

2.9. ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ PHP И MySQL

После установки и настройки базы данных MySQL приступим к рассмотрению вопроса, каким образом PHP-скрипты соединяются с базой данных, каким образом происходит обращение к нужной таблице и как обрабатываются результаты, полученные из базы данных.

Самая первая функция, изучаемая нами в данном параграфе, производит инициализацию подключения к базе данных MySQL. Функция называется **mysql_connect (\$host, \$user, \$password)**. Данной функции на вход подается три параметра:

- **\$host** – имя сервера, на котором находится база данных, в большинстве случаев используется локальный хост **localhost**;
- **\$user** – имя пользователя, которого мы создавали для подключения к базе данных. Можно использовать **root** в качестве универсального пользователя, у которого нет пароля. Но делать это можно только в период тестирования.
- **\$password** – пароль пользователя, который задавался при создании базы данных.

Результат функции **mysql_connect** сохраняется в некоторую переменную, например, **\$db**, которая впоследствии может быть использована в запросах.

Пример:

```
$db=mysql_connect("localhost","ivan","123456");
```

или для пользователя по умолчанию:

```
$db=mysql_connect("localhost","root","");
```

Функция **mysql_close([\$db_identifier])** закрывает соединение с сервером MySQL. Возвращает TRUE, в случае успешного завершения, или FALSE, в случае возникновения ошибки. Использование **mysql_close()** не обязательно для непостоянных соединений (они автоматически закрываются в конце скрипта).

Корректная установка соединения может выглядеть следующим образом:

```
<?
$db = mysql_connect("localhost","login","password");
If (!$db) {
exit("Невозможно установить соединение: ". mysql_error());}
else
{echo "Соединение установлено";
mysql_close($db);}
?>
```

После установки соединения происходит выбор базы данных. **mysql_select_db(\$database_name, [\$db_identifier])** выбирает для работы указанную базу данных **\$database_name** на сервере, на который ссылается переданный указатель **\$db_identifier**. Если параметр указателя опущен, используется последнее открытое соединение. Если нет ни

одного открытого соединения, функция попытается соединиться с сервером аналогично функции `mysql_connect()`, вызванной без параметров. Пример:

```
mysql_select_db("my_db", $db);
```

Запрос к базе данных, поступающий из сценария PHP, по сути представляет собой команду MySQL, заключенную в функцию `mysql_query()`. Именно данная функция позволяет организовать основные операторы SQL – SELECT, INSERT, UPDATE, DELETE. Общий вид функции: `mysql_query($query, [$db_identifier])`, где `$query` – это строка с запросом MySQL, `$db_identifier` – указатель на соединение с базой данных. Если указатель не указан, то используется последнее открытое соединение. Только для запросов SELECT `mysql_query()` возвращает указатель на результат запроса или FALSE, если запрос не был выполнен. В остальных случаях (INSERT, UPDATE, DELETE), `mysql_query()` возвращает TRUE, в случае успешного запроса, и FALSE, в случае ошибки. Значение не равное FALSE говорит о том, что запрос был выполнен успешно. О количестве затронутых или возвращенных рядов запрос не сообщает. Вполне возможна ситуация, когда успешный запрос не затронет ни одного ряда.

Пример обработки запроса – результат помещается в переменную `$result` для дальнейшей обработки:

```
$result = mysql_query("SELECT * FROM articles", $db);
```

В некоторых случаях, особенно когда добиться правильного написания sql-запроса сразу не удастся, рекомендуется оформлять строку с запросом в виде отдельной переменной, например `$query`. В этом случае всегда можно распечатать запрос и протестировать его в панели phpMyAdmin.

```
$query = "SELECT id, title, author FROM articles";  
print_r($query);  
$result = mysql_query($query);
```

После получения информации о выполнении запроса необходимо обработать данные на стороне PHP. При этом удобно, когда данные, полученные по запросу, оформляются в виде массива.

Функция `mysql_fetch_row` – возвращает строку (результат запроса) в виде массива с числовым индексом.

Функция `mysql_fetch_array` – возвращает строку (результат запроса) в виде ассоциативного массива. Ключевыми индексами такого массива являются названия полей в таблице mysql.

Функция `mysql_result` – возвращает один элемент массива данных. Подходит для обработки агрегатных запросов.

Обработка всех данных, полученных по запросу, с использованием функций `mysql_fetch_row()` и `mysql_fetch_array()` осуществляется в циклических конструкциях. Можно использовать цикл while, do-while или for для этих целей. Далее приведем пример обработки запроса с помощью цикла while.

```

<?php
    $sql = "SELECT * FROM articles";
    $result = mysql_query($sql);
    while ($myrow = mysql_fetch_array($result)) {
        echo $myrow["id"] . " " . $myrow["title"] . " " .
        $myrow["author"] . "<br>";
        echo $myrow["description"] . "<br>";
    }
?>

```

Если планируете использовать цикл **for**, то важно понимать, как можно заранее получить общее количество записей по запросу. Сделать это можно с помощью функции **mysql_num_rows()**.

Пример использования

```

<?
$db=mysql_connect("localhost","root","");
mysql_select_db("my_db",$db);
$sql = "select id, title, description, author, date_created
from articles";
$result=mysql_query($sql,$db);
$count = mysql_num_rows($result);
For ($i=0; $i<=$count; $i++) {
    $myrow=mysql_fetch_array($result);
    echo $myrow["id"] . " " . $myrow["title"] . " " . $my-
    row["author"] . "<br>";
    echo $myrow["description"]."<br>";
}
?>

```

Для удобства разработчиков были введены Функции возврата сообщений об ошибках из базы данных.

- **mysql_error([\$db_identifier])** – возвращает текст ошибки последней операции с MySQL.
- **mysql_errno([\$db_identifier])** – численное значение сообщения об ошибках от предыдущей операции MySQL. По этому коду можно расшифровать тип возникающей ошибки. Так, код ошибки 1049 сообщает, что имя базы данных неизвестно (возможно, при написании кода была допущена ошибка в имени или на данном сервере в принципе нет такой базы данных); код 1146 – указанной таблицы не найдено в базе данных.

```
<?php
$db = mysql_connect("localhost", "root", "");
mysql_select_db("abcdefg", $db);
echo mysql_errno($db) . ": " . mysql_error($db) . "<br>";

mysql_select_db("my_db", $db);
mysql_query("SELECT * FROM abcdefg", $db);
echo mysql_errno($db) . ": " . mysql_error($db) . "<br>";
?>
```

Результатом выполнения данного примера будет что-то подобное:

```
1049: Unknown database 'abcdefg'
1146: Table 'my_db.abcdefg' doesn't exist
```

Другие коды ошибок можно посмотреть на ресурсе [4].