

PHP-ում MySQL-ին միացում և հարցումների կատարում

ՄԱՍ 2

Աշխատանք MySQL Տվյալների բազայի հետ

Աղյուսակների ստեղծումը և հեռացումը

Աղյուսակ ստեղծելու համար մենք պետք է օգտագործենք «CREATE TABLE» SQL արտահայտությունը:

```
<?php
require_once 'connection.php';

$link = mysqli_connect($host, $user, $password, $database)
    or die("Error " . mysqli_error($link));

$query = "CREATE Table tovars
(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(200) NOT NULL,
    company VARCHAR(200) NOT NULL
)";
$result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));
if($result)
{
    echo "Աղյուսակի ստեղծումը հաջող անցավ";
}

mysqli_close($link);
?>
```

Սցենարի գործարկումից հետո տվյալների բազայում կստեղծվի tovars աղյուսակը, որն իր սահմանմամբ նման կլինի այնպիսիներին, որոնք նախկինում ստեղծվել են phpMyAdmin-ի միջոցով: Մենք կարող ենք բացել phpMyAdmin միջերեսը և տեսնել այն:

Աղյուսակը հեռացնելու համար պետք է կատարել հետևյալ SQL հայտարարությունը՝ «DROP TABLE»

```

<?php
require_once 'connection.php';

$link = mysqli_connect($host, $user, $password, $database)
    or die("Error " . mysqli_error($link));

$query = "DROP TABLE tovars";
$result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));
if($result)
{
    echo " Աղյուսակի հեռացումը հաջող անցավ";
}

mysqli_close($link);
?>

```

Տվյալների ավելացում

Նախորդ թեմայում մենք տվյալների բազայում tovars աղյուսակում երեք սյուն ստեղծեցինք՝ id, name, company: Հիմա դրան ավելացրեք տվյալները:

Տվյալներ ավելացնելու համար օգտագործենք «INSERT» արտահայտությունը:

```
$query = "INSERT INTO tovars VALUES(NULL, 'Samsung Galaxy III', 'Samsung')";
```

«INSERT» արտահայտությունը մեկ տողով գրված տվյալները մտցնում է աղյուսակի մեջ: INTO բանալի բառից հետո նշվում է աղյուսակի անունը, իսկ VALUES-ից հետո փակագծերում նշվում է բոլոր սյունակների համար նախատեսված արժեքների մի շարք: Քանի որ մենք աղյուսակում երեք սյուն ունենք, մենք նշում ենք երեք արժեք:

Քանի որ նախորդ թեմայում, աղյուսակը ստեղծելիս մենք նշեցինք սյունակների հետևյալ կարգը՝ id, name, company, այս դեպքում id սյուն է փոխանցվում NULL, անվան համար՝ «Samsung Galaxy III», իսկ ընկերության համար՝ «Samsung»:

Քանի որ id սյունը սահմանվում է որպես AUTO_INCREMENT, մենք պետք չէ դրա համար հատուկ թվային արժեք նշենք, և կարող ենք փոխանցել NULL արժեք, իսկ MySQL-ը հաջորդ մատչելի արժեքը կհանձնի սյունակին:

Այժմ դիտարկենք օրինակ, որտեղ կատարվում է տվյալների ավելացնելը: Ստեղծենք create.php ֆայլ հետևյալ բովանդակությամբ՝

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

</head>

<body>

<?php
require_once 'connection.php';

if(isset($_POST['name']) && isset($_POST['company']))){

    // միացում սերվերին
    $link = mysqli_connect($host, $user, $password, $database)
        or die("Error " . mysqli_error($link));

    // mysql-ի համար սիմվոլների էկրանավորում
    $name = htmlentities(mysqli_real_escape_string($link, $_POST['name']));
    $company = htmlentities(mysqli_real_escape_string($link, $_POST['company']));

    // հարցման տողի ստեղծում
    $query = "INSERT INTO tovars VALUES(NULL, '$name', '$company')";

    // հարցումի կատարում
    $result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));
    if($result)
    {
        echo "<span style='color:blue;'>Տվյալներն ավելացված են</span>";
    }

    // փակել միացումը
    mysqli_close($link);
```

```

}
?>
<h2>Ավելացնել նոր մոդել</h2>
<form method="POST">
<p>Մուտքագրել մոդելը:<br>
<input type="text" name="name" /></p>
<p>Արտադրողը: <br>
<input type="text" name="company" /></p>
<input type="submit" value="Ավելացնել">
</form>
</body>
</html>

```

Այստեղ տվյալների բազաների հետ փոխգործակցության համար կողք գուգորդվում է ձևերի /ֆորմաներ/ ֆունկցիոնալության հետ: Ձևը օգտագործելով՝ մուտքագրում ենք տվյալներ՝ բազայում ավելացնելու համար:

Անվտանգությունը և MySQL-ը

Այստեղ մենք օգտագործեցինք `mysqli_real_escape_string()` ֆունկցիան: Այն օգտագործվում է պետք չեկող նիշերից խուսափելու համար, որն այնուհետև օգտագործվում է SQL հարցման մեջ:

Այսպիսով, մենք օգտագործում ենք նիշերի էկրանավորումը երկու անգամ՝ նախ `sql` արտահայտության համար, օգտագործելով `mysqli_real_escape_string()` ֆունկցիան, այնուհետև `html`-ի համար՝ օգտագործելով `htmlentities()` ֆունկցիան:

Տվյալների ստացումը

Տվյալներ ստանալու համար օգտագործվում է «SELECT» SQL արտահայտությունը: Օրինակ, ստանանք `tovars` աղյուսակից բոլոր տվյալները՝

```
$query="SELECT * FROM tovars";
```

տեսքով

Հիմա եկեք մի օրինակ դիտարկենք: Ստեղծեք index.php էջը, որը tovars աղյուսակից կստանա բոլոր տվյալները`

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

</head>

<body>

<?php
require_once 'connection.php'; // միացնել սկրիպտը

$link = mysqli_connect($host, $user, $password, $database)
    or die("Error " . mysqli_error($link));

$query = "SELECT * FROM tovars";

$result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
if($result)
{
    $rows = mysqli_num_rows($result); // ստացված տողերի քանակը

    echo "<table><tr><th>Id</th><th>Մոդելը</th><th>Արտադրողը</th></tr>";
    for ($i = 0 ; $i < $rows ; ++$i)
    {
        $row = mysqli_fetch_row($result);
        echo "<tr>";
        for ($j = 0 ; $j < 3 ; ++$j) echo "<td>$row[$j]</td>";
        echo "</tr>";
    }
    echo "</table>";
}
```

```
// մաքրենք արդյունքը
mysqli_free_result($result);
}
```

```
mysqli_close($link);
```

```
?>
```

```
</body>
```

```
</html>
```

Հարցման արդյունքները ցույց տալու համար այստեղ օգտագործվում է for ցիկլը: for ցիկլով մենք պետք է իմանանք, թե քանի ընդհանուր տող է վերադարձվում \$result փոփոխականում: Դրա համար օգտագործվում է mysqli_num_rows() ֆունկցիան:

Տողերի միջով անցնելու համար օգտագործենք հետևյալ հանգույցը`

```
for ($i = 0 ; $i < $rows ; ++$i)
{
    $row = mysqli_fetch_row($result);
    echo "<tr>";
    for ($j = 0 ; $j < 3 ; ++$j) echo "<td>$row[$j]</td>";
    echo "</tr>";
}
```

Մեկ տողը ստանալու համար օգտագործենք mysqli_fetch_row() ֆունկցիան: Այս ֆունկցիան կանչելուց հետո \$result հավաքածուի ցուցիչը անցնում է նոր տողի, այնպես որ յուրաքանչյուր նոր կանչի միջոցով մենք ստանում ենք նոր տող:

Ներքին ցիկլը իրականացվում է ընթացիկ շարքի բջիջների վրա`

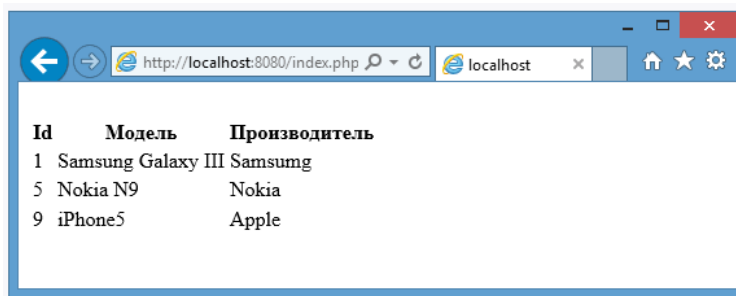
```
for ($j = 0 ; $j < 3 ; ++$j) echo "<td>$row[$j]</td>";
```

Քանի որ, երբ մենք տվյալները ստանում ենք աղյուսակի բոլոր երեք սյունակների համար, ապա \$j ցուցիչն անցնում է 0-ից մինչև 3-ը: Յուրաքանչյուր շարքը բջիջների զանգված է, այնպես որ, օգտագործելով \$ row [\$ j] արտահայտությունը, մենք կարող ենք մուտք գործել շարքի կոնկրետ բջիջ:

Կարելի է օգտագործել նաև while ցիկլը՝

```
while ($row = mysqli_fetch_row($result)) {  
    echo "<tr>";  
    for ($j = 0 ; $j < 3 ; ++$j) echo "<td>$row[$j]</td>";  
    echo "</tr>";  
}
```

Եվ քանի որ \$result փոփոխականը հարցումն իրականանալուց հետո կպահպանի տվյալները, վերջում մենք պետք է մաքրենք ավելորդ տվյալների հիշողությունը՝ օգտագործելով mysqli_free_result () ֆունկցիան:



Id	Модель	Производитель
1	Samsung Galaxy III	Samsung
5	Nokia N9	Nokia
9	iPhone5	Apple

Մենք կարող ենք նաև ստանալ ոչ բոլոր տվյալները, այլ, օրինակ, տվյալներ որոշակի սյուների համար: Օրինակ, ստանանք միայն մոդելների անունները՝

```
$query = "SELECT name FROM tovars";  
$result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));  
if($result)  
{  
    echo "<ul>";  
    while ($row = mysqli_fetch_row($result)) {  
        echo "<li>$row[0]</li>";  
    }  
    echo "</ul>";  
  
    mysqli_free_result($result);  
}
```

Այս դեպքում \$result հավաքածուի յուրաքանչյուր տող պարունակում է ընդամենը մեկ բջիջ, այսինքն՝ մենք չենք կարողանա մուտք ունենալ \$row [1], քանի որ այս դեպքում մենք տվյալներ ենք կորզում միայն մեկ սյունակի համար:

Տվյալների խմբագրում

Առկա տվյալների խմբագրումը մի փոքր ավելի դժվար կլինի տվյալների բազայից տվյալների ստացման և ավելացման խնդիրներից: Նախևառաջ, որովհետև այստեղ մենք պետք է համատեղենք, նախ և առաջ, խմբագրելի տվյալներ ձեռք բերելը, այնուհետև տվյալների բազան թարմացնենք նոր արժեքներով:

Թարմացնելու համար օգտագործեք «UPDATE» SQL արտահայտությունը՝

```
$query ="UPDATE tovars SET name='Samsung ACE II', company='Samsung' WHERE id='1'";
```

SET բառից հետո գրվում է սյունակների անվանումների ցուցակը և նրանց համար նոր արժեքները: Հարցման տողի վերջում ընտրիչը նշվում է WHERE կետի միջոցով: Այս դեպքում id = '1' –ից հետո դրված են բոլոր տողերը ` = 'Samsung ACE II' և ընկերությունը = 'Samsung':

Այժմ ստեղծենք edit.php նոր ֆայլ՝

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<?php
require_once 'connection.php';
// միացում սերվերին
$link = mysqli_connect($host, $user, $password, $database)
    or die("Error " . mysqli_error($link));

// POST-ի հարցում
if(isset($_POST['name']) && isset($_POST['company']) && isset($_POST['id'])){

    $id = htmlentities(mysqli_real_escape_string($link, $_POST['id']));
    $name = htmlentities(mysqli_real_escape_string($link, $_POST['name']));
    $company = htmlentities(mysqli_real_escape_string($link, $_POST['company']));
```



```

$query = "UPDATE tovars SET name='$name', company='$company' WHERE
id='$id'";
$result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));

if($result)
    echo "<span style='color:blue;'>Տվյալները թարմացված են</span>";
}

// GET-ի հարցում
if(isset($_GET['id']))
{
    $id = htmlentities(mysqli_real_escape_string($link, $_GET['id']));

    // հարցման տողի ստեղծում
    $query = "SELECT * FROM tovars WHERE id = '$id'";
    // հարցման կատարում
    $result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));
    // եթե հարցմանը բավարարում են մեկից ավելի տողեր
    if($result && mysqli_num_rows($result)>0)
    {
        $row = mysqli_fetch_row($result); // ստանում ենք առաջին տողը
        $name = $row[1];
        $company = $row[2];

        echo "<h2>Փոփոխել մոդելը</h2>
        <form method='POST'>
        <input type='hidden' name='id' value='$id' />
        <p>Մուտքագրող մոդելը:<br>
        <input type='text' name='name' value='$name' /></p>
        <p>Արտադրողը: <br>
        <input type='text' name='company' value='$company' /></p>
        <input type='submit' value='Պահպանել'>
        </form>";

        mysqli_free_result($result);
    }
}

// փակել միացումը
mysqli_close($link);
?>
</body>
</html>

```

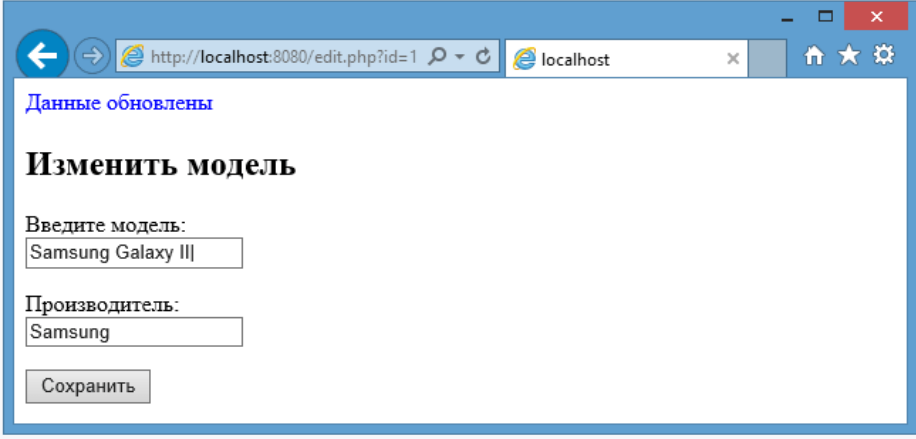
Կառուցվածքի տեսակետից, ամբողջ ծրագրային կոդը բաժանված է երկու մասի՝ POST հարցման մշակում և GET հարցման մշակում: POST հարցման մշակումը նման է տվյալների ավելացմանը, միայն այս դեպքում օգտագործվում է UPDATE արտահայտությունը՝

```
"UPDATE tovars SET name='$name', company='$company' WHERE id='$id'"
```

GET հարցումը մշակելիս մենք ստանում ենք id գրառումները և դրանից ստանում ենք ամբողջ գրառումը՝ օգտագործելով «SELECT * FROM tovars WHERE id = '\$id'" արտահայտությունը: Այժմ առաջին գրառումը /ձայնագրությունը/ խմբագրելու համար հարկավոր է մուտք գործել բրաուզերում գտնվող սցենարը՝ <http://localhost:8080/edit.php?Id=1> հասցեում:

Հայցից ստանալով id՝ PHP-ն բաց կթողնի այն մասը, որը վերաբերում է POST հարցման մշակմանը և կձեռնարկի տվյալների բազայից գրառումը id-ով հանելու և ձևը /ֆորմը/ ստեղծելու միջոցով: Այնուհետև ֆորմայում մենք կարող ենք մուտքագրել որոշ արժեքներ և ուղարկել սերվերին: Այժմ ներգրավված կլինի սցենարի այն մասը, որը մշակում է POST հարցումը:

Դիտարկիչի վրա ստանում ենք՝



Տվյալների հեռացումը

Տվյալները հեռացնելիս օգտագործվում է «DELETE» SQL արտահայտությունը՝

```
$query="DELETE FROM tovars WHERE id = '5'";
```

Այս դեպքում աղյուսակից հեռացվում է այն տողը, որն ունի id = 5 արժեքը:

Տվյալը հեռացնելիս, ինչպես խմբագրման ժամանակ, մենք պետք է սկրիպտին փոխանցենք հեռացվող ձայնագրության id-ն, որպեսզի կատարվի հարցումը:

Այժմ ստեղծեք Delete.php սցենար հետևյալ բովանդակությամբ՝

```
<?php
require_once 'connection.php';
if(isset($_GET['id']))
{
    $link = mysqli_connect($host, $user, $password, $database)
        or die("Error " . mysqli_error($link));
    $id = mysqli_real_escape_string($link, $_GET['id']);

    $query = "DELETE FROM tovars WHERE id = '$id'";

    $result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));
    mysqli_close($link);
}
?>
```

Դիմելով սցենարին հետևյալ հարցումով՝ <http://localhost:8080/Delete.php?id=1>, սցենարը կհեռացնի աղյուսակի առաջին տողը:

Հեռացման այս մեթոդը շատ պարզ է և հիմնական է աշխատում, բայց դա անվնաս չէ: Ենթադրենք, մենք ստացել ենք էլ-նամակ, որում տեղի միջոցով ներդրված է նկար՝

```

```

Արդյունքում, նամակը բացելիս, աղյուսակում 1-ին գրառումը կարող է ջնջվել: Խոցելիությունը վերաբերում է ոչ միայն տառերին, այլև կարող է առաջանալ նաև այլ վայրերում: Հետևաբար, մենք փոփոխելու ենք Delete.php սցենարը հետևյալ կերպ՝

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<?php
require_once 'connection.php'; // սկրիպտի միացումը

if(isset($_POST['id'])) {
```

```

$link = mysqli_connect($host, $user, $password, $database)
    or die("Error " . mysqli_error($link));
$id = mysqli_real_escape_string($link, $_POST['id']);

$query = "DELETE FROM tovars WHERE id = '$id'";
$result = mysqli_query($link, $query) or die("Error " . mysqli_error($link));

mysqli_close($link);
// ուղղորդում index.php սկրիպտ
header('Location: index.php');
}

if(isset($_GET['id']))
{
    $id = htmlentities($_GET['id']);
    echo "<h2>Հեռացնել մոդելը?</h2>
        <form method='POST'>
            <input type='hidden' name='id' value='$id' />
            <input type='submit' value='Հեռացնել'>
        </form>";
}
?>
</body>
</html>

```

Այժմ, սցենար մուտք գործելիս օգտագործողին կցուցադրվի մի ձև, որտեղ կա թաքնված դաշտ, որը պարունակում է հեռացված գրառման id-ն և կոճակը հեռացնելու համար:

Հարցումն ավարտվելուց հետո այն վերահղվում է index.php էջին՝ օգտագործելով header() ֆունկցիան: