

Тема 3. Основни действия за работа с MySQL в PHP

1

1. Установяване на връзка със сървър и БД

Използва се функцията `mysql_connect`:

```
resource mysql_connect ([ string $server =  
ini_get("mysql.default_host") [, string $username =  
ini_get("mysql.default_user") [, string $password =  
ini_get("mysql.default_password") [, bool $new_link = false [, int  
$client_flags = 0 ]]]]) )
```

3

В тази тема:

1. Установяване на връзка със сървър и БД
2. Избиране на БД
3. Изпращане на SQL заявка и получаване на резултата
 - 3.1. SELECT заявка
 - 3.2. Извличане на резултатите при успешна заявка
 - 3.3. Вмъкване на запис в таблица
 - 3.4. Промяна и изтриване на редове от таблица

2

server – сървър и номер на порт "hostname:port" или път до локален ресурс: ":/path/to/socket" за localhost.

username – потребителско име, подразбиращата стойност се дефинира с `mysql.default_user`. В SQL safe mode този параметър се игнорира.

password – подразбираща се стойност се задава с `mysql.default_password`. В SQL safe mode този параметър се игнорира.

new_link – при следващо извикване на `mysql_connect` със същите аргументи нова връзка няма да се създаде, освен ако не се посочи изрично.

mysql_connect – комбинация от константи: 128 (позволява обработка на LOAD DATA LOCAL), MYSQL_CLIENT_SSL, MYSQL_CLIENT_COMPRESS, MYSQL_CLIENT_IGNORE_SPACE or MYSQL_CLIENT_INTERACTIVE.

4

Връщани стойности

При успех връща идентификатора на връзката с MySQL сървър. При неуспех връща FALSE.

5

2) Използване на променливи съхраняващи стойностите на хоста, потребителското име и парола:

primer-3-01.php

```
<?php
$host_name = "localhost";
$user_name="student";
$user_password="test";

@$link1=mysql_connect($host_name,$user_name,$user_password);
if (!$link1) {
    die('Връзката не може да бъде установена: ' .
    mysql_error());
}
echo 'Връзката е установена успешно!';
?>
```

7

Примери:

1) Свързване към localhost

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
if (!$link) {
    die('Връзката не може да бъде установена: ' . mysql_error());
}
echo 'Връзката е установена успешно';
mysql_close($link);
?>
```

6

3) Използване на формата hostname:port

```
<?php
// установяваме връзка с example.com през порт 3307
$link = mysql_connect('example.com:3307', 'mysql_user', 'mysql_password');
if (!$link) {
    die('Връзката не може да бъде установена: ' . mysql_error());
}
echo 'Връзката е установена успешно';
mysql_close($link);

// установяваме връзка с localhost през порт 3307
$link = mysql_connect('127.0.0.1:3307', 'mysql_user', 'mysql_password');
if (!$link) {
    die('Връзката не може да бъде установена: ' . mysql_error());
}
echo 'Връзката е установена успешно';
mysql_close($link);
?>
```

8

2. Избиране на БД

Задаване на текущата БД, която се асоциира със специфичен идентификатор на връзка.

```
bool mysql_select_db ( string $database_name  
[, resource$link_identifier = NULL ] )
```

9

Примери:

1) Избиране на БД adventureworks от установена връзка link1
primer-3-02.php

```
<?php  
$host_name = "localhost";  
$user_name="student";  
$user_password="test";  
  
@$link1=mysql_connect($host_name,$user_name,$user_password);  
if (!$link1) {  
    die('Връзката не може да бъде установена: ' . mysql_error());  
}  
else {  
    $db_name="adventureworks";  
    $db_selected = mysql_select_db($db_name, $link1);  
    if (!$db_selected) {  
        die ('Не може да се свърже към БД : ' . mysql_error());  
    }  
}  
?>
```

11

database_name – име на избраната БД

link_identifier – ако не е посочен идентификатор на връзка, приема се последния използван линк, отворил mysql_connect(). Ако не намери връзка, или тя не е установена E_WARNING грешка се генерира.

Връщани стойности:

Връща TRUE при успех или FALSE при неуспех.

10

2) Избиране на БД от две връзки – към локален и отдалечен хост

```
<?php  
@$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');  
@link2 = mysql_connect('mysql.firm.com', 'mysql_user2', 'mysql_password2');  
  
if (!$link) {  
    die('Не е свързан : ' . mysql_error());  
}  
  
if (!$link2) {  
    die('Не е свързан : ' . mysql_error());  
}  
  
// Свързване към БД  
$db_selected = mysql_select_db('test', $link);  
if (!$db_selected) {  
    die ('Не може да се свърже към БД : ' . mysql_error());  
}  
$db_selected = mysql_select_db('spk', $link2);  
if (!$db_selected) {  
    die ('Не може да се свърже към БД : ' . mysql_error());  
}  
?>
```

12

3. Изпращане на SQL заявка и получаване на резултата

`mysql_query()` изпраща единична заявка (множество заявки не се поддържат) към текущо активната база от данни на сървъра, която е асоциирана с указания `link_identifier`.

```
mixed mysql_query ( string $query [, resource $link_identifier = NULL ] )
```

13

Връщани стойности:

За **SELECT**, **SHOW**, **DESCRIBE**, **EXPLAIN** и други изрази връщащи резултатно множество `mysql_query()` връща променлива от тип **resource** при успешно изпълнение и **FALSE** при грешка.

При други типове SQL израз, **INSERT**, **UPDATE**, **DELETE**, **DROP**, и т.н, `mysql_query()` връща **TRUE** при успешно изпълнение и **FALSE** при грешка.

Върнатият резултатен ресурс ще трябва да бъде предаден към **mysql_fetch_array()** или друга функция за работа с резултатни таблици, за да може да се осъществи достъп до върнатите данни.

15

Параметри:

query – SQL заявка. Низът на заявката не трябва да завършва с точка и запетая.

link_identifier - връзката към MySQL. Ако не е указана, се използва последната създадена връзка от `mysql_connect()`. Ако не намери такава връзка, ще направи опит да установи връзка, все едно е изпълнена `mysql_connect()` без параметри. Ако случайно не е намерена или създадена никаква връзка, ще се генерира предупреждение от ниво `E_WARNING`.

14

Може да се използва функцията **mysql_num_rows()** за определяне на броя на върнатите редове от **SELECT** заявката и функция **mysql_affected_rows()** за определяне колко реда са засегнати от **DELETE**, **INSERT**, **REPLACE**, или **UPDATE** изказа.

```
int mysql_num_rows ( resource $result )
```

```
int mysql_affected_rows ([ resource $link_identifier = NULL ] )
```

mysql_query() няма да изпълни заявката и ще върне **FALSE** ако потребителят няма права за достъп да таблицата/таблиците използвани в заявката.

16

3.1. SELECT заявка

Примери:

1) Заявката по-долу има **невалиден синтаксис**, така `mysql_query()` няма да изпълни заявката успешно и ще върне FALSE.

```
<?php
$result = mysql_query('SELECT * WHERE 1=1');
if (!$result) {
    die('Невалидна заявка: ' . mysql_error());
}

?>
```

17

3) Заявка, която получава стойностите на аргументите от променливи
Primer-3-04.php

```
// Стойностите могат да се въведат от потребителя
$customer_ID = 200;
$sub_total = 1000;

// Формулиране на заявката
// mysql_real_escape_string() освобождава стринга на SQL заявката от специалните символи; sprintf -
// връща форматиран низ
$sql = sprintf("SELECT SalesOrderID, OrderDate, DueDate, ShipDate, CustomerID, SubTotal FROM
salesorderheader WHERE CustomerID = '%s' AND SubTotal > '%s'",
    mysql_real_escape_string($customer_ID),
    mysql_real_escape_string($sub_total));

$result = mysql_query($sql);
if (!$result) {
    $message = 'Невалидна заявка: ' . mysql_error();
}
else {
    echo "Брой редове в заявката: " . mysql_num_rows($result);
}
```

19

2) Валидна заявка и `mysql_query()` връща resource

primer-3-03.php

// Продължение на пример primer-3-02.php

```
$sql='SELECT ProductID, Name, ProductNumber, StandardCost, ListPrice FROM
product';
$result = mysql_query($sql);
if (!$result) {
    die('Невалидна заявка: ' . mysql_error());
}
else {
    echo "Брой редове в заявката: " . mysql_num_rows($result);
}
```

18

3.2. Извличане на резултатите при успешна заявка

Обработката на резултата от успешната заявка се извършва с няколко функции:

```
mysql_fetch_array()
mysql_fetch_assoc()
mysql_fetch_row()
```

20

mysql_fetch_array ()

Връща масив, съответстващ на обработените резултати от заявката, съдържащ указател към данните като заглавен ред.

array mysql_fetch_array (resource \$result [, int \$result_type = MYSQL_BOTH])

Параметри:

result – обработения резултат от изпълнението на mysql_query().

result_type – тип на връщания масив. Той е константа има значения: MYSQL_ASSOC, MYSQL_NUM и MYSQL_BOTH.

21

Примери:

1) Заявка с използване на псевдоними за дублиращите се имена на колони

```
SELECT table1.field AS foo, table2.field AS bar FROM table1, table2
```

23

Връщани стойности:

Връща масив от редове, съответстващ на обработената заявка или **FALSE**, ако няма редове.

Типът на връщания масив зависи от значението на параметъра result_type:

MYSQL_BOTH (по подразбиране) – връща масив състоящ се, както от асоциираните индекси, така и от числените;

MYSQL_ASSOC – връща само асоциативни индекси (аналогично на функцията mysql_fetch_assoc());

MYSQL_NUM – връща само числови стойности (аналогично на функцията mysql_fetch_row());

Ако няколко колони в резултата имат еднакви наименования, тогава се връща последната от тези колони. За да се получи достъп до другите колони със същите имена, трябва да се използва числовите индекси на масива или псевдоними в заявката. При използване на псевдоними, не могат да се използват настоящите имена на колони.

22

2) Обработка на резултата с mysql_fetch_array() с MYSQL_ASSOC

```
primer-3-05.php
// .... продължение на primer-3-04.php ....
while ($row = mysql_fetch_array($result, MYSQL_NUM))
{
    printf("<br>");
    printf("Поръчка ID: %s Поръчка дата: %s Клиент
ID: %s Общо: %s" , $row[0], $row[1], $row[4],
$row[5]);
}

mysql_free_result($result);
```

24

3) Обработка на резултата с `mysql_fetch_array()` с `MYSQL_ASSOC`

```
primer-3-06.php
// .... продължение на primer-3-04.php ....
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
    printf("<br>");
    printf("Поръчка ID: %s Поръчка дата: %s Клиент ID: %s  
Общо: %s", $row["SalesOrderID"], $row["OrderDate"], $row["CustomerID"], $row["SubTotal"]);
}

mysql_free_result($result);
```

25

mysql_fetch_assoc()

Връща резултатите от заявката като асоцииран масив

```
array mysql_fetch_assoc ( resource $result )
```

result – обработения от заявката резултат.

Връща асоциативен масив от редове, съответстващ на обработената заявка или **FALSE**, ако няма редове.

27

4) Обработка на резултата с `mysql_fetch_array()` с `MYSQL_BOTH`

```
primer-3-07.php
// .... продължение на primer-3-04.php ....
while ($row = mysql_fetch_array($result, MYSQL_BOTH)) {
    printf("<br>");
    printf("Поръчка ID: %s Поръчка дата: %s Клиент ID: %s  
Общо: %s", $row[0], $row[1], $row["CustomerID"], $row["SubTotal"]);
}

mysql_free_result($result);
```

26

Пример:

```
while ($row = mysql_fetch_assoc($result)) {
    echo $row["userid"];
    echo $row["fullname"];
    echo $row["userstatus"];
}
```

28

mysql_fetch_row()

Връща резултата от заявката в масив с числови индекси

```
array mysql_fetch_row ( resource $result )
```

result – обработения от заявката резултат.

Връща асоциативен масив от редове, съответстващ на обработената заявка или **FALSE**, ако няма редове.

29

3.3. Вмъкване на запис в таблица

```
primer-3-08.php
// ... в продължение на primer-3-02.php ...
mysql_query('SET NAMES utf8');

$department_ID = 999;
$name = "Компютърен център";
$group_name = "IT инфраструктура";

$sql = "INSERT into department (DepartmentID, Name, GroupName)
VALUES ('$department_ID', '$name', '$group_name')";

$result = mysql_query ($sql)
or die ("Не може да се изпълни заявката: ". mysql_error());

echo "Информацията за новия отдел е добавена!";
```

31

Пример:

```
$row = mysql_fetch_row($result);

echo $row[0]; // 42
echo $row[1]; // email
?>
```

30

3.4. Промяна и изтриване на редове от таблица

Промяна на ред от таблица се извършва с **UPDATE**, а изтриване с **DELETE**. Те се използват чрез **mysql_query()** подобно, подобно на INSERT (вж. т.3.3).

32