

Лабораторно упражнение № 10

PHP и MySQL. Функции на PHP за операции с бази от данни

PHP може да работи с различни бази от данни, включително Oracle, но основно се използва с MySQL бази от данни. MySQL е Система за Управление на релационни БД, която предоставя достъп до данните посредством езика SQL. Синтаксисът на заявките към MySQL е почти идентичен със стандартния SQL синтаксис. MySQL е СУБД, която осигурява възможност за едновременен достъп до База от Данни на много потребители в мрежова среда. Връзката между PHP и MySQL се осъществява чрез драйверите:

- ✓ php_mysql.dll
- ✓ php_mysqli.dll
- ✓ php_pdo_mysql.dll

Тези драйвери на практика управляват достъпа до MySQL Бази от данни. В дистрибуцията на PHP за Windows, 3 – те драйвера php_mysql.dll, php_mysqli.dll и php_pdo_mysql.dll (библиотеки на PHP за операции с MySQL/MariaDB БД) са в папка ext на php, това са:

- ✓ ...\\ext\\php_mysql.dll
- ✓ ...\\ext\\php_mysqli.dll
- ✓ ...\\ext\\php_pdo_mysql.dll

Драйверите **mysqli** (т.е **mysql improved**) и **pdo (PHP Data Objects)** са изцяло разработени за 5-та версия на PHP - за да се предостави на програмистите достъп до новите възможности след MySQL 4.1.

1. Свързване с MySQL сървър - mysql_connect()

За да създадете и използвате каквато и да е База от данни в PHP код, както и в код, на друг език, първото нещо е да се свържете със сървъра за MySQL. Свързването към една MySQL база от данни изисква извикване на функция mysql_connect(), подавайки следната информация:

- Име на хост (hostname);
- Потребителско име;
- Парола за достъп (password), ако е била зададена;

Функцията mysql_connect() установява връзката с MySQL сървъра и връща като резултат стойност от тип ресурс (resource), съдържаща идентификатор на връзката при успех и FALSE при неуспех.

!!! Всички примери в упражнението са реализирани в два варианта - за драйвер php_mysql.dll и php_mysqli.dll

Пример 1. Осъществяване на връзка със сървър.

за драйвер php_mysql.dll

connect.php

```
<?php
    $host= 'localhost'; //Машина, на която работи MySQL сървърът
    $dbUser= 'root'; // Потребителско име за MySQL
    $dbPass= 'root'; // Парола за MySQL
    //връзка със сървъра
    if(!$dbConn=mysql_connect($host, $dbUser, $dbPass)) {
        // или mysql_connect('localhost', 'root', 'root')
        die('Не може да се осъществи връзка със сървъра:'.mysql_error());
    }
    // тестване на връзката
    echo 'Връзката е успешна!';
?>
```

за драйвер php_mysqli.dll

connect.php

```
<?php
    $host= 'localhost'; //Машина, на която работи MySQL сървърът
    $dbUser= 'root'; // Потребителско име за MySQL
    $dbPass= 'root'; // Парола за MySQL
    //връзка със сървъра
    if(!$dbConn=mysqli_connect($host, $dbUser, $dbPass)) {
        die('Не може да се осъществи връзка със сървъра.');
```

```
// тестване на връзката
echo 'Връзката е успешна!';
?>
```

Тъй като MySQL е независима СУБД, тя може да работи на същия или друг компютър, на който е интерпретаторът на PHP. В примерния файл MySQL работи на същия компютър, на който и PHP, затова за име на host машината е зададено localhost. За връзка към MySQL сървър, работещ на отдалечен host може да се използва DNS името на сървъра или неговия IP адрес, например \$host= 'www.myserver.com' или \$host= '212.91.184.2'. По подразбиране връзката с MySQL се осъществява на порт 3306. Ако е необходимо да се укаже друг порт, той се указва след името на сървъра, разделен с двоеточие. Пример: \$host= '212.91.184.2:4321'. Условният оператор проверява дали връзката със сървъра е установена. При неуспех функцията die() извежда съобщение и прекъсва изпълнението на скрипта.

2. Създаване на тестова база от данни

Пример2. Осъществяване на връзка със сървър и създаване на база данни.

за драйвер php_mysql.dll

connect_createDB.php

```
<?php
    $host= 'localhost'; //Машина, на която работи MySQL сървърът
    $dbUser= 'root'; // Потребителско име за MySQL
    $dbPass= 'root'; // Парола за MySQL
    //връзка със сървъра
    if(!$dbConn=mysql_connect($host, $dbUser, $dbPass)) {
        // или mysql_connect('localhost', 'root', 'root')
        die('Не може да се осъществи връзка със сървъра:'.mysql_error());
    }
    $sql = 'CREATE Database student61';
    if ($queryResource=mysql_query($sql, $dbConn))
    //или mysql_query('CREATE DATABASE student61', $dbConn)
    {
        echo "Базата данни е създадена. <br>";
    }
    else
    {
        echo "Грешка при създаване на базата данни: " . mysql_error();
    }
?>
```

за драйвер php_mysqli.dll

connect_createDB.php

```
<?php
    $host= 'localhost'; //Машина, на която работи MySQL сървърът
    $dbUser= 'root'; // Потребителско име за MySQL
    $dbPass= ''; // Парола за MySQL
    //връзка със сървъра
    if(!$dbConn=mysqli_connect($host, $dbUser, $dbPass)) {
        die('Не може да се осъществи връзка със сървъра. ');
    }
    $sql = 'CREATE Database student61';
    if ($queryResource=mysqli_query($dbConn,$sql))
    //или mysqli_query($dbConn,'CREATE DATABASE student61' )
    {
        echo "Базата данни е създадена. <br>";
    }
    else
    {
        echo "Грешка при създаване на базата данни: ";
    }
?>
```

mysql_query("заявка", [“идентификатор на връзката”]) – изпраща заявка към активната БД на MySQL сървъра, като връзката се идентифицира с помощта на \$dbConn(идентификатора на връзката). Ако \$dbConn не е зададено, се използва последното направено съединение. Връща тип ресурс – т.е. указател към резултата, иначе False. След изпълнение на заявката MySQL не връща директно резултата на скрипта, а го съхранява в паметта на сървъра в очакване на следващи команди. PHP има достъп до

резултата чрез ресурсен идентификатор `$queryResource`, който се връща като резултат от функцията `mysql_query()`.

mysql_query() (“идентификатор на връзката”, “заявка“)

mysql_error() – показва текстово съобщение за грешка върнато директно от MySQL.

3. Селектиране на базата от данни - `mysql_select_db()`

Тъй като един MySQL сървър може да управлява достъпа до няколко бази от данни, след установяване на връзката със сървъра трябва да се укаже с коя точно база от данни ще се работи.

Функцията `mysql_select_db()` установява връзка с базата от данни, дадена като първи параметър на функцията. Втория параметър на функцията е идентификаторът на установената връзка с MySQL сървъра. Този параметър не е задължителен, тъй като вече установената връзка се подразбира, но се използва при сложни скриптове, които извършват връзка с много сървъри и различни бази от данни.

Пример 3. Осъществяване на връзка със сървър и селектиране на база данни.

за драйвер `php_mysql.dll`

`connect_selectDB.php`

```
<?php
    $host= 'localhost'; //Машина, на която работи MySQL сървърът
    $dbUser= 'root'; // Потребителско име за MySQL
    $dbPass= 'root'; // Парола за MySQL
    $dbName= 'student61'; // Име на базата от данни
    //връзка със сървъра
    if(!$dbConn=mysql_connect($host, $dbUser, $dbPass)) {
        // или mysql_connect('localhost', 'root', 'root')
        die('Не може да се осъществи връзка със сървъра:'.mysql_error());
    }
    echo "Връзката е успешна. <br>";
    if (!mysql_select_db($dbName, $dbConn))
    {
        die('Не може да се селектира базата от данни:'. mysql_error());
    }
    echo "Базата данни е селектирана. <br>";
?>
```

за драйвер `php_mysqli.dll`

`connect_selectDB.php`

```
<?php
    $host= 'localhost'; //Машина, на която работи MySQL сървърът
    $dbUser= 'root'; // Потребителско име за MySQL
    $dbPass= 'root'; // Парола за MySQL
    $dbName= 'student61'; // Име на базата от данни
    //връзка със сървъра
    if(!$dbConn=mysqli_connect($host, $dbUser, $dbPass)) {
        // или mysqli_connect('localhost', 'root', 'root')
        die('Не може да се осъществи връзка със сървъра.');
```

```
    }
    echo "Връзката е успешна. <br>";
    if (!mysqli_select_db($dbConn, $dbName))
    {
        die('Не може да се селектира базата от данни.');
```

```
    }
    echo "Базата данни е селектирана. <br>";
?>
```

4. Затваряне на връзката - `mysql_close()`

От PHP 4 не е необходимо затваряне на връзката със сървъра за MySQL бази от данни, тъй като това се контролира автоматично. Установената връзка може да бъде затворена чрез използване на функцията `mysql_close()`,

```
.....
mysql_close($dbConn);
//или mysql_close();
.....
```

`$dbConn` - идентификатор на връзката, върнат от `mysql_connect()`.

5. Създаване на файл `config.php`, който осъществява връзка със сървъра и селектира базата от данни.

за драйвер php_mysql.dll**config.php**

```
<?php
    $host= 'localhost';
    $dbUser= 'root';
    $dbPass= 'root';
    $dbName= 'student61';
    if (!$dbConn=mysql_connect($host, $dbUser, $dbPass))
    {
        die('Не може да се осъществи връзка със сървъра:'.mysql_error());
    }

    if (!mysql_select_db($dbName, $dbConn))
    {
        die('Не може да се селектира базата от данни:'.mysql_error());
    }
    mysql_query("SET NAMES 'UTF8'");
?>
```

за драйвер php_mysql.dll**config.php**

```
<?php
    $host= 'localhost';
    $dbUser= 'root';
    $dbPass= '';
    $dbName= 'student61';
    if (!$dbConn=mysqli_connect($host, $dbUser, $dbPass))
    {
        die('Не може да се осъществи връзка със сървъра.');
```

6. Създаване на таблица

За създаване на таблица е необходимо да създадем SQL заявката и да я изпълним, като я подадем за параметър на функцията `mysql_query()`.

Синтаксис на командата за създаване на таблица в MYSQL

CREATE [TEMPORARY] TABLE [IF NOT EXISTS]

[Име на БД].име на таблица[(дефиниция на колона,...)] [машина за съхранение] [select израз]

дефиниция на колона - изброяват се колоните, които трябва да съдържа таблицата.

Формалното описание на „дефиниция на колона“ :

Име на стълб тип [NOT NULL | NULL] [DEFAULT стойност по подразбиране]

[AUTO_INCREMENT] [UNIQUE [KEY]]/[PRIMARY KEY] [reference_definition]

Всеки стълб може да няма стойност (NULL по подразбиране), да бъде NOT NULL, да има стойност по подразбиране, да бъде ключ или автоматично да нараства (AUTO_INCREMENT). Полето, което е първичен ключ на таблицата трябва да има уникална стойност!!! Не се допуска NULL.

Машината за съхранение на данните (ENGINE) зависи от типа на таблицата, например ENGINE = InnoDB е за таблици тип InnoDB. InnoDB – поддържа таблици със сигурни транзакции, предназначен за големи и често използвани бази. Има механизъм за заключване на редове, за да се предотврати промяната на ред, едновременно ползван от няколко потребители. Това е добрият избор ако в приложението ще има много INSERT/UPDATE SQL заявки, ако има нужда от външни ключове (т.е InnoDB поддържа и външни ключове за разлика от MyISAM). Заема повече памет от MyISAM.

Задаване на кодиране на ниво БД, таблица и колона (CHARACTER SET).

При създаване на базата от данни, на таблица на БД и колона на таблица може да се посочи и в какъв код да бъдат съхранени данните (latin1, Windows-1251, UTF- 8 или друг). Това става чрез клаузата CHARACTER SET, която се поставя на последно място в оператор CREATE.

CHARACTER SET utf8; или DEFAULT CHARSET=utf8;

Пример 4. Създаване на таблица “students” в тестова база от данни “student61”, чрез PHP скрипт.

за драйвер `php_mysql.dll`

`create_table.php`

```
<?php
    include "config.php";
    $sql ="CREATE TABLE students(
    id INT(10) NOT NULL,
    name VARCHAR(32) DEFAULT NULL,
    city VARCHAR(30) DEFAULT NULL,
    age SMALLINT(6) DEFAULT NULL,
    PRIMARY KEY (id)
    ) ENGINE=INNODB DEFAULT CHARSET=utf8";
    $result = mysql_query($sql);
    if(!$result)
        die('Грешка при създаване на таблицата: ' . mysql_error());
    echo "Таблицата е създадена!";
?>
```

за драйвер `php_mysql.dll`

`create_table.php`

```
<?php
    include "config.php";
    $sql ="CREATE TABLE students (
    id INT(10) NOT NULL,
    name VARCHAR(32) DEFAULT NULL,
    city VARCHAR(30) DEFAULT NULL,
    age SMALLINT(6) DEFAULT NULL,
    PRIMARY KEY (id)
    ) ENGINE=INNODB DEFAULT CHARSET=utf8";
    $result = mysqli_query($dbConn,$sql);
    if(!$result)
        die('Грешка при създаване на таблицата.');
```

echo "Таблицата е създадена!";

?>

7. Въвеждане на данни в MySQL таблица

Синтактични правила:

- ✓ SQL заявката трябва да бъде оградена с кавички в PHP кода;
- ✓ Стойностите от тип стринг в SQL заявката трябва да бъдат оградени с кавички;
- ✓ Цифровите стойности не трябва да се ограждат с кавички;
- ✓ Думата NULL не трябва да се огражда с кавички.
- ✓ Синтаксисът на `mysql` изисква променливите да са оградени с апострофи.

За въвеждане на данни в таблица на MySQL се използва SQL командата `INSERT INTO`.

`INSERT INTO tablename (field1, field2,...,fieldN) VALUES (value1, value2,...,valueN)`

`tablename` – Указва се таблицата, в която ще се въвеждат данните.

`field1,...,fieldN` – Задават се полетата, в които ще се добавят данни. Ако стойността на полето се генерира автоматично, може да не се укаже в списъка.

`value1, value2,...,valueN` – Задават се стойностите, които трябва да бъдат добавени в указаните полета.

Ако колоната е със статус `AUTO_INCREMENT` или е със статус `TIMESTAMP` не е необходимо да я упоменаваме изрично в SQL заявката. MySQL автоматично ще добави стойността.

Пример 5. Добавяне на данни в таблица `students` (в тестова база от данни “student61”) чрез PHP скрипт (добавете 5 записа).

Въвеждане на един запис в таблица `students`:

```
INSERT INTO students (id, name, city, age) VALUES (2234567, 'Иван', 'Варна', 25);
```

Съхраняваме този SQL оператор в една променлива и я подаваме като параметър на функция `mysql_query()`.

```
$sql=" INSERT INTO students (id, name, city, age) VALUES (2234567, 'Иван', 'Варна', 25)";
$result = mysql_query($sql);
```

за драйвер `php_mysql.dll`

`insert.php`

```
<?php
    include "config.php";
```

```

$sql=" INSERT INTO students (id, name, city, age) VALUES (22345767, 'Иван', 'Варна',
25)";
$result = mysql_query($sql);
if (!$result)
{
    die('Error: ' . mysql_error());
}
echo "Добавихте един запис.";
?>

```

за драйвер `php_mysql.dll`

insert.php

```

<?php
include "config.php";
$sql=" INSERT INTO students (id, name, city, age) VALUES (22345767, 'Иван', 'Варна',
25)";
$result = mysqli_query($dbConn,$sql);
if (!$result)
{
    die('Грешка!!!');
}
echo "Добавихте един запис.";
?>

```

Пример 6. Добавяне на данни в таблица `students` чрез използване на HTML форма.

localhost / 127.0.0.1 / student6 X localhost/PhpProject12/test.php X

localhost/PhpProject12/test.php

факултетен номер:	61444444
Име:	Иван Иванов
Град:	Варна
Възраст:	21

Въведи

index.php

```

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title></title>
</head>
<body>
    <form action="vhod.php" method="post">
    <pre>
        Въведи факултетен номер: <input type="text" name="id"/>
        Име: <input type="text" name="name" />
        Град: <input type="text" name="city" />
        Възраст: <input type="text" name="age"/>
        <input type="submit" name="submit" value="Въведи"/>
    </pre>
    </form>
</body>
</html>

```

за драйвер `php_mysql.dll`

vhod.php

```

<?php
include "config.php";
if (isset($_POST["submit"]))
{
    $id = $_POST['id'];
    $name = $_POST['name'];
}

```

```

$city= $_POST['city'];
$age = $_POST['age'];
if ( !empty($id)&&!empty($name)&&!empty($city)&&!empty($age) )
{
    $sql="INSERT INTO students (id, name, city, age) VALUES ('$id', '$name',
'$city', '$age)";
    $result = mysql_query($sql);
    if (!$result)
    {
        die('Error: ' . mysql_error());
    }
    echo "Добавихте един запис.";
}
else
    echo "Не сте въвели всички данни!!!";
}
?>

```

за драйвер **php_mysqli.dll**

vhod.php

```

<?php
include "config.php";
if (isset($_POST["submit"]))
{
    $id = $_POST['id'];
    $name = $_POST['name'];
    $city= $_POST['city'];
    $age = $_POST['age'];
    if ( !empty($id)&&!empty($name)&&!empty($city)&&!empty($age) )
    {
        $sql="INSERT INTO students (id, name, city, age) VALUES ($id, '$name',
'$city', $age)";
        $result = mysqli_query($dbConn,$sql);
        if (!$result)
        {
            die('Грешка!!!');
        }
        echo "Добавихте един запис.";
    }
    else
        echo "Не сте въвели всички данни!!!";
}
?>

```

8. Извличане на данни

За извличане на данни от MySQL бази данни се използва SQL командата SELECT.

Синтакси:

```

SELECT [fieldnames]
    AS [alias]
FROM [tablename]
WHERE [criteria]
GROUP BY[fieldnames]
HAVING[criteria]
ORDER BY [fieldname to sort on] [DESC]
LIMIT [offset, maxrows]

```

SELECT [имена на полета]- Указват се полетата, които ще се извлекът. Ако искаме да изведем всички полета може да напишем символа *.

AS – Указва псевдоним за групиране на две или повече полета заедно, така че да могат да се адресират с една променлива.

FROM–Указват се таблиците, от които искаме да извлечем информация.

WHERE – Задават се критериите за филтриране на данни. Чрез тази клауза се определя кой запис да бъде избран. В клауза *WHERE* се използват оператори за сравнение (<, <=, >, >=, =, !=), *LIKE* и % (*LIKE* предоставя възможност за сравняване на части от текст или числа, а % се използва, като

заместител на символи), логически оператор (*and, or, not*). Чрез % може да реализираме търсене без да знаем точното съдържание на поле.

GROUP BY – задават се колоните. Комбинира редовете, които имат еднакви стойности в зададените колони.

HAVING - Използва се с критерия **GROUP BY**. Връща само записите, които отговарят на условия израз. Използва се в **SQL** команда, съдържаща агрегатни функции, задава условия към полета, които са обобщени в командата **SELECT**.

ORDER BY – Сортиране на данни по конкретно поле. Ако данните трябва да бъдат в намаляващ ред се задава и думата **DESC**.

LIMIT – Ограничава се броя на върнатите данни и да се зададе отместване от първия запис.

Пример 7. Да се извлекат всички записи от таблица **students** на БД, чрез PHP скрипт.

Заявка:

```
SELECT * FROM students;
```

Създадената заявка трябва да се изпрати на сървъра и да се съхрани отговора чрез функция `mysql_query()`.

Код:

за драйвер **php_mysql.dll**

```
<?php
    include "config.php";
    $result =mysql_query("SELECT * FROM students");
?>
```

за драйвер **php_mysqli.dll**

```
<?php
    include "config.php";
    $result =mysqli_query($dbConn, "SELECT * FROM students");
?>
```

Необходимо е да се изведат данните от променлива `$result` като се използва една от следните функции.

- ✓ **mysql_fetch_assoc()** – връща ред от резултата във вид на асоциативен масив
- ✓ **mysql_fetch_row()** - връща ред от запитване към база от данни като номериран масив
- ✓ **mysql_fetch_array()** - връща ред от резултата във вид на асоциативен, номериран масив или и двата. По подразбиране всяка данна се среща в масива по два пъти: веднъж с цифров индекс (`$row[0]`, `$row[1]` и т.н.) и веднъж с индекс низ, съвпадащ с името на полето в таблицата (`$row['Type']` и т.н.). Функцията `mysql_fetch_array()` има втори незадължителен параметър, който указва вида, в който да бъдат върнати данните. Ако вторият параметър на функцията е `MYSQL_BOTH` (това е стойността по подразбиране), функцията връща масив, съдържащ данните по 2 пъти – с цифрови и низови индекси; ако стойността му е `MYSQL_ASSOC`, функцията връща масив, съдържащ само данните с низови индекси и ако стойността му е `MYSQL_NUM`, функцията връща масив, съдържащ само данните с числови индекси.

Функциите връщат масив, ако има извлечен запис и `false` – ако няма записи. Всички тези функции конвертират един запис от резултата в масив, който след това може да се използва.

В примера е използван асоциативен масив. Всеки елемент на получения масив е поле от реда, като ключ на елемента е името на полето в реда.

Код:

за драйвер **php_mysql.dll**

```
.....
    $row = mysql_fetch_assoc($result);
    echo "ID: ".$row['id'].", Name:".$row['name'].", City:".$row['city'].",
    Age:".$row['age']. "<br/>";
.....
```

за драйвер **php_mysqli.dll**

```
.....
    $row = mysqli_fetch_assoc($result);
    echo "ID: ".$row['id'].", Name:".$row['name'].", City:".$row['city'].",
    Age:".$row['age']. "<br/>";
.....
```


Тъй като функцията връща само един запис, а трябва да се разпечата 5 записа, то е необходимо да се използва оператор за цикъл.

за драйвер `php_mysql.dll`

`select.php`

```
<?php
include "config.php";
$result =mysql_query("SELECT * FROM students");
while($row = mysql_fetch_assoc($result)){
    echo " Факултетен номер: ".$row['id'].", Име, Фамилия:
        ".$row['name'].", Адрес:".$row['city'].", Възраст:".$row['age']."<br>";
}
?>
```

за драйвер `php_mysqli.dll`

`select.php`

```
<?php
include "config.php";
$result =mysqli_query($dbConn, "SELECT * FROM students");
while($row = mysqli_fetch_assoc($result)){
    echo " Факултетен номер: ".$row['id'].", Име, Фамилия:
        ".$row['name'].", Адрес:".$row['city'].", Възраст:".$row['age']."<br>";
}
?>
```

Пример 8. Да се извлекат всички записи от таблица `students` на БД (Пример 5.) и да се разпечатат като подреден списък.

за драйвер `php_mysql.dll`

`select_list.php`

```
<?php
include "config.php";
$result =mysql_query("SELECT * FROM students");
echo "<ol>";
while($row = mysql_fetch_array($result)){
    echo "<li>".$row['id']." ".$row['name']." ".$row['city']."
        ".$row['age']."</li>";
}
echo "</ol>";
?>
```

за драйвер `php_mysqli.dll`

`select_list.php`

```
<?php
include "config.php";
$result =mysqli_query($dbConn, "SELECT * FROM students");
echo "<ol>";
while($row = mysqli_fetch_array($result)){
    echo "<li>".$row['id']." ".$row['name']." ".$row['city']."
        ".$row['age']."</li>";
}
echo "</ol>";
?>
```

9. Актуализиране на данни в MySQL таблица

SQL UPDATE командата се използва за промяна на съществуващите данни в MySQL таблица.

Синтаксис:

```
UPDATE tablename SET field1=newvalue1, field2=newvalue2,...[WHERE Clause]
```

`tablename` – Задава се таблицата, в която ще се променят данните.

`field1, field2,...` – Задават се полетата, в които ще се променят данните.

`newvalue1, newvalue2,...newvalueN` – Задават се новите стойности на указаните полета.

`WHERE` – Задават се критериите за филтриране на данни. Чрез тази клауза се определя, кой запис да бъде променен.

- ✓ Може да се обновяват едно или повече полета едновременно.
- ✓ Може да се зададе всякакво условие като се използва клаузата `WHERE`.
- ✓ Може да се обновяват стойностите само на една таблица (не може да се обновяват две таблици едновременно).

Ако тази клауза бъде пропусната ще се обновят всички записи.

Пример 9. Да се актуализират данните за студент с факултетен номер 61444444, чрез PHP скрипт. Да се промени града в адреса от Варна на София.

за драйвер `php_mysql.dll`

update.php

```
<?php
include "config.php";
$sql = "UPDATE students SET city='София' WHERE id=61444444";
mysql_query($sql);
$result = mysql_query("SELECT * FROM students");
while($row=mysql_fetch_array($result))
{
    echo $row['id'].", ".$row['name'].", ".$row['city'].", ".$row['age']."<br>";
}
?>
```

за драйвер `php_mysqli.dll`

update.php

```
<?php
include "config.php";
$sql = "UPDATE students SET city='София' WHERE id=61444444";
mysqli_query($dbConn,$sql);
$result = mysqli_query($dbConn,"SELECT * FROM students");
while($row=mysqli_fetch_array($result))
{
    echo $row['id'].", ".$row['name'].", ".$row['city'].", ".$row['age']."<br>";
}
?>
```

10. Изтриване на данни от MySQL таблица

Синтаксис:

DELETE FROM tablename [WHERE Clause]

tablename – Задава се таблицата, от която ще изтрием данните.

WHERE – Задават се критериите за филтриране на данни. Чрез тази клауза се определя кой запис да бъде изтрит.

Ако клаузата WHERE не е зададена, всички записи ще бъдат изтрети от дадената MySQL таблица.

Записи могат да се изтриват само от една таблица. (Не може да се изтриват записи от две или повече таблици едновременно).

Пример 10. Да се изтрият данните за студент с факултетен номер 61444444, чрез PHP скрипт.

за драйвер `php_mysql.dll`

delete.php

```
<?php
include "config.php";
$sql = "DELETE FROM students WHERE id=61444444";
$result = mysql_query($sql);
$result = mysql_query("SELECT * FROM students");
while($row=mysql_fetch_array($result))
{
    echo $row['id']." ".$row['name']." ".$row['city']." ".$row['age']."<br>";
}
?>
```

за драйвер `php_mysqli.dll`

delete.php

```
<?php
include "config.php";
$sql = "DELETE FROM students WHERE id=61444444";
$result = mysqli_query($dbConn,$sql);
$result = mysqli_query($dbConn,"SELECT * FROM students");
while($row=mysqli_fetch_array($result))
{
    echo $row['id']." ".$row['name']." ".$row['city']." ".$row['age']."<br>";
}
?>
```

11. Изтриване на таблица от базата данни

В `mysql_query` командата се подава SQL команда за изтриване на таблица от БД (`DROP TABLE student61.students`).

Пример 11. Да се изтрие таблица “students” от тестова база от данни “student61”, чрез PHP скрипт.

за драйвер `php_mysql.dll`

`delete_table.php`

```
<?php
include "config.php";
$sql = 'DROP TABLE student61.students';
$result = mysql_query( $sql);
if(!$result )
{
    die('Грешка при изтриване на таблицата: ' . mysql_error());
}
echo "Таблицата е изтрита.";
?>
```

за драйвер `php_mysqli.dll`

`delete_table.php`

```
<?php
include "config.php";
$sql = 'DROP TABLE student61.students';
$result = mysqli_query($dbConn, $sql);
if(!$result )
{
    die('Грешка при изтриване на таблицата!!!' );
}
echo "Таблицата е изтрита.";
?>
```

12. Изтриване на базата данни

В `mysql_query` командата се подава SQL команда за изтриване на БД (`DROP DATABASE student61`).

Пример 12. Да се изтрие тестова база от данни “student61”, чрез PHP скрипт.

за драйвер `php_mysql.dll`

`delete_db.php`

```
<?php
include "config.php";
$sql = 'DROP DATABASE student61';
$result = mysql_query( $sql);
if(! $result )
{
    die('Грешка при изтриване на базата данни: ' . mysql_error());
}
echo "Базата данни е изтрита.";
?>
```

за драйвер `php_mysqli.dll`

`delete_db.php`

```
<?php
include "config.php";
$sql = 'DROP DATABASE student61';
$result = mysqli_query($dbConn, $sql);
if(! $result )
{
    die('Грешка при изтриване на базата данни!!!');
}
echo "Базата данни е изтрита.";
?>
```

Създаването на тестовата база от данни и таблица чрез `phpMyAdmin`

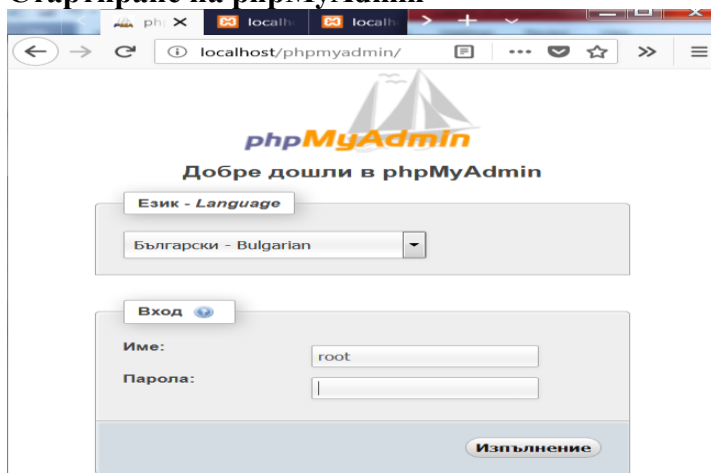
`phpMyAdmin` е проект с отворен код, който позволява достъп до бази от данни на MySQL посредством графичен интерфейс.

С помощта на този софтуер може:

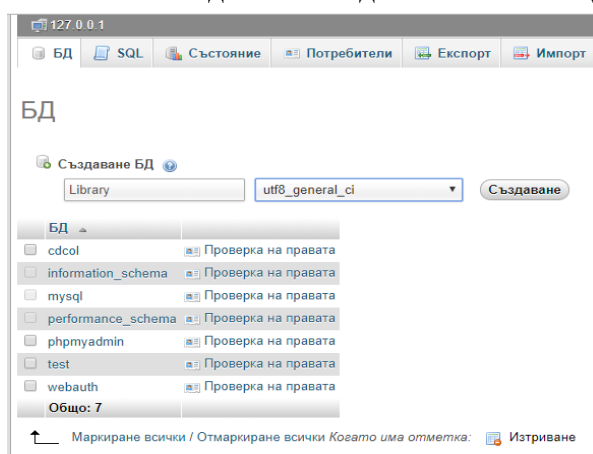
- ✓ да изтривате и създавате бази от данни.
- ✓ да създавате, редактирате и изтривате таблици.
- ✓ да създавате, редактирате и изтривате полета.

- ✓ да въвеждате запитвания към MySQL.
- ✓ да преглеждате и отпечатвате структура на таблици.
- ✓ да създавате код на PHP.
- ✓ да преглеждате данни в табличен вид.
- ✓ да създавате релации между таблиците.

1. Стартиране на phpMyAdmin



2. Създаване на база от данни и задаване на колагия



3. Създаване на таблица

