

Тема: «Знакомство с phpMyadmin. Создание базы данных MySQL. Работа с базой данных»

Цель: Ознакомиться с инструментом создания баз данных phpMyadmin. Научиться создавать базы данных MySQL.

Теоретическая часть

Введение

База данных (сокращенно БД) — это место, в котором хранятся данные сайта. Это могут быть тексты страниц, списки пользователей с их логинами и паролями, каталоги продукции и другое.

База данных состоит из таблиц. Что такое таблица вы знаете из жизни: это строки и столбцы. А на пересечении строк и столбцов располагаются ячейки. В базах данных столбцы часто называют полями. Это легко можно вообразить себе, представив документ Excel. Базой данных будет являться сам документ (книга), а таблицами — каждый лист этой книги.

Проектирование базы данных

Для создания базы данных (БД) необходима определенная работа по планированию. Для самой БД необходимо задать:

- Имя базы данных (БД);
- Пользователь БД (Логин);
- Пароль доступа к базе данных

Следующим шагом необходимо создать таблицу, входящую в базу данных. При этом задаются имя таблицы, наименования и типы полей. Все имена записываются на латинице. Таблица заполняется данными.

Инструменты для работы с базой данных

Работу с базой данных будем производить с помощью визуального редактора PhpMyAdmin. PhpMyAdmin — это оболочка для работы с БД прямо в браузере. С помощью этой программы можно править содержимое таблиц, создавать новые базы данных и таблицы — и все это через веб-интерфейс!

Переменные и функции для работы с базой данных

В SQL довольно много типов переменных, но чаще всего приходится пользоваться следующими:

- **integer** — целочисленный.
 - **text** — большое текстовое поле.
 - **varchar** — не очень большое текстовое поле, при этом мы должны задать его размер (он должен быть степенью двойки: 8, 16, 32, 64, 128, 256 и т.д.).
 - **date** — поле для хранения даты (дата хранится в SQL-формате: год-месяц-день, пример: 2013-06-24).
- id** — номера записей. **AUTO_INCREMENT** увеличивает номер каждый раз на 1.

Работа с БД из PHP осуществляется всего лишь с помощью трех функций:

- **mysql_connect** — соединение с сервером.
- **mysql_select_db** — выбор базы данных.

- **mysql_query** — универсальная функция отправки запросов к БД, с помощью нее можно сделать все.

Далее подробнее про каждую из функций.

Устанавливаем соединение с БД

Прежде, чем начать работать с базой данных из PHP, нужно установить соединение с сервером, на котором эта база находится. Делается это с помощью функции PHP **mysql_connect**, которая принимает 3 параметра: **имя хоста** (сервера), **имя пользователя**, под которым мы работаем с базой и **пароль** для этого пользователя.

Если вы работаете на своем компьютере то это будут «localhost», «root» и пароль в виде пустой строки. Если ваша база данных в интернете — то эти данные дает вам хостер.

После того, как соединение с сервером установлено, нужно выбрать базу данных, с которой мы хотим работать (так как она может быть не одна на этом сервере для этого пользователя). Делается это с помощью функции **mysql_select_db()**, которое принимает параметром имя базы данных, с которой мы хотим установить соединение.

Смотрите пример:

```
//Устанавливаем доступы к базе данных:
$host = 'localhost';
$user = 'root';
$password = '';
mysql_connect($host, $user, $password); //Соединяемся с базой данных
используя наши доступы:
$db_name = 'test';
mysql_select_db($db_name); //Выбираем базу данных (в нашем случае test):
```

Посылаем запросы к базе данных

Запросы к базе данных представляют собой обычные строки, которые мы вставляем в функцию PHP **mysql_query**

```
mysql_query ("SELECT*FROM workers WHERE id>0");
```

Принято правило, по которому команды SQL следует писать в верхнем регистре (то есть большими буквами), а все остальное — в нижнем. Это относится к командам SELECT, UPDATE, FROM, DELETE, WHERE и другим такого рода.

Проблемы с кодировками

Зачастую у новичков возникает проблема с кодировками — казалось бы нормальный русский текст в базу данных записывается абракадаброй или вопросиками. Чтобы не было проблем с кодировками следует придерживаться простых правил:

- Базу данных следует создавать в кодировке utf8_general_ci.
- Документ PHP должен быть в кодировке utf8.
- Таблицы в БД должны быть в utf8_general_ci!
- На всякий случай сразу после команды mysql_select_db добавьте такое запрос:
mysql_query("SET NAMES 'utf8'");

Функции для работы с базой данных

Работа с БД из PHP осуществляется всего лишь с помощью трех функций:

- **mysql_connect** – соединение с сервером (принимает 3 параметра: **имя хоста** (сервера), **имя пользователя**, под которым мы работаем с базой и **пароль** для этого пользователя).
- **mysql_select_db** — выбор базы данных (принимает параметром имя базы данных, с которой мы хотим установить соединение).

• **mysql_query** — универсальная функция отправки запросов к БД, с помощью нее можно сделать все (Запросы к базе данных представляют собой обычные строки).

Соединение с базой данных.

Пример:

```
//Устанавливаем доступы к базе данных:
$host = 'localhost'; //Имя хоста, на локальном компьютере это
localhost.
$user = 'root'; //Имя пользователя, по умолчанию это root.
$password = ''; //Пароль, по умолчанию пустой.
//Соединяемся с базой данных используя наши доступы:
mysql_connect($host, $user, $password);
//Выбираем базу данных (в нашем случае test):
$db_name = 'test';
mysql_select_db($db_name);
```

Посылаем запросы к базе данных

Пример:

```
mysql_query ("SELECT*FROM workers WHERE id>0");
//Мы можем формировать эту строку с помощью переменных:
$table = 'workers'; //Задаем имя таблицы в переменной.
mysql_query ("SELECT*FROM ".$table." WHERE id>0");
```

Принято правило, по которому команды SQL следует писать в верхнем регистре (то есть большими буквами), а все остальное — в нижнем. Это относится к командам SELECT, UPDATE, FROM, DELETE, WHERE и другим такого рода.

Чтобы вывести ошибки, следует пользоваться конструкцией `or die (mysql_error())`, которую необходимо добавлять к каждому запросу к БД.

Смотрите пример:

```
mysql_query($query) or die( mysql_error() );
```

Таким образом вы сразу будете получать сообщения об ошибках синтаксиса!

Выборка данных из таблицы

Команда SELECT позволяет нам достать несколько строк из нашей таблицы.

Синтаксис ее таков:

//ВЫБРАТЬ все_столбцы ИЗ таблицы_такой_то ГДЕ условие_такое_то.

```
"SELECT * FROM имя_таблицы WHERE условие по которому следует
выбрать строки";
```

После команды SELECT знак * означает, что выбраны все поля таблицы.

Пример:

```
$query = "SELECT * FROM workers WHERE id > 0"; //выбраны все поля у
таблицы workers, и строки, у которых id > 0
```

Другие примеры выборки:

```
$query = "SELECT * FROM workers WHERE id=2"; //строка с данными на Петю
$query = "SELECT * FROM workers WHERE id>2"; //все работники, кроме первого и второго
$query = "SELECT * FROM workers WHERE id != 2"; //все работники, кроме второго
$query = "SELECT * FROM workers WHERE age=23"; //работники с возрастом 23 года
$query = "SELECT * FROM workers WHERE salary=500"; //работники с зарплатой 500$
$query = "SELECT * FROM workers WHERE name='Дима'"; //работник с именем Дима
```

\$query = "SELECT * FROM workers WHERE salary=500 AND age=23"; //работники с зарплатой 500\$ и возрастом 23 года

Вставка данных в таблицу

С помощью команды **INSERT** можно добавлять новую информацию в таблицу. Результат данного действия нужно смотреть в PhpMyAdmin!

\$query = "INSERT INTO workers (name, age, salary) VALUES ('Гена', 30, 1000)"; //вставить в таблицу workers имя='Гена', возраст=30, зарплата=1000. (id не указывается)

Обновление данных

С помощью команды **UPDATE** можно обновлять какие-либо ячейки таблицы.

Пример: Изменим работнику с id=1 зарплату на цифру 1000\$ и возраст на 31 год:

\$query = "UPDATE workers SET age=31, salary=1000 WHERE id=1";

Если не задать условие WHERE id=1, то обновится вся таблица.

Другие примеры:

- Поставим Диме зарплату в 1200\$:
\$query = "UPDATE workers SET salary=1200 WHERE name='Дима'";
- Поставим работнику с id=4 возраст 31 год:
\$query = "UPDATE workers SET age=31 WHERE id=4";
- Поставим Васе возраст 30 лет и зарплату 300\$:
\$query = "UPDATE workers SET age=30, salary=300 WHERE name='Вася'";
- Поставим ВСЕМ работникам с возрастом 23 года - возраст 24 лет и зарплату 300\$:
\$query = "UPDATE workers SET age=24, salary=300 WHERE age=23";
- Для работников с id > 2 и id < 5 - зарплату 333\$:
\$query = "UPDATE workers SET salary=333 WHERE id>2 AND id<5";
- Поставим Васе имя Женя:
\$query = "UPDATE workers SET name='Женя' WHERE name='Вася'";

Удаление записей

С помощью команды **DELETE** можно удалять записи из таблицы. Ее синтаксис очень похож на команду **SELECT**.

"DELETE FROM имя_таблицы WHERE условие по которому следует удалять строки";

Давайте удалим запись с id=6:

//УДАЛИТЬ ИЗ workers ГДЕ id=6.

\$query = "DELETE FROM workers WHERE id=6";

Практическая часть

Задание 1. Проектирование базы данных

Имя базы данных “test”

По умолчанию у нас имя сервера **localhost** , имя пользователя **root** , пароль- **пустая строка**.

В базе данных содержится одна таблица **workers**

i d	name	ag e	salary
1	Дима	23	400
2	Петя	25	500
3	Вася	23	500
4	Коля	30	1000
5	Иван	27	500
6	Кирил л	28	1000

В ней нужно создать 4 поля (столбца):

- **id** – тип `integer`, не забудьте поставить ему галочку `AUTO_INCREMENT` (чтобы в этом столбце номера проставлялись автоматически).
- **name** (англ. имя) – тип `varchar`, размером в 256 знаков.
- **age** (англ. возраст) — тип `integer`.
- **salary** (англ. зарплата) — тип `integer`.

Задание 2. Создание Базы данных

1. Создание пользователя

Прежде, чем начинать работу с БД нужно задать имя пользователя и пароль:

1. Перейти на главную страницу программы (нажать кнопку с домиком) и в верхнем меню выбрать пункт «Пользователи».
2. Нажать кнопку «Добавить пользователя»
3. Заполнить нужные окна формы.
В проекте указаны следующие параметры: Пользователь: **root**. Пароль: "" . Хост выбираем «Локальный»
4. Установка прав пользователя (привилегии): в этом же окне в пункте «Глобальные привилегии» нажимаем ссылку «Отметить все».
5. Нажимаем кнопку «Добавить пользователя» и он появляется в списке пользователей

2. Создание базы данных

1. Зайти на главную страницу **PHPMyAdmin**. Используйте настройку кодировки **utf8_general_ci**.

2. Вкладка «Базы данных» форма «Создать базу данных». Задайте имя для базы данных, выберите кодировку **utf8_general_ci** и нажмите на кнопку "Создать".
3. В левом окне заходим в созданную базу данных (двойной щелчок по названию БД)
4. В окнах формы вводим имя таблицы **workers** и количество полей (столбцов) – **4** нажимаем ОК.
5. Открывается окно структуры таблицы и заполняем данные:

id – тип integer, не забудьте поставить ему галочку AUTO_INCREMENT (чтобы в этом столбце номера проставлялись автоматически).

name (англ. имя) – тип varchar, размером в 256 знаков.

age (англ. возраст) — тип integer.

6. На следующем этапе заполняем таблицу данными из проекта. Для этого нужно в левом окне войти в таблицу **workers** и в верхнем меню выбрать пункт «Вставить» и в открывшиеся окна вставить данные из проекта таблицы. Данные вставляются для каждой строки отдельно. В окне **id** нумерация будет проставляться автоматически.

3. Редактирование Базы данных и таблицы

При редактировании можно изменять любые параметры Базы данных и таблиц. Для редактирования базы данных нужно в левом окне войти в выбранную базу данных и в верхнем меню выбрать пункт «Операции». В открывшемся новом окне можно внести нужные изменения.

Для редактирования таблицы нужно в левом окне войти в выбранную таблицу и в верхнем меню выбрать пункт «Операции»:

Для редактирования отдельных строк нужно войти в таблицу и в нужном поле нажать кнопку «Изменить»

4. Просмотр созданной таблицы

Для просмотра созданной таблицы необходимо:

- Войти на страницу выбранной таблицы (левое окно).
- В верхнем меню выбрать пункт «Экспорт»
- В окне «Формат» выбрать пункт «Microsoft Word 2000» и нажать ОК
- В открытом окне редактора Word можно добавить границы в таблицу и увидеть результат, который должен совпадать с проектом.

Задание 3. Использование функций для работы с базой данных

1. Выборка данных. Функция SELECT.

Пример:

```

$host = 'localhost'; //Имя хоста, на локальном компьютере это localhost.
$user = 'root'; //Имя пользователя, по умолчанию это root.
$password = ''; //Пароль, по умолчанию пустой.
//Соединяемся с базой данных используя наши доступы:
mysql_connect($host, $user, $password) or die(mysql_error());
//Выбираем базу данных (в нашем случае test):
$db_name = 'test'; //Имя базы данных.
mysql_select_db($db_name) or die(mysql_error());
//Устанавливаем кодировку (не обязательно, но поможет избежать проблем):
mysql_query("SET NAMES 'utf8'");
//ВЫБРАТЬ все столбцы ИЗ workers ГДЕ id больше нуля (т.е. все!)
$query = "SELECT * FROM workers WHERE id > 0";
//Делаем запрос к БД, результат запроса пишем в $result:
$result = mysql_query($query) or die( mysql_error() );
//преобразуем то, что отдала нам база в нормальный массив PHP $data:
for ($data = array(); $row = mysql_fetch_assoc($result); $data[] = $row);
//массив результата лежит в $data, выведем его на экран:
var_dump($data);

```

Var_dump покажет нам, что в массиве \$data лежат все строки нашей таблицы.

1. Выбрать работника с id = 3. Узнать его имя.
 2. Выбрать работников с id = 2 и id = 3.
 3. Выбрать работника с зарплатой 1000\$.
 4. Выбрать работников в возрасте 23 года.
- 20

Трепачев Д.П. Простой учебник SQL+PHP, <http://phphtml.net>, <http://sitetech.by>

5. Выбрать работников с зарплатой более 400\$.
6. Выбрать работников в возрасте от 25 (не включительно) до 28 лет (включительно).
7. Выбрать работников с зарплатой не равной 500\$.
8. Выбрать работника Петю.
9. Выбрать работников Петю и Васю.
10. Выбрать всех, кроме работника Петя.
11. Выбрать всех работников в возрасте 27 лет или с зарплатой 1000\$.
12. Выбрать всех работников в возрасте от 23 лет до 27 лет или с зарплатой 1000\$.
13. Выбрать всех работников в возрасте от 23 лет до 27 лет или с зарплатой от 400\$ до 1000\$.

2. Вставка данных. Функция INSERT

Добавьте нового работника Никиту, 26 лет, зарплата 300\$. Воспользуйтесь первым синтаксисом.

2. Добавьте нового работника Светлану с зарплатой 1200\$. Воспользуйтесь вторым синтаксисом.

3. Обновление данных. Функция UPDATE

1. Поставьте Жене зарплату в 200\$.
2. Работнику с id=4 поставьте возраст 35 лет.
3. Всем, у кого зарплата 333\$ сделайте ее 700\$.
4. Работникам с id больше 2 и меньше 4 включительно поставьте возраст 23.
5. Поменяйте Женю на Васю и прибавьте ему зарплату до 900\$.

4. Удаление записей. Функция

Удалить работника с id=7.

2. Удалить Колю.

3. Удалить всех работников, у которых возраст 23 года.

Задание 1. Создание базы данных с помощью клиента phpMyAdmin:

1. Создание базы данных **"firma"**.
2. Создание таблицы **"firstbd"**.

№	Имя	Фамилия	Должность
1	Алексей	Алексеев	Директор
2	Иван	Иванов	Заместитель
3	Александр	Александров	Программист

3. Создание пользователя: имя – **alex**, хост – **localhost**, пароль – **12345**.

Соединение с базой данных:

1. Соединение с сервером MySQL и получение идентификатора (функция *mysql_connect*)
`$db = mysql_connect ("Сервер", "пользователь", "пароль");`
2. Выбор базы данных, с которой нужно работать
`mysql_select_db ("имя базы", идентификатор);`

Операции с базой данных:

1. **select** – выборка данных;
2. **insert** – вставка данных;
3. **update** – редактирование данных;
4. **delete** – удаление данных.

Порядок выборки данных из базы данных

1. Запрос к базе данных (функция *mysql_query()*):
`$result = mesql_query (" SELECT * FROM имя таблицы, $db);`
2. Создание массива \$myrow из выбранной таблицы (первая строка из таблицы) (функция *mysql_fetch_array()*):
`$myrow = mysql_fetch_array($result);`
3. Вывод данных на экран (*echo* выводит только один элемент массива, расположенный в данном индексе):
`echo $myrow["индекс массива"];`
4. Вывод всего массива из данной таблицы с помощью функции *printf()*:
`printf ("Компудник № - %S
%S
%S

", $myrow['id'], $myrow['name'], $myrow['lastname'], $myrow['dol']);`
Данная функция работает внутри цикла `do {.....} while`.

Задание 2. Создание и вывод на экран базы “site_name” с полями (№, work_place, last_name, name, my_site)