο clientpayment.php

```
<html>

<head>
    <title>General Insurance</title>
    <center></center>
</head>

<body>
    <h1>
        <center>General Insurance</center>
    </h1>
    <?php
        echo "<body style='background-color:rgb(241, 229, 247)'">";

        $mysqli = mysqli_connect("localhost", "paola", "1234",
"general_insurance");
        if (mysqli_connect_errno()) {
            printf("Connect failed: %s\n", mysqli_connect_errno());
            exit();
        } else {

            $sql = " call getPayment( " . $_POST['input_afm'] . ", ' " .
$_POST['input_date'] . "')";

            $res = mysqli_query($mysqli, $sql);
            if ($res == TRUE) {
                echo "<center><table border = '1'">";
                echo "<tr>
<th>Number of contracts</th>
<th>Montly Cost</th>
</tr>";

                while ($newArray = mysqli_fetch_array($res, MYSQLI_BOTH)) {
                    $input_afm = $newArray['number_of_contracts'];
                    $input_date = $newArray['monthly_cost'];

                    echo "<tr align=\"center\"><td>" . $input_afm .
"</td><td>" . $input_date . "</td></tr>";
                }

                echo "</table></center>";
            } else {
                echo "Error: Cannot retrieve client' data";
            }

            mysqli_close($mysqli);
            //echo "Hello " . $_POST["stud_am"] . " " . $_POST["stud_name"]
```



```
. " " . $_POST["stud_semester"];  
}  
  
?>  
<center>  
  <p>  
  
  </p>  
  
  <a  
href="http://localhost/General%20Insurance/clientoption.html">Choose  
another client or date</a>  
  <p><a href="http://localhost/General%20Insurance/">Go to main  
page</a></p>  
  </center>  
  
</body>  
</html>
```

Τεκμηρίωση

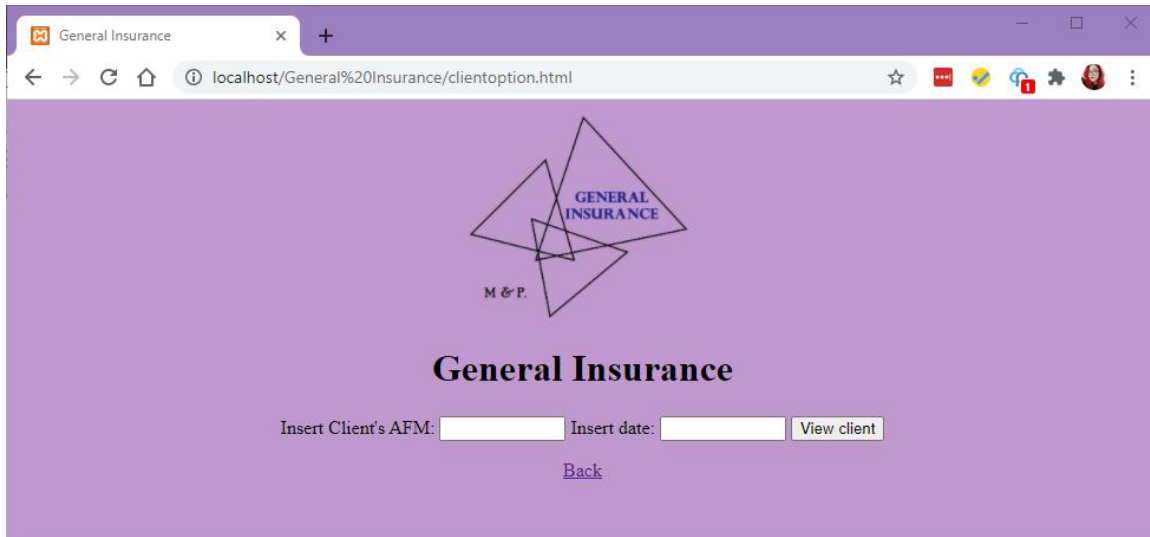
Το αρχείο **index.html** στο τρίτο πεδίο form, οδηγεί στην εκτέλεση του αρχείου **clientoption.html** το οποίο υλοποιεί τον κώδικα **client_payment.php**. Το τελευταίο λειτουργεί ανάλογα με το procedure που είχαμε δημιουργήσει στην εργασία 2, το οποίο υπολογίζει τον αριθμό των συμβολαίων που κατέχει εκείνο τον μήνα, και το ποσό που πρέπει να πληρώσει.

Η εντολή που τρέχει είναι η εξής:

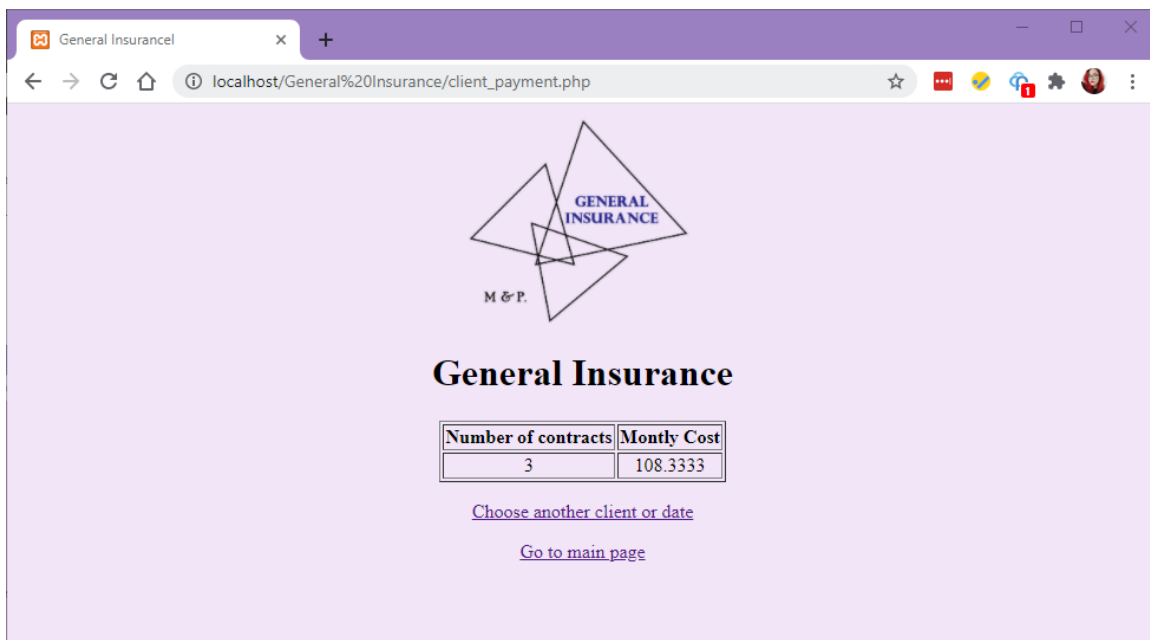
```
call getPayment( " . $_POST['input_afm'] . ", ' " . $_POST['input_date']  
  . "' );
```

Αποτελέσματα

<http://localhost/General%20Insurance/clientoption.html>



http://localhost/General%20Insurance/client_payment.php



ΣΥΜΠΕΡΑΣΜΑΤΑ

Με την διεκπεραίωση της άσκησης αυτής, εμπεδώσαμε καλύτερα τη λειτουργία των βάσεων σε ένα περιβάλλον όπου έπρεπε να συνδυαστούν οι ήδη υπάρχουσες γνώσεις μας πάνω στη διαχείριση των βάσεων δεδομένων με το web developing. Είδαμε και κατανοήσαμε την υλοποίηση καθώς και τον συνδυασμό των γλωσσών html (γλώσσα σήμανσης για τις ιστοσελίδες) και php (γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο). Τέλος, μάθαμε να μορφοποιούμε τις σελίδες κατάλληλα με αισθητικά κριτήρια.

ΠΑΡΑΡΤΗΜΑ

Κώδικας/ Εντολές – Εργ. Άσκηση 2

```
# ΒΕΛΑΣΚΟ ΠΑΟΛΑ - 161020
# ΚΟΥΜΠΟΥΡΗ ΜΑΡΓΑΡΙΤΑ - 18390109
# Εργασία 2 - ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ 2

DROP DATABASE IF EXISTS General_Insurance;
CREATE DATABASE General_Insurance;
USE General_Insurance;

#table customer
CREATE TABLE customer
(
    AFM int(30) not null,
    name varchar(30),
    surname varchar(30),
    address varchar(30),
    DOY varchar(4),
    phone varchar(13),
    PRIMARY KEY (AFM),
    CONSTRAINT phone_check CHECK (REGEXP_LIKE(phone, '^[+][0-9]{12}$'))
);

#table coverage ( type of service coverage )
CREATE TABLE coverage
(
    coverage_code int not null auto_increment,
    coverage_name varchar(70),
    PRIMARY KEY (coverage_code)
);

#table insurance
CREATE TABLE insurance
(
    insurance_code int not null,
    insurance_name varchar(60),
    annual_cost int(5),
    min_duration int not null,
    PRIMARY KEY (insurance_code),
    CONSTRAINT cost_check CHECK (annual_cost > 0)
);

#table insurance_coverages
CREATE TABLE insurance_coverages
(
    insurance_code int not null,
    coverage_code int not null auto_increment,
    PRIMARY KEY (insurance_code, coverage_code),
```

```

    FOREIGN KEY (insurance_code) REFERENCES insurance (insurance_code),
    FOREIGN KEY (coverage_code) REFERENCES coverage (coverage_code)
);

#table contract
CREATE TABLE contract
(
    contract_code int not null,
    cost int(5),
    start_date date,
    END_date date,
    AFM int(30) not null,
    insurance_code int not null,
    PRIMARY KEY (contract_code),
    FOREIGN KEY (AFM) REFERENCES customer (AFM),
    FOREIGN KEY (insurance_code) REFERENCES insurance (insurance_code)
);

#table insurance_customer
CREATE TABLE insurance_customer
(
    insurance_code int not null,
    AFM int(30) not null,
    PRIMARY KEY (insurance_code, AFM),
    FOREIGN KEY (AFM) REFERENCES customer (AFM),
    FOREIGN KEY (insurance_code) REFERENCES insurance (insurance_code)
);

#show tables of database
show tables;

#DESCRIBE tables of General_Insurance
DESCRIBE contract;
DESCRIBE customer;
DESCRIBE insurance;
DESCRIBE coverage;
DESCRIBE insurance_coverages;
DESCRIBE insurance_customer;

SELECT * FROM customer;
#INSERT INTO customer
INSERT INTO customer (AFM, name, surname, phone, address, DOY)
VALUES
    (1023452569, 'Charlie', 'Hunnam', '+602103625956', 'California',
    'A'),
    (1445525690, 'Henry', 'Cavill', '+902104135956', 'Chicago', 'B'),
    (1785592569, 'Orlando', 'Bloom', '+302103681956', 'Athens', 'C'),
    (1785527877, 'James', 'Smith', '+302103765556', 'Thessaloniki',
    'D'),
    (1884560847, 'Jason', 'Steven', '+302108741262', 'Kavala', 'E'),
    (1542147114, 'Paola', 'Velasco', '+302114522704', 'Athens', 'A'),
    (1445547451, 'Margarita', 'Koumpouri', '+302115475668', 'Athens',

```

```

'B'),
  (1445477457, 'Ramez', 'Elmasri', '+662108544615', 'Arlington', 'C'),
  (1024782463, 'Shamkant', 'Navathe', '+852103659569', 'Michigan',
'D'),
  (1841204754, 'Maria', 'Papadopoulou', '+302117941554', 'Patras',
'A'),
  (1784585254, 'Kostas', 'Pappas', '+302135874572', 'Lamia', 'B'),
  (1745242426, 'Andreas', 'Dimitrios', '+302108564524', 'Volos', 'I'),
  (1415541574, 'Tatiana', 'Raptis', '+302105448542', 'Thessaloniki',
'Z'),
  (1124834578, 'Periklis', 'Megalos', '+302115858645', 'Volos', 'A'),
  (1486542484, 'Liliana', 'Beckham', '+902138524656', 'Chicago', 'C'),
  (1946923008, 'HENDERSON', 'Gibb', '+602126799446', 'California',
'C'),
  (1256452238, 'Alexandra', 'Denman', '+302100585464', 'Patras', 'A'),
  (1592227645, 'Thea', 'Wade', '+302118548646', 'Athens', 'B');

#INSERT INTO coverage
INSERT INTO coverage (coverage_code, coverage_name)
VALUES
  (1, 'medicine');

SELECT * FROM coverage;

INSERT INTO coverage (coverage_name)
VALUES
  ('maternity'),
  ('funeral'),
  ('accident'),
  ('pharmacy'),
  ('repair'),
  ('transplant'),
  ('vacation cover for terminal people'),
  ('property');

SELECT * FROM coverage;

SELECT * FROM insurance;

#INSERT INTO insurance
INSERT INTO insurance (insurance_code, insurance_name, annual_cost,
min_duration)
VALUES (10, 'Health', 500, 1),
  (20, 'Critical', 600, 1),
  (30, 'Home', 300, 3),
  (40, 'Car', 200, 2);
SELECT * FROM insurance;

SELECT * FROM insurance_coverages;

#INSERT INTO insurance_coverages
INSERT INTO insurance_coverages (insurance_code, coverage_code)

```

```
VALUES
```

```
(10, 1),  
(10, 2),  
(10, 4),  
(20, 1),  
(20, 3),  
(30, 6),  
(30, 9),  
(40, 4),  
(40, 6);
```

```
DELETE FROM contract;  
SELECT * FROM contract;
```

```
#INSERT INTO contract DATE (YYYY-M-D)
```

```
INSERT INTO contract
```

```
(contract_code, cost, start_date, END_date, AFM, insurance_code)
```

```
VALUES
```

```
(512, 11000, '2000-1-1', '2022-1-1', 1023452569, 10),  
(513, 1800, '2006-1-1', '2009-1-1', 1445525690, 20),  
(514, 4000, '2004-1-1', '2006-1-1', 1785592569, 30),  
(515, 900, '2004-1-1', '2010-1-1', 1785592569, 10),  
(516, 1000, '2005-1-1', '2010-1-1', 1884560847, 10),  
(517, 2500, '2005-3-5', '2014-3-5', 1884560847, 20),  
(518, 1800, '2008-3-6', '2020-3-6', 1884560847, 30),  
(519, 2000, '2011-6-4', '2017-6-4', 1884560847, 40),  
(520, 1200, '2011-6-4', '2015-6-4', 1542147114, 40),  
(521, 2000, '2011-11-9', '2014-11-9', 1445547451, 10),  
(522, 900, '2011-7-5', '2015-7-5', 1445477457, 30),  
(523, 1500, '2012-3-8', '2015-3-8', 1024782463, 10),  
(524, 1800, '2012-8-9', '2019-8-9', 1024782463, 20),  
(525, 900, '2013-9-5', '2019-9-5', 1841204754, 30),  
(526, 900, '2013-9-5', '2018-9-5', 1784585254, 30),  
(527, 900, '2015-11-10', '2022-11-10', 1745242426, 30),  
(528, 1200, '2016-12-10', '2019-12-10', 1415541574, 40),  
(529, 400, '2017-4-30', '2023-4-30', 1124834578, 40),  
(530, 1200, '2017-6-2', '2020-6-2', 1486542484, 40),  
(531, 1500, '2017-6-3', '2025-6-3', 1946923008, 10),  
(532, 4800, '2017-6-4', '2026-6-4', 1256452238, 20),  
(533, 4000, '2018-9-5', '2026-9-5', 1592227645, 10),  
(534, 4800, '2018-9-5', '2026-9-5', 1592227645, 20),  
(535, 1800, '2018-9-5', '2024-9-5', 1592227645, 30),  
(536, 1600, '2018-9-5', '2026-9-5', 1592227645, 40);
```

```
#INSERT INTO insurance_customer
```

```
INSERT INTO insurance_customer (insurance_code, AFM)
```

```
VALUES
```

```
(10, 1023452569),  
(10, 1785592569),  
(10, 1884560847),  
(20, 1445525690),  
(20, 1884560847),
```

```
(30, 1884560847),
(30, 1784585254),
(40, 1486542484),
(40, 1592227645);

#show current contents of tables
SELECT * FROM contract;
SELECT * FROM customer;
SELECT * FROM insurance;
SELECT * FROM insurance_coverages;
SELECT * FROM insurance_customer;

#updatable view (unsafe)
DROP VIEW IF EXISTS customerInfo;

CREATE VIEW customerInfo
AS SELECT name, surname, address FROM customer;

SELECT * FROM customerInfo;

UPDATE customerInfo SET address = "Dublin" WHERE name = "Ramez" AND
surname = "Elmasri";

SELECT * FROM customerInfo;
SELECT * FROM customer;

#updatable view (unsafe with certain join)
DROP VIEW IF EXISTS customerContract_view;
CREATE VIEW customerContract_view
AS SELECT customer.AFM, name, surname
FROM customer INNER JOIN contract ON customer.AFM = contract.AFM;

SELECT * FROM customerContract_view;
SELECT * FROM customer;

UPDATE customerContract_view SET name = "Stefan" WHERE AFM =
1023452569;
SELECT * FROM customerContract_view;

#updatable view (safe)
DROP VIEW IF EXISTS customers_view3;
CREATE VIEW customers_view3 (AFM, name, surname, address, DOY, phone)
AS SELECT * FROM customer
WHERE DOY in ("A", "C")
WITH CHECK OPTION;
```



```

INSERT INTO customers_view3
VALUES
    (178663326, 'John', 'Velasco', 'Athens', 'C', '+302115055293');
SELECT * FROM customers_view3;
SELECT * FROM customer;

INSERT INTO customers_view3 ( AFM, name, surname, phone, address, DOY )
VALUES
    (177452826, 'Christopher', 'Velasco', '+102115783293', 'London', 'L');

#non updatable view
DROP VIEW IF EXISTS cust_distinct_names;

CREATE VIEW cust_distinct_names(name)
AS SELECT DISTINCT name FROM customer ORDER BY name;

SELECT * FROM cust_distinct_names;
INSERT INTO cust_distinct_names VALUES ('Gerard');

# Show how many contracts have been signed for each insurance
SELECT insurance_name, count(*) AS 'contracts signed'
FROM contract INNER JOIN insurance
ON contract.insurance_code = insurance.insurance_code
GROUP BY contract.insurance_code;

# Show the clients according to the total of the contracts that they
have signed (in descending order)
SELECT DISTINCT name, surname, sum(cost) AS total
FROM customer INNER JOIN contract
ON customer.AFM = contract.AFM
GROUP BY customer.AFM
ORDER BY total DESC;

#*****TRIGGERS*****

ALTER TABLE customer ADD (con_sum INT(3));
UPDATE customer SET con_sum =
    (SELECT IFNULL(sum(cost),0) FROM contract WHERE customer.AFM =
    contract.AFM);

SELECT * FROM customer;
SELECT * FROM contract;

#*****trigger: after insert*****

DROP TRIGGER IF EXISTS contract_insert;

delimiter //
CREATE TRIGGER contract_insert
AFTER INSERT ON contract
FOR EACH ROW
BEGIN
    UPDATE customer

```

```
        SET con_sum = con_sum + NEW.cost
        WHERE customer.AFM = NEW.AFM;
END;
//
delimiter ;

INSERT INTO contract VALUES (539, 500, '2024-12-15', '2025-12-15',
178663326 , 10);
INSERT INTO contract VALUES (540, 1000, '2024-12-15', '2025-12-15',
1023452569 , 10);

SELECT * FROM customer;
SELECT * FROM contract;

#*****trigger: after delete*****

DROP TRIGGER IF EXISTS contract_delete;

delimiter //
CREATE TRIGGER contract_delete
AFTER DELETE on contract
FOR EACH ROW
BEGIN
    UPDATE customer
    SET con_sum = con_sum - OLD.cost
    WHERE customer.AFM = OLD.AFM;
END;
//
delimiter ;

DELETE FROM contract WHERE contract_code = 528;
SELECT * FROM customer;
SELECT * FROM contract;

#*****trigger: after update*****

DROP TRIGGER IF EXISTS contract_update;
delimiter //

CREATE TRIGGER contract_update
AFTER UPDATE on contract
FOR EACH ROW
BEGIN
    UPDATE customer SET con_sum =
        (SELECT ifnull(sum(cost),0) FROM contract WHERE customer.AFM =
contract.AFM);
END;
//
delimiter ;

UPDATE contract SET cost = 623 WHERE contract_code = 517;
SELECT * FROM customer;
```

```

SELECT * FROM contract;

#*****info about TRIGGERS*****

DESCRIBE Information_schema.TRIGGERS;
SELECT TRIGGER_NAME, EVENT_MANIPULATION, TRIGGER_SCHEMA
FROM INFORMATION_SCHEMA.TRIGGERS
WHERE TRIGGER_SCHEMA = 'general_insurance'
ORDER BY TRIGGER_NAME;

#***** BHMA 7 *****
#function: date_check that counts years duration

DROP FUNCTION IF EXISTS date_check;
delimiter //
CREATE FUNCTION date_check()
RETURNS varchar(255)
DETERMINISTIC
BEGIN

DECLARE record_not_found int default 0;
DECLARE dStart Date;
DECLARE dEND Date;
DECLARE contractDuration varchar(255) default ' ';
DECLARE my_cursor cursor for SELECT start_date, END_date FROM contract;
DECLARE continue_handler for not found SET record_not_found = 1;

OPEN my_cursor;
  allclients: LOOP
    FETCH my_cursor INTO dStart, dEND;
    SELECT timestampdiff(YEAR,dStart, dEND) INTO @f;
    IF record_not_found THEN leave allclients;
    END IF;
    SET contractDuration = CONCAT(contractDuration, @f, ", ");
  END LOOP allclients;
CLOSE my_cursor;

return substr(contractDuration,1,255);
END
//

delimiter ;

SELECT date_check();

#***** BHMA 8 *****

#table that contains the results of each getPayment_proc call
DROP TABLE IF EXISTS monthlyPayment;
CREATE table monthlyPayment
(
  m_afm INT,
  m_date DATE,

```

```
m_no INT,  
m_cost decimal(7,2),  
PRIMARY KEY (m_afm, m_date)  
);  
  
#creating procedure getPayment_proc  
DROP PROCEDURE IF EXISTS getPayment_proc;  
delimiter !  
CREATE procedure getPayment_proc(  
    IN p_afm INT,  
    IN p_date DATE,  
    OUT p_no INT,  
    OUT p_cost decimal(7,2)  
)  
  
BEGIN  
  
    #number of active contracts  
    SET p_no = (SELECT count(*) FROM insurance INNER JOIN contract  
    ON insurance.insurance_code = contract.insurance_code  
    WHERE AFM = p_afm  
    AND p_date BETWEEN start_date AND END_date  
    );  
  
    #total cost of certain month  
    SET p_cost =(SELECT sum(annual_cost/12) FROM insurance INNER JOIN  
contract  
    ON insurance.insurance_code = contract.insurance_code  
    WHERE AFM = p_afm  
    AND p_date BETWEEN start_date AND END_date  
    );  
  
    INSERT INTO monthlyPayment VALUES (p_afm, p_date, p_no, p_cost);  
END !  
  
delimiter ;  
SELECT * FROM monthlyPayment;  
CALL getPayment_proc (1592227645,'2025-09-27', @out_no, @out_cost);  
SELECT * FROM monthlyPayment;  
  
#duplicate entry  
CALL getPayment_proc (1592227645,'2025-09-27', @out_no, @out_cost);  
  
CALL getPayment_proc (1592227645,'2023-09-27', @out_no, @out_cost);  
SELECT * FROM monthlyPayment;  
  
CALL getPayment_proc(1445547451, '2013-12-27',@out_no, @out_cost);  
SELECT * FROM monthlyPayment;
```