

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

С.В. Одиночкина

**Web-программирование: PHP
практикум**



Санкт-Петербург

УДК 004.655, 004.657, 004.62

С.В.Одиночкина

Web-программирование PHP - СПб: НИУ ИТМО, 2012. – 79 с.

В пособии излагаются методические рекомендации к выполнению лабораторных работ по дисциплине «Web-программирование PHP-технологии».

Предназначено для студентов, обучающихся по всем профилям подготовки бакалавров направления: 210700 Инфокоммуникационные технологии и системы связи.

Рекомендовано к печати Ученым советом факультета Инфокоммуникационных технологий, протокол №4 от 13 декабря 2011г.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2012

© С.В.Одиночкина, 2012.

Лабораторная работа №3: Создание базы данных MySQL

В ходе выполнения данной лабораторной работы необходимо создать в MySQL новую базу данных с названием «**MySiteDB**» и добавить в нее две таблицы: **notes** и **comments**. **Notes** содержит заметки блога; **comments** – комментарии к этим заметкам. Схема данных (рис.3.1):

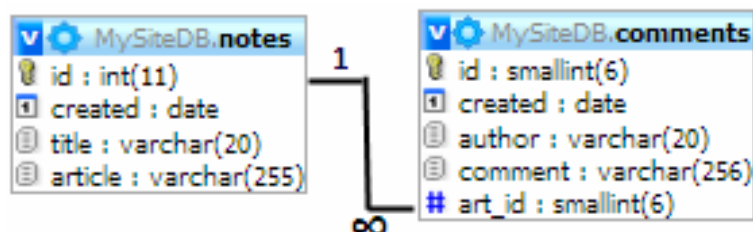


Рис.3.1. Схема базы данных “MySiteDB”

В Приложении 4 представлена информация об основных понятиях, необходимых для работы с базой данных.

Упражнение 1: Создание БД «MySiteDB»

В этом упражнении реализуется запрос на создание новой базы данных.

1. Создайте новый php документ, который будет называться **create_db.php**.
2. Создайте соединение с сервером localhost. Имя сервера **localhost**, пользователь **root**, пароля нет.
3. Создайте базу данных:
 - 3.1. Сформируйте запрос на создание базы **MySiteDB** с использованием SQL;
 - 3.2. Реализуйте запрос на создание БД с помощью функции `mysqli_query()`.
4. Сохранить документ, выполнить запрос.
5. С помощью утилиты **PhpMyAdmin** убедитесь, что создана новая база данных. Для этого запустите утилиту: <http://localhost/tools/phpmyadmin> (или <http://localhost> и выберите PhpMyAdmin из списка утилит).
6. Вторично выполните запрос, чтобы убедиться, что соединение есть, а база не создается (т.к. она была уже создана ранее, в ходе предыдущего выполнения скрипта).
7. Желательно добавить цикл `if` для обнаружения неполадок в работе.

Вариант реализации создания БД MySiteDB

```
<?php
//Создать соединение с сервером
$link = mysqli_connect ("localhost", "root", "");
if ($link) {
    echo "Соединение с сервером установлено", "<br>";
} else {
    echo "Нет соединения с сервером";
}

//Создать БД MySiteDB
//Сначала формирование запроса на создание
$db = "MySiteDB";
$query = "CREATE DATABASE $db";

//Затем реализация запроса на создание. Важна последовательность
аргументов функции: соединение с сервером, SQL-запрос.
$create_db = mysqli_query($link, $query);
if ($create_db) {
    echo "База данных $db успешно создана";
} else {
    echo "База не создана";
}

?>
```

Упражнение 2: Создание пользователя admin

В этом упражнении Необходимо создать нового пользователя базы данных с именем **admin** и паролем **admin** с правами администратора. Пользователей можно добавлять двумя способами:

- при помощи SQL-запроса GRANT
- в таблице назначения привилегий MySQL (Privileges) с помощью утилиты PhpMyAdmin.

Выберите один из двух приведенных далее способов.

Способ 1: создание нового пользователя с помощью SQL-запроса GRANT

1. Создайте новый php-документ, который будет называться `create_user.php`;
2. Создайте соединение с сервером;
3. Сформируйте SQL-запрос на создание нового пользователя базы данных:

```
$query = "GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost'
IDENTIFIED BY 'admin'
WITH GRANT OPTION";
```

//. * - глобальный уровень привилегий, применяется ко всем базам на сервере.*

4. Реализуйте запрос.

- Проверка создания пользователя. С помощью утилиты PhpMyAdmin убедитесь, что создан новый пользователь. Для этого запустите утилиту PhpMyAdmin и перейдите на вкладку Privileges. Изучите список пользователей.

Способ 2: создание нового пользователя с помощью утилиты PhpMyAdmin

1. Запустите утилиту PhpMyAdmin и перейдите на вкладку Privileges. Нажмите кнопку «Add a new user».
2. Введите имя пользователя (***admin***), имя сервера (***localhost***), пароль с подтверждением (***admin***). Предоставьте новому пользователю все права (***global privileges – Check All***).
3. Убедитесь, что новый пользователь создан корректно.
4. Все дальнейшие действия с базой данных будут проводиться под пользователем **admin** с паролем **admin** и соответствующими правами, если иное не указано в задании.

Упражнение 3: Создание таблицы notes

В данном упражнении будет продемонстрирован один из способов создания таблиц в ранее созданной базе данных на примере создания таблицы **notes**. Таблица **notes** содержит заметки автора блога. Данная таблица будет создана средствами PHP. Информацию о полях таблицы см. в *Приложении 3*.

1. Создайте новый php-документ, который будет называться **create_tbl.php**;
2. Создайте соединение с сервером уже под созданным ранее пользователем **admin** с паролем **admin**.
3. Подключитесь к базе данных MySiteDB.
4. Сформируйте запрос на создание таблицы **notes** с полями, указанными в Приложении 3.

//Формирование запроса

```
$query = "CREATE TABLE notes
        (id INT NOT NULL AUTO_INCREMENT,
        PRIMARY KEY (id),
        created DATE,
        title VARCHAR (20),
        article VARCHAR (255))";
```

5. Реализуйте запрос на создание таблицы.
6. С помощью утилиты PhpMyAdmin убедитесь, что создана новая таблица. Для этого запустите утилиту, перейдите к базе данных MySiteDB и просмотрите ее структуру. В ней должна появиться соответствующая таблица.

Вариант реализации создания таблицы notes

<?php

//Соединение с сервером

```
$link = mysqli_connect ('localhost', 'admin', 'admin');
```

//Выбор БД

```
$db = "mySiteDB";
```

```
$select = mysqli_select_db($link, $db);
```

```
if ($select){
    echo "База успешно выбрана", "<br>";
} else {
    echo "База не выбрана";
}
```

//Создание таблицы

//Формирование запроса

```
$query = "CREATE TABLE notes
        (i  d INT NOT NULL AUTO_INCREMENT,
        PRIMARY KEY (id),
        cr  eated DATE,
        title VARCHAR (20),
        ar  ticle VARCHAR (255))";
```

```
//Реализация запроса
$create_tbl = mysqli_query ($link, $query);
if ($create_tbl){
    echo "Таблица успешно создана", "<br>";
} else {
    echo "Таблица не создана";
}
?>
```

Упражнение 4: Создание таблицы comments

В данном упражнении будет продемонстрирован другой способ создания таблиц в ранее созданной базе данных на примере создания таблицы **comments**. Таблица **comments** содержит комментарии пользователей к

заметкам автора блога. Таблица будет создана с помощью утилиты PhpMyAdmin. Информацию о полях таблицы см. в *Приложении 3*.

1. Запустите браузер.
2. Запустите утилиту *phpMyAdmin*. В главном окне *PHPMysqlAdmin* выберите БД MySiteDB.
3. В поле “Create new table”, присвойте имя таблице – comments; количество полей - 5, нажмите кнопку «Go» (рис. 3.2).

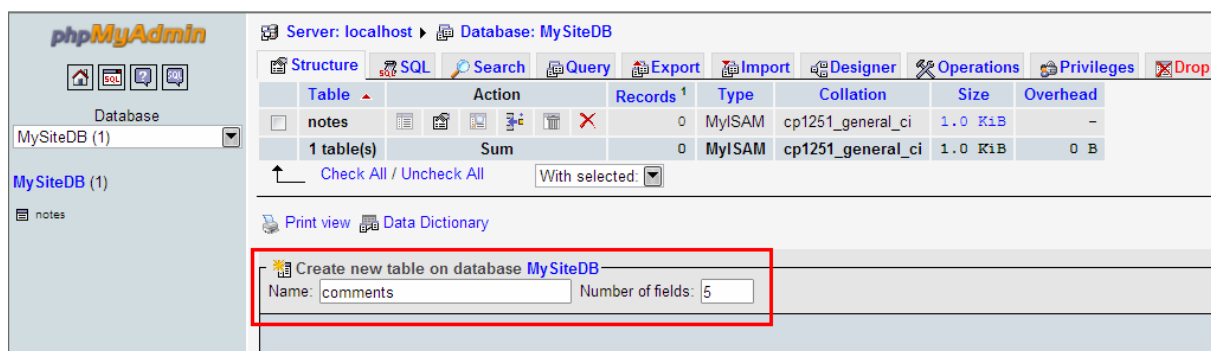


Рис. 3.2. Создание новой таблицы с помощью утилиты *phpMyAdmin*.

4. Создание полей таблицы comments:
 - 4.1. В открывшемся окне заполните необходимые поля таблицы (рис. 3.3) и нажмите кнопку «Save».
 - 4.2. Для поля **id** добавьте следующие атрибуты: обозначьте автоинкремент **A_I** и первичный ключ **PRIMARY** в поле со списком **INDEX**.

Server: localhost Database: MySiteDB Table: comments

Field	Type	Length/Values ¹	Default ²
id	SMALLINT		None
created	DATE		None
author	VARCHAR	20	None
comment	VARCHAR	256	None
art_id	SMALLINT		None

Рис. 3.3. Заполнение полей таблицы

5. Полученный результат должен выглядеть следующим образом (рис. 3.4):




































	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	id	smallint(6)			No	None	auto_increment	      
<input type="checkbox"/>	created	date			No	None		      
<input type="checkbox"/>	author	varchar(20)	cp1251_general_ci		No	None		      
<input type="checkbox"/>	comment	varchar(256)	cp1251_general_ci		No	None		      
<input type="checkbox"/>	art_id	smallint(6)			No	None		      

Рис. 3.4. Результат создания таблицы

Упражнение 5: Создание межтабличных связей

В данном упражнении необходимо создать связи между таблицами для поддержания целостности данных web-приложения.

1. Для организации межтабличных связей выберите БД MySiteDB, вкладку Designer. Откроется окно схемы данных.
2. С помощью инструментов окна Designer создайте связь «один ко многим» (рис. 3.5).

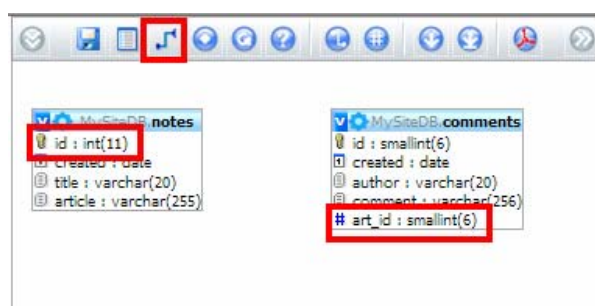


Рис. 3.5 Поле окна инструментов Designer

3. Введите в созданные таблицы несколько записей – для проверки их работы и для использования на будущих серверных страницах сайта. Для этого выберите нужную таблицу и нажмите кнопку **Insert** в группе **Action**. После заполнения соответствующих полей таблицы нажмите кнопку **Go**.

⇒ Помните, что поля **id** в таблицах заполнять не надо – они заполняются автоматически.

⇒ Поле **art_id** таблицы **comments** должно быть привязано к полю **id** таблицы **notes**, т.к. комментарии создаются только в привязке к конкретной заметке. MySQL автоматически в поле **art_id** таблицы **comments** подставляет выпадающий список **id** уже созданных заметок, вам необходимо лишь выбрать **id** заметки из выпадающего списка.

Упражнение 6: Регистрация базы данных в Adobe Dreamweaver для подключения к сайту

Регистрация включает задание имени и пароля пользователя, от имени которого ведется работа с базой (в нашем случае admin), а также адрес сервера данных (localhost) и имя базы (MySiteDB).

1. Запустите сервер и базу данных MySQL.
2. В Adobe Dreamweaver в меню **Базы данных** нажмите кнопку «плюс (+)». Появится единственный пункт – **Подключение MySQL**.
3. Далее появится диалоговое окно, которое необходимо заполнить (с.м. рис.3.6). Последний раздел «База данных» можно заполнить как самостоятельно, так и нажав **Выбрать**. Протестируйте соединение. Затем нажмите **ОК**.

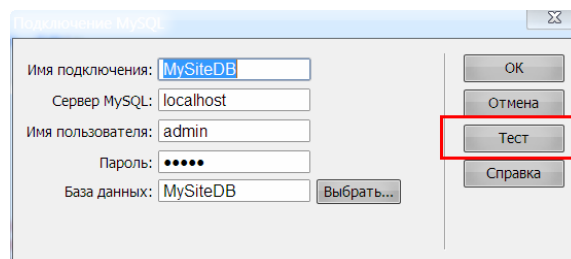


Рис. 3.6. Заполнение полей окна подключения базы данных к проекту

4. После подключения в меню **Базы данных** должна отобразиться подключенная нами БД MySiteDB с двумя таблицами (рис. 3.7).

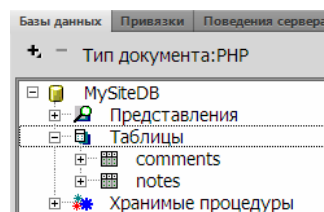


Рис. 3.7. Отображение подключенной к проекту базы данных

5. Убедитесь, что в меню **Файлы** появилась новая папка **Connections** и в ней файл **MySiteDB.php** (название файла совпадает с именем соединения).

Упражнение 7: Файл подключения базы данных

В ходе выполнения данного упражнения необходимо внести изменения в код автоматически созданного файла подключения базы данных для настройки корректной работы подключенной базы данных с кириллицей.

1. Откройте файл MySiteDB.php и внесите в него следующие изменения:
 - 1.1. Измените наименования переменных для удобства дальнейшей работы;
 - 1.2. Измените функцию *mysql_pconnect()* на *mysqli_connect()*;
 - 1.3. Внесите дополнения для корректной кодировки символов в базе данных:

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$localhost = "localhost";
$db = "MySiteDB";
$user = "admin";
$password = "admin";

$link = mysqli_connect($localhost, $user, $password) or
trigger_error(mysql_error(),E_USER_ERROR);
//trigger_error выводит на страницу сообщение об ошибке. Первый параметр
- сообщение об ошибке
//в строковом виде, в данном случае возвращается функция mysql_error(),
второй - числовой код //ошибки(почти всегда используется значение
константы E_USER_ERROR, равное 256)

//Следующие строки необходимы для того, чтобы MySQL воспринимал
кириллицу.
//Параметры функции mysqli_query(): идентификатор соединения с сервером
и запрос SQL

mysqli_query($link, "SET NAMES cp1251;") or die(mysql_error());
mysqli_query($link, "SET CHARACTER SET cp1251;") or die(mysql_error());
?>
```