

## АДМИНИСТРИРОВАНИЕ СЕРВЕРА БАЗ ДАННЫХ MYSQL. РАЗРАБОТКА ПРОСТЕЙШЕГО КЛИЕНТСКОГО ПРИЛОЖЕНИЯ

### *Установка базового пакета Denwer.*

Базовый пакет содержит большинство необходимых программ и утилит: Apache, SSL, SSI, PHP5 с поддержкой GD, MySQL5 с поддержкой транзакций, систему управления виртуальными хостами, систему управления запуском и завершением всех компонентов Денвера, phpMyAdmin — система управления MySQL через Web-интерфейс, эмулятор sendmail и SMTP-сервера. Пакет предназначен для быстрой настройки компонентов, необходимых для разработки WEB-приложений с использованием СУБД MySQL и скриптов PHP.

Для установки пакета воспользуйтесь инсталлятором, который можно загрузить на сайте [www.denwer.ru](http://www.denwer.ru). На этом же сайте находится подробная инструкция по установке системы.

### *Создание новой базы данных*

Запустить WEB-браузер. Инициализировать работу приложения phpMyAdmin. Для этого в строке адреса ввести ссылку <http://localhost/Tools/phpmyadmin/>

В разделе «Actions/MySQLlocalhost/Createnewdatabase» ввести произвольное имя базы данных, например lab1, нажать кнопку «Create» (рисунок 1.1).

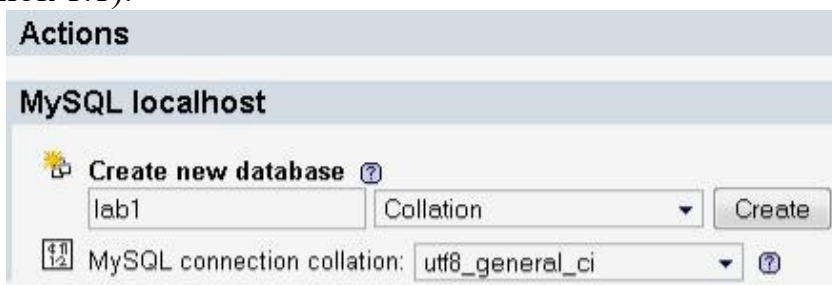


Рисунок 1.1 – Создание новой базы данных

Обратите внимание на SQL-скрипт, расположенный вверху страницы после создания базы данных (рисунок 1.2).

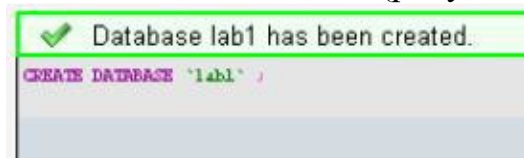


Рисунок 1.2 – SQL-скрипт для создания базы данных.

### *Создание отношения (таблицы)*

Так как БД является моделью реального мира перед созданием таблицы необходимо описать предметную область, которую она представляет. Рассмотрим простейшую модель: телефонный справочник. Данные этой

модели определяются следующими атрибутами: уникальный номер абонента (он должен создаваться автоматически), фамилия, имя, отчество, дата рождения (рисунок 1.3).

PHONELIB	
ID	INTEGER
FIO	VARCHAR(50)
BDATE	DATE

Рисунок 1.3 – Атрибуты отношения «Телефонный справочник»

Для создания таблицы в среде phpMyAdmin воспользуемся интерфейсом CreateNewTable (рисунок 1.4).



Рисунок 1.4 – Интерфейс создания таблицы

Описываем в интерфейсе имена атрибутов и их типы (рисунок 1.5).

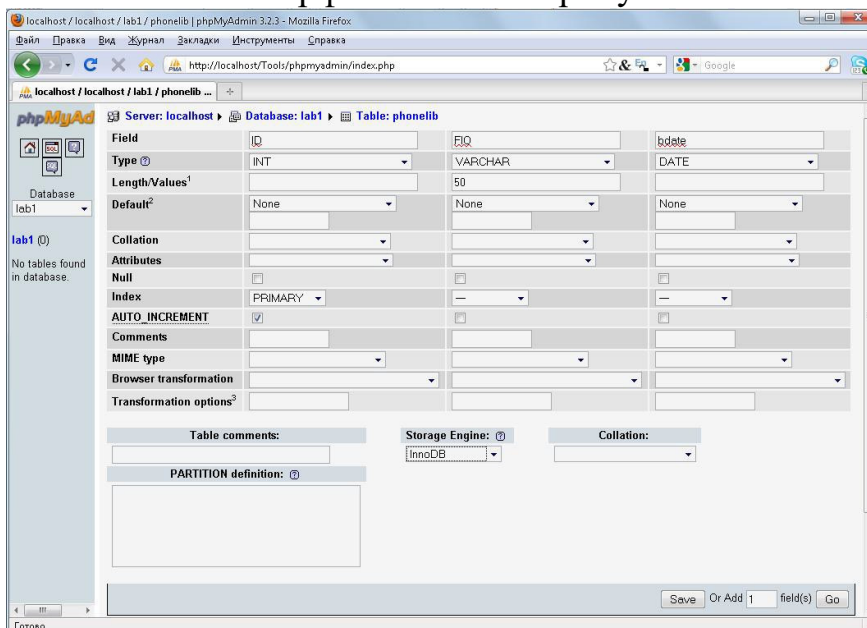


Рисунок 1.5 – Описание атрибутов

Обратите внимание, что поле для поля ID установлен признак AUTO\_INCREMENT, для того чтобы при добавлении новой записи (картежа) значение поля генерировалось автоматически. На будущее, установим для этого атрибута признак PRIMARY в поле index. Атрибут FIO (фамилия, имя, отчество) является текстовым, поэтому для него указываем длину 50 (в символах).

Для создания таблицы после описания атрибутов нажимаем кнопку «Save».

Среда phpMyAdmin автоматически генерирует SQL-скрипт для создания новой таблицы:

```
CREATE TABLE `lab1`.`phonelib` (
  `ID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `FIO` VARCHAR( 50 ) NOT NULL ,
  `bdate` DATE NOT NULL
) ENGINE = InnoDB;
```

Синтаксис операции CREATETABLE в MySQL, в целом, соответствует стандарту SQL-92, однако имеет ряд особенностей. В частности, фрагмент фразы ENGINE = InnoDB определяет тип «движка» для работы с данными, каждый из которых имеет свои характерные особенности.

**MyISAM.** Движок по умолчанию. Не поддерживает транзакций, средняя надежность хранения данных. Превосходная производительность чтения данных (через SELECT). Блокирует всю таблицу при записи в неё данных, отчего маленькая производительность при частых записях.

**InnoDB.** Транзакционный тип движка, применяемый при интенсивных операциях записи, благодаря возможности блокировки уровня строк таблицы. Великолепная восстанавливаемость и высокая надежность хранения данных. Поддерживает внешние ключи. Именно этот движок рекомендуется использовать при выполнении лабораторных работ.

### ***Ввод данных в таблицу***

Для ввода данных в таблицу выбираем операцию «Insert» (рисунок 1.6).

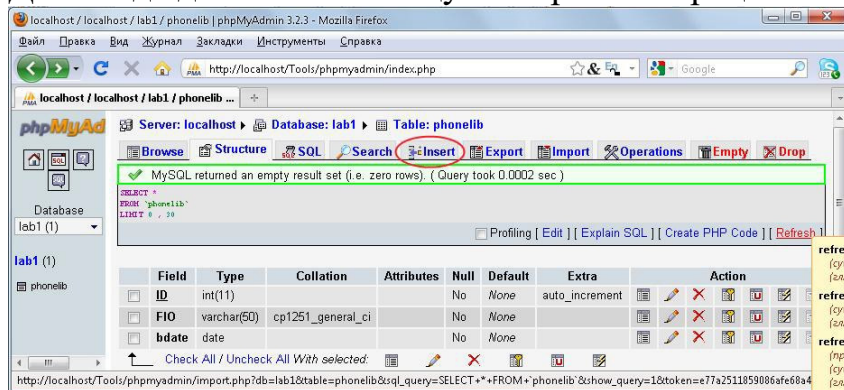


Рисунок 1.6 – Переход к вводу данных

Заполняем колонку «VALUE» в интерфейсе ввода данных и нажимаем на кнопку «Go» (рисунок 1.7). Обратите внимание, что поле ID оставляем пустым.

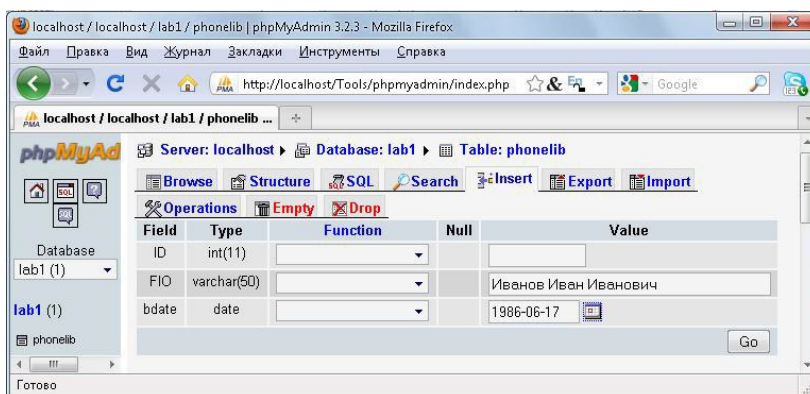


Рисунок 1.7 – Ввод данных в таблицу

SQL-скрипт добавления записи имеет вид: `INSERT INTO 'lab1`.`phonelib` ('ID', 'FIO', 'bdate') VALUES (NULL, 'Иванов Иван Иванович', '1986-06-17')`. Переходим на вкладку «Browse», для того чтобы убедиться в правильности заполнения данных. При этом выполняется SQL-скрипт выборки данных из таблицы `SELECT * FROM 'phonelib'` (рисунок 1.8).

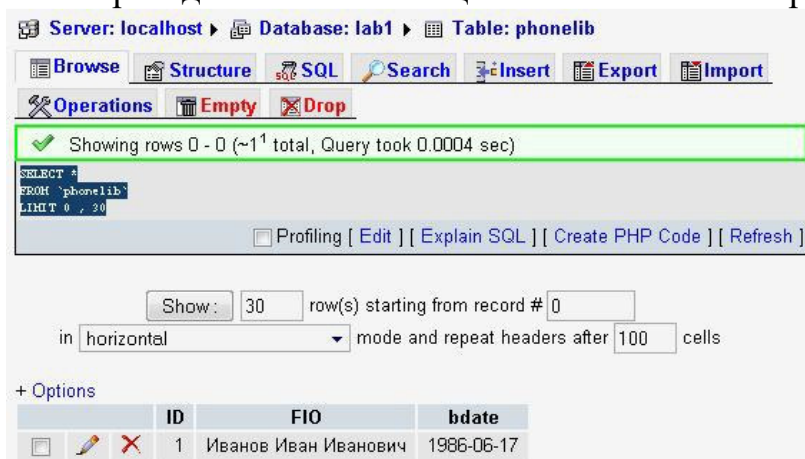


Рисунок 1.8 – Выборка данных из таблицы PhoneLib

### ***Администрирование базы данных. Создание новых пользователей, паролей и привилегий***

Пользователь имеет определенные привилегии (может редактировать, удалять, создавать новые таблицы и позиции в базе данных) и авторизуется по заданному логину и паролю.

При покупке хостинга у большинства хостинг-провайдеров для вас автоматически создается база данных и аккаунт пользователя с логином и паролем, имеющий все необходимые привилегии. На нормальном хостинге есть точно такой же phpMyAdmin посредством которого вы сможете управлять таблицами и их содержимым, а также импортировать данные из локальной базы данных в базу данных хостинга. Логин и пароль для доступа к базе данных, а также ссылка на phpMyAdmin высылается хостинг-провайдером на ваш e-mail.

На локальном же компьютере нам придется самим создать пользователя. Для этого выбираем базу данных lab1 и вкладку Priveleges (рисунок 1.9).

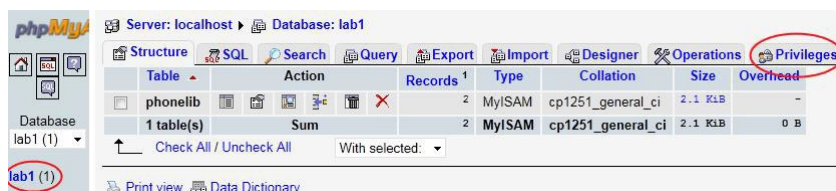


Рисунок 1.9 – Редактирование привилегий

Переходим по ссылке Add a new user и в интерфейсе администратора создаем нового пользователя и пароль (рисунок 1.10). В нашем случае имя пользователя будет совпадать с паролем (student).

Рисунок 1.10 – Создание нового пользователя, пароля и привилегий

В расположенном ниже интерфейсе определяем права пользователя при работе с базой данных. Как минимум, для обеспечения функциональности необходимо разрешить добавлять записи в таблицу и выбирать данные из нее (операции Insert и Select). В нашем случае разрешим также обновлять и удалять записи (Update и Delete).

### ***Разработка клиентского приложения***

Клиент — это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.

Программа, являющаяся клиентом, взаимодействует с сервером, используя определённый протокол. Она может запрашивать с сервера какие-либо данные, манипулировать данными непосредственно на сервере, запускать на сервере новые процессы и т. п. Полученные от сервера данные клиентская программа может предоставлять пользователю или использовать как-либо иначе, в зависимости от назначения программы. Программа-клиент и программа-сервер могут работать как на одном и том же компьютере, так и на разных. Во втором случае для обмена информацией между ними используется сетевое соединение.

Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер и сервер баз данных. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами. Для создания веб-приложений на стороне сервера используются разнообразные технологии и любые языки программирования, способные

осуществлять вывод в стандартную консоль.

В данном курсе приложение-клиент будет разрабатываться средствами языков HTML, JAVA-скрипт, PHP.

Рассмотрим структуру информации после установки пакета разработчика «Denwer».

Предположим, что папка «C:\WebServers», в которую был установлен Denwer проектируется на виртуальный диск S. Тогда в папе «S:\usr\local\mysql-5.1\data\lab1\» будет находится разработанная нами база данных. Для создания клиентского приложения необходимо войти в папку «S:\home\localhost\www\» и создать в ней каталог Lab1. Создаем файл «S:\home\localhost\www\Lab1\index.html», в котором будет размещен HTML-скрипт титульной страницы проекта:

```
<HTML>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<meta name="Author" content="СергеевГеоргийГеоргиевич">
<META NAME="ROBOTS" CONTENT="ALL">
<METANAME="Keywords" CONTENT="лабораторная работа, MySQL, соединение с базой данных">
<META NAME="Description" CONTENT="Лабораторная работа '1. Соединение с базой данных'">
</head>
<title>Лабораторная работа '1. Соединение с базой данных</title>
<BODY>
<A HREF="phlib.php">Перечень абонентов</A>
</BODY>
</HTML>
```

Если в обозревателе теперь ввести адрес «http://localhost/lab1/», увидим ссылку «Перечень абонентов».

Задачей приложения-клиента в данной лабораторной работе является вывод состояния отношения в виде таблицы. Для этого необходимо выполнить следующие действия:

- 1) установить соединение с базой данных;
- 2) выполнить выборку данных;
- 3) отобразить данные в виде таблицы.

Для общения с MySQL из PHP понадобятся следующие функции:

1) *создать соединение с MySQL:*

***intmysql\_connect (string hostname, string username, string password).***

*Параметры:*

**Hostname** – имя хоста, на котором находится база данных,

**Username** – имя пользователя,

**Password** – пароль пользователя.

Функция возвращает параметр типа int, который больше 0, если соединение прошло успешно, и равен 0 в противном случае.

2) *выбрать базу данных для работы:*

***intmysql\_select\_db(string database\_name, intlink\_identifier).***

*Параметры:*

**Database\_name** – имя базы данных;

**link\_identifier** – ID соединения, которое получено в функции *mysql\_connect*. (параметр необязательный, если он не указывается, то



используется ID от последнего вызова `mysql_connect()`.  
Функция возвращает значение `true` или `false`.

3) *запрос к базе данных:*

***intmysql\_query(string query, intlink\_identifier).***

*Параметры:*

***Query*** – строка, содержащая запрос,

***link\_identifier*** – см. предыдущую функцию.

Функция возвращает ID результата или 0, если произошла ошибка.

4) *В цикле выбираем записи, полученные в результате запроса:*

***arraymysql\_fetch\_array(resource result [, intresult\_type]).***

*Параметры:*

***result*** – указатель на ресурс результатов выборки, второй (необязательный) – флаг, отвечающий за содержимое возвращаемого функцией ассоциативного массива.

Функция возвращает массив, соответствующий извлечённому ряду, или `FALSE`, если рядов больше нет. Функция `mysql_fetch_array()` – это расширенная версия `mysql_fetch_row()`. Помимо сохранения данных в числовых индексах результирующего массива, она также хранит данные в ассоциативных индексах, используя в качестве ключей имена полей.

5) *закрывать соединение*

***intmysql\_close (intlink\_identifier).***

*Параметры:*

***link\_identifier*** – см. выше.

Функция возвращает значение `true` или `false`.

Скрипт для выдачи списка абонентов в виде таблицы, расположенный в файле «S:\home\localhost\www\Lab1\phlib.php» будет иметь вид:

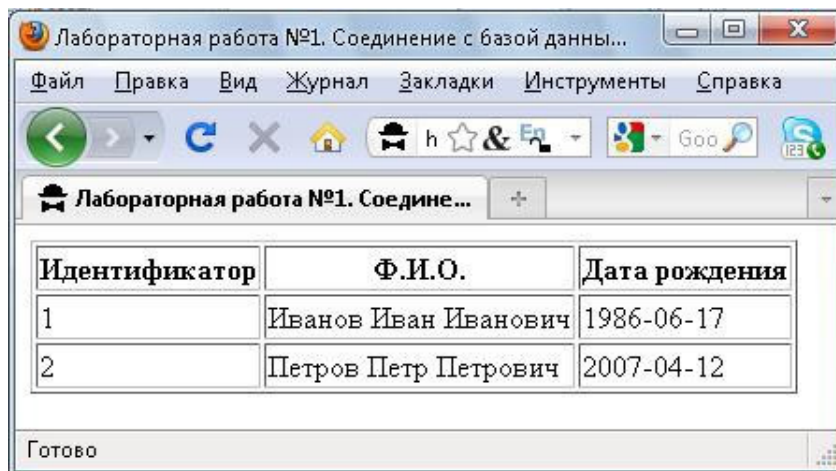
```
<HTML>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<meta name="Author" content="Сепреев Георгий Георгиевич">
<META NAME="ROBOTS" CONTENT="ALL">
<METANAME="Keywords" CONTENT="лабораторная работа, MySQL, соединение с базой данных">
<META NAME="Description" CONTENT="Лабораторная работа №1. Соединение с базой данных">
</head>
<title>Лабораторная работа №1. Соединение с базой данных</title>
<BODY>
<?
/* Переменные для соединения с базой данных */
$hostname = "localhost";
$username = "student";
$password = "student";
$dbName = "lab1";
/* Таблица MySQL, в которой хранятся данные */
$userstable = "phonelib";
/* создать соединение */
mysql_connect($hostname,$username,$password) OR DIE("Не могу создать соединение");
/* выбрать базу данных. Если произойдет ошибка - вывести ее */
mysql_select_db($dbName) or die(mysql_error());
/* составить запрос для выборки информации */
$query = "SELECT * FROM $userstable";
/* Выполнить запрос. */
```

```

$result=mysql_query($query) or die(mysql_error());
echo "<TABLE BORDER=1>";
echo"<TR><TH>Идентификатор</TH><TH>Ф.И.О.</TH><TH>Датарождения</TH></TR>";
/* Выбрать очередную запись из таблицы. */
while($row=mysql_fetch_array($result))// берем результаты из каждой строки
{
echo;
/* Вывести ее в виде HTML*/
echo "<tr><td>".$row['ID']. "</td><td>".$row['FIO']. "</td><td>".$row['bdate']. "</td></tr>";
};
echo "</TABLE>";
/* Закрыть соединение */
mysql_close();
?>
</BODY>
</HTML>

```

Результат работы скрипта приведен на рисунке 1.11.



Идентификатор	Ф.И.О.	Дата рождения
1	Иванов Иван Иванович	1986-06-17
2	Петров Петр Петрович	2007-04-12

Готово

Рисунок 1.11 -Результат выполнения скрипта выборки из базы данных