

6.6.1. Установка соединения

Установить соединение можно двумя способами:

```
$db = mysqli_connect(<Имя хоста>, <Имя пользователя>, <Пароль>,  
    <База данных>);  
$db = new mysqli(<Имя хоста>, <Имя пользователя>, <Пароль>,  
    <База данных>);
```

Все параметры необязательные. Процедурный стиль возвращает идентификатор соединения, а в случае неудачи возвращает false. Проверить соединение можно следующим образом:

```
if (@$db = mysqli_connect("localhost", "root", "123456", "tests")) {  
    // Выполняем работу с базой данных  
}  
else {  
    echo "Не удалось установить подключение к базе данных";  
}
```

При объектном стиле такой способ не подходит. Проверить отсутствие ошибок при подключении позволяет функция `mysqli_connect_errno()`. Проверить соединение можно следующим образом:

```
@$db = new mysqli("localhost", "root", "123456", "tests");  
if (!mysqli_connect_errno()) {  
    // Выполняем работу с базой данных  
}  
else {  
    echo "Не удалось установить подключение к базе данных";  
}
```

Также можно обратиться к свойству `connect_errno`:

```
@$db = new mysqli("localhost", "root", "123456", "tests");  
if (!$db->connect_errno) {  
    // Выполняем работу с базой данных  
}  
else {  
    echo "Не удалось установить подключение к базе данных";  
}
```

Закреть соединение при процедурном стиле позволяет функция `mysqli_close()`:

```
mysqli_close(<Идентификатор>);
```

При объектном стиле используется метод `close()`:

```
<Экземпляр класса>->close();
```

Приведем код для подключения к серверу MySQL. Процедурный стиль:

```
if (@$db = mysqli_connect("localhost", "root", "123456", "tests")) {  
    // Выполняем работу с базой данных
```

```

mysqli_close($db); // Закрываем соединение
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

Объектный стиль:

```

@$db = new mysqli("localhost", "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    // Выполняем работу с базой данных
    $db->close(); // Закрываем соединение
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

6.6.2. Выбор базы данных

Выбрать базу данных можно при подключении в функции `mysqli_connect()` или в конструкторе класса. При процедурном стиле выбор базы данных уже после подключения осуществляет функция `mysqli_select_db()`. Формат функции:

```
mysqli_select_db(<Идентификатор>, <Имя базы данных>);
```

Пример:

```

if (@$db = mysqli_connect("localhost", "root", "123456")) {
    mysqli_select_db($db, "tests");
    // Выполняем работу с базой данных
    mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

При объектном стиле используется метод `select_db()`. Формат метода:

```
<Экземпляр класса>->select_db(<Имя базы данных>);
```

Пример:

```

@$db = new mysqli("localhost", "root", "123456");
if (!mysqli_connect_errno()) {
    $db->select_db("tests");
    // Выполняем работу с базой данных
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

6.6.3. Выполнение запроса к базе данных

Выполнить запрос к базе данных при процедурном стиле позволяет функция `mysqli_query()`. Формат функции:

```
mysqli_query(<Идентификатор>, <SQL-запрос>);
```

ВНИМАНИЕ!

В конце SQL-запроса не следует указывать точку с запятой.

Функция возвращает идентификатор результата. Для удаления идентификатора результата и освобождения используемых ресурсов применяется функция `mysqli_free_result()`. Формат функции:

```
mysqli_free_result(<Идентификатор результата>);
```

Получить все записи таблицы `City` позволяет следующий код:

```
if (@$db = mysqli_connect("localhost", "root", "123456", "tests")) {
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        // Обрабатываем извлеченные записи
        mysqli_free_result($res);
    }
    mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

Выполнить запрос к базе данных при объектном стиле позволяет метод `query()`. Формат метода:

```
<Экземпляр класса>->query(<SQL-запрос>);
```

Метод возвращает экземпляр результата. Для удаления экземпляра результата следует применить метод `close()` или `free()`. Формат метода:

```
<Экземпляр результата>->close();
```

Получить все записи таблицы `City` позволяет следующий код:

```
@$db = new mysqli("localhost", "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    if ($res = $db->query('SELECT * FROM `City`')) {
        // Обрабатываем извлеченные записи
        $res->close();
    }
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

Для того чтобы записи возвращались в нужной кодировке, следует после подключения выполнить запрос

```
mysqli_query($db, 'SET NAMES cp1251'); // Для кодировки windows-1251
mysqli_query($db, 'SET NAMES utf8'); // Для кодировки UTF-8
```

при процедурном стиле или

```
$db->query('SET NAMES cp1251'); // Для кодировки windows-1251
$db->query('SET NAMES utf8'); // Для кодировки UTF-8
```

при объектном стиле.

6.6.4. Обработка результата запроса

Для обработки результата запроса при процедурном стиле используются следующие функции:

❑ `mysqli_num_rows(<Идентификатор результата>)` — возвращает число записей в результате:

```
$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cp1251");
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        echo mysqli_num_rows($res) . "<br>";
        mysqli_free_result($res);
    }
    mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

❑ `mysqli_field_count(<Идентификатор соединения>)` — возвращает число полей в результате последнего SQL-запроса:

```
$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cp1251");
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        echo mysqli_field_count($db) . "<br>";
        mysqli_free_result($res);
    }
    mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

❑ `mysqli_fetch_array(<Идентификатор результата>, [<Флаг>])` — возвращает результат в виде списка и (или) ассоциативного массива и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`. Тип возвращенного результата зависит от необязательного параметра `<Флаг>`, который может принимать следующие значения:

- `MYSQLI_BOTH` — результат в виде списка и ассоциативного массива (значение по умолчанию);
- `MYSQLI_NUM` — результат в виде списка;
- `MYSQLI_ASSOC` — результат в виде ассоциативного массива.

Приведем пример, в котором сочетаются все эти варианты:

```
$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cp1251");
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        while ($pole = mysqli_fetch_array($res)) {
            echo $pole[0] . ' - ' . $pole['City'] . '<br>';
        }
        mysqli_free_result($res);
    }
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        $pole = mysqli_fetch_array($res, MYSQLI_BOTH);
        echo $pole[0] . ' - ' . $pole['City'] . '<br>';

        $pole = mysqli_fetch_array($res, MYSQLI_NUM);
        echo $pole[0] . ' - ' . $pole[1] . '<br>';

        $pole = mysqli_fetch_array($res, MYSQLI_ASSOC);
        echo $pole['id_City'] . ' - ' . $pole['City'] . '<br>';
        mysqli_free_result($res);
    }
    mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

❑ `mysqli_fetch_row(<Идентификатор результата>)` — возвращает результат в виде списка и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`. Вызов этой функции эквивалентен вызову `mysqli_fetch_array` с флагом `MYSQLI_NUM`:

```
$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cp1251");
```

```

if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
    while ($pole = mysqli_fetch_row($res)) {
        echo $pole[0] . ' - ' . $pole[1] . '<br>';
    }
    mysqli_free_result($res);
}
mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

- ❑ `mysqli_fetch_assoc(<Идентификатор результата>)` — возвращает результат в виде ассоциативного массива и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`. Вызов этой функции эквивалентен вызову `mysqli_fetch_array` с флагом `MYSQLI_ASSOC`:

```

$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cpl251");
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        while ($pole = mysqli_fetch_assoc($res)) {
            echo $pole['id_City'] . ' - ' . $pole['City'] . '<br>';
        }
        mysqli_free_result($res);
    }
    mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

- ❑ `mysqli_fetch_object(<Идентификатор результата>)` — возвращает результат в виде объекта и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`:

```

$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cpl251");
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        while ($pole = mysqli_fetch_object($res)) {
            echo $pole->id_City . ' - ' . $pole->City . '<br>';
        }
        mysqli_free_result($res);
    }
    mysqli_close($db);
}

```

```
else {
    echo "Не удалось установить подключение к базе данных";
}
```

- ❑ `mysqli_data_seek(<Идентификатор результата>, <Смещение>)` — перемещает указатель результата на выбранную строку. Нумерация строк начинается с нуля:

```
$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cp1251");
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        mysqli_data_seek($res, 1);
        $pole = mysqli_fetch_object($res);
        echo $pole->id_City . ' - ' . $pole->City . '<br>';
        mysqli_free_result($res);
    }
    mysqli_close($db);
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

Для обработки результата запроса при объектном стиле используются следующие методы и свойства:

- ❑ `num_rows` — возвращает число записей в результате:

```
$host = "localhost";
@$db = new mysqli($host, "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    $db->query("SET NAMES cp1251");
    if ($res = $db->query('SELECT * FROM `City`')) {
        echo $res->num_rows . '<br>';
        $res->close();
    }
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

- ❑ `field_count` — возвращает число полей в результате:

```
$host = "localhost";
@$db = new mysqli($host, "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    $db->query("SET NAMES cp1251");
    if ($res = $db->query('SELECT * FROM `City`')) {
        echo $res->field_count . '<br>';
    }
}
```

```

        $res->close();
    }
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

❑ `fetch_array([<Флаг>])` — возвращает результат в виде списка и (или) ассоциативного массива и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`. Тип возвращенного результата зависит от необязательного параметра `<Флаг>`, который может принимать следующие значения:

- `MYSQLI_BOTH` — результат в виде списка и ассоциативного массива (значение по умолчанию);
- `MYSQLI_NUM` — результат в виде списка;
- `MYSQLI_ASSOC` — результат в виде ассоциативного массива.

Следующий пример иллюстрирует все эти варианты:

```

$host = "localhost";
@$db = new mysqli($host, "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    $db->query("SET NAMES cp1251");
    if ($res = $db->query('SELECT * FROM `City`')) {
        while ($pole = $res->fetch_array()) {
            echo $pole[0] . ' - ' . $pole['City'] . '<br>';
        }
        $res->close();
    }
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        $pole = $res->fetch_array(MYSQLI_BOTH);
        echo $pole[0] . ' - ' . $pole['City'] . '<br>';

        $pole = $res->fetch_array(MYSQLI_NUM);
        echo $pole[0] . ' - ' . $pole[1] . '<br>';

        $pole = $res->fetch_array(MYSQLI_ASSOC);
        echo $pole['id_City'] . ' - ' . $pole['City'] . '<br>';
        $res->close();
    }
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```


- ❑ `fetch_row()` — возвращает результат в виде списка и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`. Вызов этого метода эквивалентен вызову `fetch_array` с флагом `MYSQLI_NUM`:

```
$host = "localhost";
@$db = new mysqli($host, "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    $db->query("SET NAMES cp1251");
    if ($res = $db->query('SELECT * FROM `City`')) {
        while ($pole = $res->fetch_row()) {
            echo $pole[0] . ' - ' . $pole[1] . '<br>';
        }
        $res->close();
    }
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

- ❑ `fetch_assoc()` — возвращает результат в виде ассоциативного массива и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`. Вызов этого метода эквивалентен вызову `fetch_array` с флагом `MYSQLI_ASSOC`:

```
$host = "localhost";
@$db = new mysqli($host, "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    $db->query("SET NAMES cp1251");
    if ($res = $db->query('SELECT * FROM `City`')) {
        while ($pole = $res->fetch_assoc()) {
            echo $pole['id_City'] . ' - ' . $pole['City'] . '<br>';
        }
        $res->close();
    }
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}
```

- ❑ `fetch_object()` — возвращает результат в виде объекта и сдвигает внутренний указатель на следующую запись; если записей больше нет, возвращается `false`:

```
$host = "localhost";
@$db = new mysqli($host, "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    $db->query("SET NAMES cp1251");
```

```

if ($res = $db->query('SELECT * FROM `City`')) {
    while ($pole = $res->fetch_object()) {
        echo $pole->id_City . ' - ' . $pole->City . '<br>';
    }
    $res->close();
}
$db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

- ❑ `data_seek(<Смещение>)` — перемещает указатель результата на выбранную строку. Нумерация строк начинается с нуля:

```

$host = "localhost";
@$db = new mysqli($host, "root", "123456", "tests");
if (!mysqli_connect_errno()) {
    $db->query("SET NAMES cp1251");
    if ($res = $db->query('SELECT * FROM `City`')) {
        $res->data_seek(1);
        $pole = $res->fetch_object();
        echo $pole->id_City . ' - ' . $pole->City . '<br>';
        $res->close();
    }
    $db->close();
}
else {
    echo "Не удалось установить подключение к базе данных";
}

```

Функция `mysqli_real_escape_string(<Идентификатор соединения>, <Строка>)` в процедурном стиле и метод `real_escape_string(<Строка>)` в объектном стиле экранируют все специальные символы в строке, учитывая кодировку соединения. Возвращают строку, которую можно безопасно использовать в SQL-запросах:

```

$new_city = "Д'Арк"; // Такие данные передаются из формы
$host = "localhost";
if (@$db = mysqli_connect($host, "root", "123456", "tests")) {
    mysqli_query($db, "SET NAMES cp1251");
    // Экранируем спецсимволы
    $new_city = mysqli_real_escape_string($db, $new_city);
    $query = "INSERT INTO `City` VALUES(NULL, '$new_city')";
    mysqli_query($db, $query);
    if ($res = mysqli_query($db, 'SELECT * FROM `City`')) {
        while ($pole = mysqli_fetch_row($res)) {
            echo $pole[0] . ' - ' . $pole[1] . '<br>';
        }
    }
}

```

```
mysqli_free_result($res);  
}  
mysqli_close($db);  
}  
else {  
    echo 'Не удалось установить подключение к базе данных';  
}
```

ВНИМАНИЕ!

Никогда напрямую не передавайте в SQL-запрос данные, полученные из полей формы. Это потенциальная угроза безопасности. Всегда применяйте функцию `mysqli_real_escape_string()` или метод `real_escape_string()`.

6.7. Операторы MySQL

Операторы позволяют выполнить определенные действия с данными. Например, математические операторы предназначены для арифметических вычислений. Рассмотрим операторы, доступные в MySQL.

ПРИМЕЧАНИЕ

Выполнять SQL-команды мы будем в программе MySQL Command Line Client. Для запуска программы в меню **Пуск** выбираем пункт **Программы | MySQL | MySQL Server 5.5 | MySQL 5.5 Command Line Client**. Откроется черное окошко с запросом ввести пароль. Вводим пароль, заданный при установке сервера MySQL. Если установка проводилась по инструкциям из разд. 4.7, то пароль "123456". В случае успешного входа отобразится приветствие сервера, и программа перейдет в режим ожидания команд. В командной строке набираем `USE tests;` для выбора базы данных.

6.7.1. Математические операторы

Математические операторы:

☐ **+** — сложение:

```
SELECT 8 + 5;
```

☐ **-** — вычитание:

```
SELECT 10 - 5;
```

☐ ***** — умножение:

```
SELECT 10 * 5;
```

☐ **/** — деление:

```
SELECT 10 / 5;  
/* Выведет: 2.0000 */
```

☐ **DIV** — целочисленное деление:

```
SELECT 10 DIV 5;  
/* Выведет: 2 */
```