

КАТЕДРА: КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ
ДИСЦИПЛИНА: ИНФОРМАЦИОННИ СИСТЕМИ

ЛАБОРАТОРНО УПРАЖНЕНИЕ № 5

ТЕМА: Свързване на PHP и MYSQL база данни

ЦЕЛ:

Целта на упражнението е студентите да получат практически знания и умения за достъп на PHP скрипт към MySQL бази данни. След упражнението студентите би следвало и да могат да вмъкват информация в базата данни чрез PHP.

!ВАЖНО: За да можете да тествате упражнението ви е необходим следния софтуер. Web server, Php интерпретатор, MYSQL база данни. Всеки един от тези софтуери ги има в пакетите WAMP или XAMP. Настоящото упражнение е тествано върху WAMP Version 3.2.0.

I. ТЕОРЕТИЧНА ЧАСТ

За да може скрипт на PHP да работи с база от данни, първо трябва да се създаде връзка между скрипта и системата за управление на базата (DBMS). Свързването на PHP с DBMS е действие от няколко стъпки. Всяка една от тях е разгледана поотделно. Това, което първо трябва да се направи в PHP скрипта, е да създаде връзка към DBMS, с която да взаимодейства. За да се постигне това, се използва функцията **mysqli_connect()**:

ресурс=mysqli_connect(сървър, потребителскоИме, парола);

Тя приема три параметъра. Първият е името на сървъра, вторият е потребителското име в DBMS с което се осъществява връзката, а третият е неговата парола. Функцията **mysqli_connect()** връща стойност от тип ресурс. Ако разработвате приложението на вашия личен компютър, а не през интернет, тази функция би изглеждала по същия начин: стойност на сървър „localhost“, потребителско име „root“ и паролата с която сте защитили системата за управление на бази от данни MySQL. Дори парола да не е зададена то пак трябва да се включи последния параметър, като се остави празен низ по следния начин ‘ ‘.

\$con=mysqli_connect('localhost','root','');

ЗАБЕЛЕЖКА: В упражнението за улеснение се конфигурира достъп без зададена парола на потребителя root на MySQL. В реална среда това представлява заплаха за сигурността и трябва да се използва достатъчно силна парола.

След като е осъществена връзка с желаната DBMS се преминава към следващата стъпка – избор на базата от данни която ще се използва. Това става посредством функцията **mysqli_select_db()**. В нея се посочва точната база от данни която да се достигне, от ресурса, дефинират в **mysqli_connect()**:

mysqli_select_db(ресурс, ИмеНаБазатаОтДанни);

Функцията приема два параметъра. Първият е ресурсът получен от функцията **mysqli_connect()**, а вторият е името на базата данни към която се осъществява достъп. Функцията **mysqli_select_db()** връща стойност true ако изборът на базата е бил успешен и false, ако не е.

Функцията `die()`, спира изпълнението на скрипта, ако заявената преди нея връзка с база от данни не може да бъде осъществена. Тя изглежда по следния начин:

`die` ("Съобщение за грешка");

Тази функция има само един параметър – съобщението което се показва, преди изпълнението на скрипта да бъде спряно. Стандартна практика е да се използва комбинация от функцията `die()` с функцията `mysqli_error()`, която връща текста на съобщението за грешка от предходната операция в MySQL. Функцията `mysqli_error()` изглежда по следния начин:

`mysqli_error(ресурс);`

Тя най често приема за параметър ресурса върнат от функцията `mysqli_connect()`. Алтернатива вместо `mysqli_error()`, често се използва е функцията `mysqli_connect_error()`, която работи по същия начин. Комбинацията между тази функция и функция `die()` изглежда така:

`die` ("Неуспешно свързване: ".`mysqli_connect_error()`);

Следващата стъпка от процеса на свързване на PHP с MySQL база от данни е да се изпращат оператори на SQL заявки към базата, за да се вмъква и извлича информация от нея. В настоящото упражнение се разглежда как се вмъква информация в DBMS.

Функцията чрез която се изпращат SQL заявки към базата се нарича `mysqli_query()`. Тя изисква да и се подадат два параметъра. Първият е ресурса сочещ към базата от данни, който е получен от функцията `mysqli_connect()`, а вторият е низ със SQL заявка. При изпълнение на функцията се получава ресурс, който води към обработените данни.

връзкаКъмДанни=`mysqli_query`(ресурс, заявка);

В настоящото упражнение ще се запознаем как можем да създаваме нова база от данни, създаваме нова таблица в съществува база от данни, както и вмъкване на записи в съществуваща таблица. За целта ще трябва да си припомним следните SQL заявки ***CREATE DATABASE***, ***CREATE TABLE*** и ***INSERT INTO***.

Операторът ***CREATE DATABASE*** се използва за създаване на нова база данни. След него се посочва името на базата от данни. Има следния вид:

CREATE DATABASE ИмеНаБазаОтДанни;

Операторът ***CREATE TABLE*** се използва за създаване на нова таблица в съществуваща база от данни. След него се посочва името на таблицата след което в скоби със запетайки се отделят имената на полетата за съответната таблица. За всяко поле се посочва неговия тип. Последния параметър във скобите е за посочване на ключови индексни полета. Оператора има следния вид:

CREATE TABLE ИмеНаТаблица

(поле1 ТИП, поле2 ТИП, поле3 ТИП, ключов индекс (поле))

Операторът ***INSERT INTO*** се използва за вмъкване на записи във вече съществуваща таблица. Той има следния вид:

INSERT INTO таблица (поле1, поле2, ...) VALUES('стойност1', 'стойност2', ...)

В него след ключовите думи ***INSERT INTO*** се поставя името на таблицата, в която ще се добавя записа. След това в скоби се посочва името на всяко от полетата в таблицата който ще се попълват, като се разделят със запетая. Следва ключовата дума ***VALUES***, след която, отново в скоби и разделени със запетайки, се поставят стойностите, които да се вмъкнат в съответните полета.

Операторът може да се опрости след като се пропуснат имената на полетата по следния начин:

INSERT INTO *таблица* **VALUES**(‘стойност1’, ‘стойност2’,...)

Така операторът добавя изброените стойности в полетата на базата според реда, в който са подадени.

II. ПРАКТИЧЕСКА ЧАСТ

В практическата част е показан пример чрез който са обяснени нагледно отделните етапи за осъществяване достъп от PHP скрипт към MySQL база от данни. За целта е решена следната задача:

ЗАДАЧА: Да се осъществи достъп от PHP скрипт към MySQL база от данни. Да се създаде БД “University”. Към нея да се създаде таблица “Students” показана на табл.1. Да се създаде HTML форма с 4 полета в които се посочва информация и чрез натискане на бутон Submit да се вмъкне информацията в таблицата.

Табл.1. Students

<i>fnum</i>	<i>name</i>	<i>age</i>	<i>grade</i>
2344	Dian	21	3.56
2234567	Ivan	21	4.59

1. Свързване със сървъра за MYSQL База от Данни

КОД:

```
<?php
$con = mysqli_connect('localhost', 'root', '', 'university') or die(mysqli_connect_error());
?>
```

ПОЯСНЕНИЕ

За да осъществим достъп до MySQL база от данни се използва функцията `mysqli_connect()`. Предполага се че имате инсталиран WAMP или XAMP пакет и осъществявате достъп до сървъра който се намира на локалната машина. Затова първия параметър на функцията е „localhost“. В примера се осъществява достъп с потребител root който е без парола.

ЗАБЕЛЕЖКА В реална среда това представлява заплаха за сигурността и трябва да се използва достатъчно силна парола.

Горния код работи по следния начин: Осъществява се достъп до MySQL сървър инсталиран на локалната машина като се използва потребител root. Ако поради някаква причина не бъде осъществена връзката се връща съобщението „Неуспешна връзка“ и се показва точната причина за грешката.

2. Създаване на базата от данни „University“

КОД:

```
<?php
//1. Свързване със сървъра за MYSQL БД и проверка на връзката
```

```

$con=mysqli_connect('localhost','root','');
if (!$con) die(mysqli_connect_error());

//2. Създаване на заявка към сървъра за създаване на база от
//данни с име university
$str='CREATE DATABASE University';

//3. Изпращане на заявката към сървъра за създаване на база от
//данни с име University

mysqli_query($str,$con) or die('Error creating database: '.
mysqli_error($con));

?>

```

ПОЯСНЕНИЕ

В първата част от кода се осъществява достъп до същата база данни както в т.1. и кода работи по същия начин. Във втората част от кода се създава променлива \$str от тип низ като в нея задава SQL заявка с която се създава база данни „University“. В третата част от кода чрез функцията mysqli_query се изпраща посочената заявка във връзката която сме осъществили чрез функцията mysqli_connect. Ако поради някаква причина заявката не се изпълни функция die ще върне съобщение с грешката.

ЗАБЕЛЕЖКА. Ако запишете този код в php файл и го изпълните то в браузъра няма да се покаже нищо. Това ще означава че най вероятно кода се е изпълнил успешно и е създадена базата данни „University“. След изпълнението му бихте могли заредите <http://localhost/phpmyadmin> и да влезнете чрез интерфейса на phpMyAdmin в сървъра на базата данни където можете да видите че вашата база данни е създадена. Това е възможно само ако имате инсталиран phpMyAdmin. Ако сте инсталирали WAMP или XAMP най вероятно имате и пакета phpMyAdmin.

3. Проверка за съществуването на вече създадената база данни „university“.

За да се направи проверка дали базата данни наистина е създадена то може да се добавят echo команди които да ни показват съответното съобщение.

КОД

```

<?php
$con=mysqli_connect('localhost','root','') or die(mysqli_connect_error());
echo "Connection to the server was successful! <br/>";
mysqli_select_db($con, "university") or die(mysqli_error($con));
echo "Database was selected! <br/>";

?>

```

ПОЯСНЕНИЕ

Този код осъществява достъп отново до посочения сървър и се опитва да избере база данни със име „university“. Ако не намери такава база данни ще изкара съответното съобщение за грешка. Ако няма грешки при изпълнението на кода ще изкара съобщение за успех.

4. Създаване на таблица „Students“ в базата от данни „University“.

КОД

```
<?php
// 1 част. Осъществяване достъп до базата от данни "University"
$con=mysqli_connect('localhost', 'root', '', 'University') or die(mysqli_connect_error());
echo 'Connected successfully! ';
// 2 част. Създаване на таблица Студенти
$sql="CREATE TABLE Students( fnom INT(10) NOT NULL, name VARCHAR(32),
age INT(6), grade DOUBLE (5,2), PRIMARY KEY (fnom) ) DEFAULT CHARSET=utf8";
mysqli_query( $con, $sql ) or die(mysqli_error($con));
echo "Table students created successfully";
// 3 част. Затваряне на връзката към базата от данни
mysqli_close($con);
?>
```

ПОЯСНЕНИЕ

В първата част от кода отново се осъществява достъп до вече съществуваща база от данни "University". Обърнете внимание къде е посочено името на базата от данни. Както се вижда това може да става във функцията `mysqli_connect()` като 4ти параметър. Тази част от кода работи на същия принцип както и в предходните точки.

Във втората част от кода се създава променлива `$str` от тип низ в която се задава SQL команда за създаване на нова таблица с посочените полета. Тази заявка се изпраща чрез функцията `mysqli_query` и ако не се върне някакво съобщение за грешка то ще се отпечата съобщението „Table students created successfully“.

В третата част от кода е използвана функция `mysqli_close()` чрез която се затваря отворената връзка от `mysqli_connect`.

ЗАБЕЛЕЖКА: Ако една връзка вече е създадена и не е затворена то не е необходимо тя да бъде създавана отново с `mysqli_connect()`. За по голяма сигурност обаче прието след като се приключи работа със дадена връзка тя да бъде затворена с функцията `mysqli_close()`.

5. Вмъкване на един запис в таблицата Студенти.

Кода който трябва да се използва е същия като досега само че се изпраща друга sql заявка със функцията `mysqli_query()`. Разликата е в следния фрагмент от кода:

КОД

```
$sql="INSERT INTO Students (fnom,name,age,grade) VALUES (2234567,
'Иван', 21, 4.59)";
mysqli_query( $con, $sql ) or die(mysqli_error($con));
```

6. Добавяне на данни чрез използване на HTML форма:

КОД

forma.php

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    <form action="test.php" method="post">
      Въведи факултетен номер: <input type="text" name="fnom" />
      Име: <input type="text" name="name" />
      Възраст: <input type="text" name="age" />
      Успех: <input type="text" name="uspeh" />
      <input type="submit" />
    </form>
  </body>
</html>
```

test.php

```
<?php
$con = mysqli_connect('localhost', 'root', '', 'university') or die(mysql_error());
$sql="INSERT INTO Students (fnom, name, age, grade)
VALUES ('$_POST ['fnom']", "$_POST ['name']", "$_POST ['age']",
"$_POST ['uspeh'])";
mysqli_query($sql, $con) or die(mysql_error());
echo "1 record added"; mysqli_close($con);
?>
```

ПОЯСНЕНИЕ

Задачата е решена като се създават два файла Forma.php и test.php. Във първия файл са html таговете за създаване на формата показана на фиг.1., а във файла test.php се намира php скрипта който осъществява достъп до съответната база данни и записва един запис в таблица студенти чрез натискането на бутона от създадената форма. Необходимата информация която трябва да се запише в таблицата се въвежда в полетата от формата затова тя се достъпва и чрез глобална променлива вземаща информацията от заявката \$_POST[' ']. В апострофите се посочва името на полето от което четем информацията.

Въведи факултетен номер: Име: Град: Възраст:

Фиг.1. Визуализация на формата

III. Задача за самостоятелна работа.

1. Осъществете връзка с php скрипт към MySQL базата от данни.
2. Създайте база данни с наименование "Firma".
3. В нея създайте таблица „Products“ със следните полета в таблицата „ИмеНаПродукта“, „Описание“, „Цена“. Типа на полетата го задайте спрямо тяхното предназначение.
4. Създайте HTML форма с три полета и бутон. Създайте PHP скрипт който след натискането на бутона да прочита въведената информация в полетата и да я добавя като нов запис в таблица Products.
5. Затворете осъществената връзка към MySQL