

## Практическая работа №7 «PHP: работа с СУБД MySQL»

**Цель работы:** познакомиться с возможностями и приёмами работы с СУБД MySQL.

### ЗАДАНИЯ

#### А. Подготовка к работе:

1. Запустите менеджер виртуальных машин VMware Player на **реальном** компьютере.
2. В программе VMware Player запустите образ **виртуального** компьютера **WWW** с предустановленным ПО: Apache (HTTP-сервер), PHP (интерпретатор) и MySQL (СУБД).
3. В **адресной строке браузера** с реального компьютера откройте поочередно ссылки <http://www/>, <http://www/info.php> и <http://www/sqltest.php>. Убедитесь в работоспособности перечисленных выше программных компонентов виртуального веб-сервера.
4. В **проводнике на реальном** компьютере введите адрес [\\www](http://www), укажите по запросу имя пользователя – Администратор; пароль – 1234. Отобразится список общих папок веб-сервера, в т.ч. папка www – **корневая папка сайта** (локальный путь – C:\wamp\www).

Если проводник сообщает, что нет доступа к ресурсу, то на реальном компьютере (для ОС Windows Vista / 7 / 8.x / 10) откройте «Панель управления» → «Сеть и Интернет» → «Центр управления сетями и общим доступом» → «Изменить дополнительные параметры общего доступа». В нужном профиле выберите «Включить сетевое обнаружение» и «Включить общий доступ, чтобы сетевые пользователи могли читать и записывать файлы в общих папках». Сохраните изменения и повторите попытку.

Возможной причиной блокирования также может быть **пустой пароль** у текущего пользователя реального компьютера.

*Примечание:* если запрещено изменение настроек на реальном компьютере, редактирование файлов сайта в дальнейшем осуществляйте прямо на виртуальном компьютере.

#### Б. Простые команды для работы с табличной базой данных средствами СУБД MySQL.

**MySQL** – современная мощная *система управления базами данных* (СУБД). MySQL использует стандартную форму языка запросов SQL, распространяется бесплатно и работает с большим количеством операционных систем (ОС). MySQL может использоваться со многими средами разработки приложений, со многими языками программирования, в том числе PHP. В язык PHP встроено множество функций для работы с MySQL.

MySQL – это программный пакет, включающий в себя серверный компонент – резидентный модуль, клиентскую часть – консоль командной строки MySQL (в ОС Microsoft Windows – mysql.exe в подкаталоге bin каталога программы MySQL), а также несколько вспомогательных утилит.

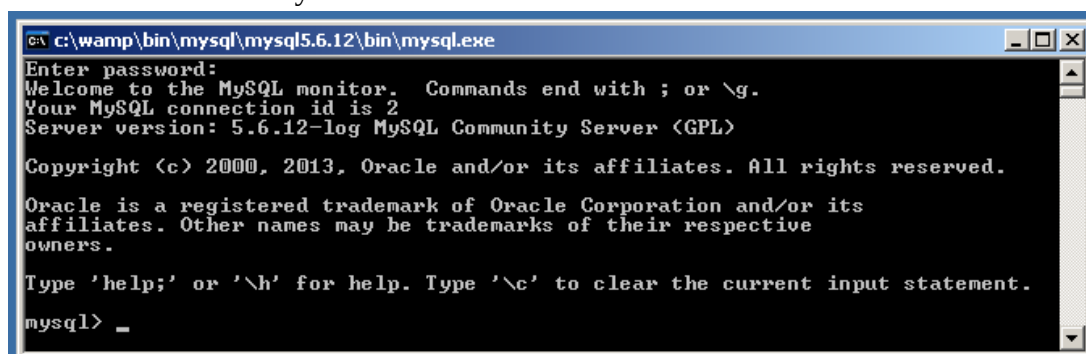


Рис. 1. Вид консоли MySQL после соединения с сервером.

*Консоль MySQL* – это наиболее распространенный инструмент для низкоуровневого доступа к базам данных. Типичная процедура работы с MySQL состоит из соединения с сервером, выбор базы данных и использование SQL-скриптов для работы с ней.

Для иллюстрации основных понятий баз данных создадим простую табличную базу данных – список телефонов. Табличная база данных состоит из строк – *записей*, состоящих из набора *полей*. Одноимённые поля всех записей в таблице образуют столбцы.

Определение соответствующего поля (имя и тип данных) для всех записей таблицы одинаково, но значения поля в различных записях может быть различным. Обязательно различными во всех записях таблицы должны быть значения *ключевого* поля.

В нашем списке телефонов будет 5 полей: id, firstName, lastName, e-mail, phone:

id	firstName	lastName	email	phone
0	Михаил	Рысин	mrysin@mail.ru	+79689106245
1	Grandfa	Moroz	feedback@dom-dm.ru	8 (81738) 5-21-32

СУБД MySQL использует язык структурированных запросов SQL для создания и управления базами данных. Пример кода (скрипта) SQL:

```
# создание списка телефонов
# скрипт для mysql
USE PHP7;
DROP TABLE IF EXISTS phoneList;
CREATE TABLE phoneList (
    id int PRIMARY KEY,
    firstName VARCHAR(15),
    lastName VARCHAR (15),
    email VARCHAR(20),
    phone VARCHAR(15)
);
INSERT INTO phoneList
VALUES (
    0, 'Михаил', 'Рысин', 'mrysin@mail.ru', '+79689106245'
);
SELECT * FROM phoneList;
```

Подобный код SQL интерпретируется и выполняется СУБД MySQL.

5. С помощью **WampServer** (значок на панели индикации) или в папке программы MySQL (C:\wamp\bin\mysql\mysql5.6.12) откройте в Блокноте файл конфигурации my.ini и добавьте в него в соответствующие разделы следующие параметры настройки:

```
[mysqld]
init_connect='SET collation_connection = utf8_unicode_ci'
character-set-server = utf8
collation-server = utf8_unicode_ci
[client]
default-character-set = utf8
```

Эти параметры настроят СУБД MySQL на работу с данными в уже привычной нам многобайтовой кодировке UTF-8 (см. подробнее [6]).

6. С помощью **WampServer** (значок на панели индикации) или в командной строке Windows (cmd.exe) запустите клиентскую часть – консоль MySQL, в консоли введите пароль

пользователя root (пустой). Успешное *соединение клиента с сервером* позволит пользователю далее работать с базами данных, контролируемые этой СУБД.

В командной строке консоли MySQL введите команду:

```
CREATE DATABASE PHP7;
```

Это команда создаст новую базу данных с именем PHP7 (задать имя можно произвольно). Большинство команд в SQL должны заканчиваться символом точка с запятой. Следующей командой выберите только что созданную базу данных для работы с ней:

```
USE PHP7;
```

Найдите в сети Интернет подходящий справочник языка SQL (например, [4]) для его более детального изучения в дальнейшем.

Следующей командой создайте таблицу phoneList:

```
CREATE TABLE phoneList (
    id int PRIMARY KEY,
    firstName VARCHAR(15),
    lastName VARCHAR (15),
    email VARCHAR(20),
    phone VARCHAR(15)
);
```

Примечание: указанную команду можно было ввести и одной строкой, однако, это снизило бы наглядность и читаемость кода.

Получите подтверждение успешного создания таблицы. С помощью найденного ранее справочника по языку SQL изучите типы данных, использованные для полей этой таблицы, а также прочие числовые, дата-время, символьные типы, используемые в языке SQL.

В этом примере символьные поля заданы с определенной длиной. Определение длины полей является важным для работы с таблицей в дальнейшем. При недостаточной длине поля могут возникнуть трудности у пользователей, при излишней – будет неэффективно расходоваться память.

Проверьте структуру созданной таблицы с помощью команды консоли MySQL:

```
DESCRIBE phoneList;
```

7. Добавьте в таблицу новую запись (внесите произвольные данные латиницей):

```
INSERT INTO phoneList VALUES (
0, 'Рысин', 'Михаил', 'mrysin@mail.ru', '+79689106245');
```

Значения полей должны приводиться в том же порядке, в котором были определены поля таблицы. Сами значения в команде отделены запятыми, значения символьных типов указываются в одинарных кавычках.

С помощью справочника по языку SQL изучите синтаксис и возможности команды LOAD DATA для загрузки в таблицу больших массивов данных из файла.

Командой SELECT отобразите всё содержимое таблицы phoneList:

```
SELECT * FROM phoneList;
```

8. С помощью команды DROP TABLE удалите ранее созданную таблицу:

```
DROP TABLE IF EXISTS phoneList;
```

Если таблица phoneList существует, она будет удалена.

9. Создайте в Блокноте на виртуальном компьютере текстовый файл, перепишите в него текст скрипта SQL со стр. 2 (это скрипт для создания таблицы, добавления в неё записи и отображения результата).

В указанном примере первые две строки начинаются с символа # – знака начала однострочного комментария. Уточните в справочнике SQL, как сделать в скрипте многострочный комментарий.

Сохраните скрипт в файле buildPhoneList.sql (в кодировке UTF-8) в корень диска C: виртуального компьютера.

После этого в консоли MySQL наберите команду SOURCE, пробел, затем перетащите на окно консоли значок файла со скриптом и нажмите Enter. Убедитесь, что таблица была успешно воссоздана.

Командой EXIT отключите консоль MySQL от сервера (окно закроется автоматически).

## В. Утилита phpMyAdmin

Язык SQL и консоль MySQL – мощные средства управления базами данных, однако для комфортной работы в наше время предпочтительны средства, предоставляющие пользователю визуальные инструменты управления. Одним из таких средств является программа **phpMyAdmin**.

PhpMyAdmin – это клиентское приложение, написанное на языке PHP для работы (в т.ч. удалённой) с базами данных серверов MySQL непосредственно в браузере пользователя. Эта бесплатная утилита позволяет создавать и редактировать таблицы, а также управлять базами данных, осуществлять экспорт/импорт, а также экспериментировать со структурами данных.

В настоящее время предустановленная утилита phpMyAdmin по умолчанию предоставляется многими хостинговыми компаниями в общем пакете услуг, наряду с интерпретатором языка PHP и СУБД MySQL. Уже установлена phpMyAdmin и на используемом в данной работе веб-сервере в пакете WampServer.

10. Подберите в сети Интернет подходящий информационный ресурс, посвященный phpMyAdmin (например, [5]), найдите по возможности полное описание возможностей этой утилиты. В разделе для скачивания программы уточните, версии для каких операционных систем имеются в наличии, на каких условиях предоставляется программа.

11. Откройте в браузере на реальном компьютере приветственную страницу WampServer виртуального веб-сервера по адресу <http://www/>, на открывшейся странице в разделе «Tools» перейдите по текстовой ссылке «phpMyAdmin».

В открывшемся стартовом окне phpMyAdmin введите имя администратора СУБД MySQL – root с пустым (в нашем случае) паролем. Т.о. phpMyAdmin как MySQL-клиент соединится с сервером MySQL.

Примечание: утилита phpMyAdmin и сервер MySQL не обязательно должны быть на одном компьютере, возможно подключение к любому удалённому серверу СУБД, если его настройки допускают такую возможность.

12. На домашней странице phpMyAdmin (рис. 2) в левой части окна найдите список имеющихся баз данных, в том числе созданную ранее PHP7, и в ней – таблицу phoneList.

Изучите имеющиеся в phpMyAdmin визуальные инструменты для работы как с базами данных (на примере PHP7), так и таблиц (на примере phoneList).

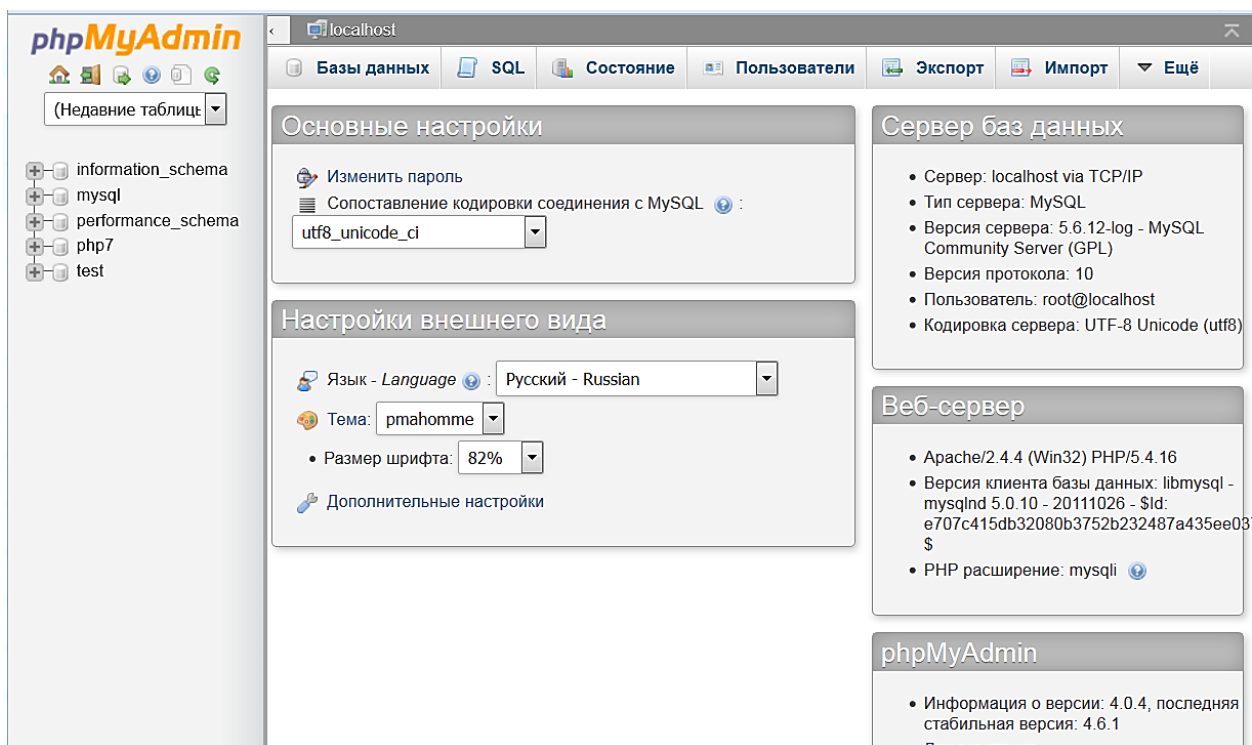


Рис. 2. Домашняя страница браузерной утилиты phpMyAdmin.

13. Во вкладке «Структура» отобразите свойства полей таблицы phoneList. С помощью вкладки «Вставить» добавьте в таблицу phoneList еще несколько записей произвольного содержания.

Добавьте в структуру таблицы phoneList за полем 'lastName' (фамилия) поле 'birthDay' – день рождения. Заполните вручную новое поле произвольными значениями.

Вкладка «Экспорт» открывает широкие возможности вывода данных в самых разнообразных форматах: файл с разделителями CSV, HTML- или XML-документ. Выбор того или иного формата при экспорте данных диктуется используемым для последующей обработки этих данных приложением.

Интересным способом экспорта является формат SQL. Данный формат позволяет не только вывести данные в файл, но и сгенерировать скрипты SQL для автоматического создания баз данных и заполнения таблиц по примеру скрипта buildPhoneList.sql. Этот формат удобен при переносе базы данных динамического веб-сайта на новый хостинг.

Осуществите экспорт таблиц базы данных PHP7 в файл формата SQL. Откройте созданный файл в любом текстовом редакторе, изучите его содержимое. Добавьте в файл комментарии, поясняющие каждую часть скрипта.

## Г. Создание сложных запросов

Большая часть работы с базами данных состоит из процедур выборки необходимой информации (для её последующей обработки).

Уже знакомая нам команда SELECT открывает широкие возможности фильтрации полей и записей по самым разнообразным критериям. В самом простом виде этой командой мы получали всё содержимое таблицы:

```
SELECT * FROM phoneList;
```

Примечание: приложение phpMyAdmin является удобным инструментом для экспериментов с командой SELECT: написав код SQL вручную, можно сразу увидеть результаты его выполнения. Те же эксперименты могут быть выполнены и в консоли MySQL, но выводимый результат не всегда бывает удобно читаем.

Во многих случаях пользователя могут интересовать не все поля в записях таблицы, а лишь некоторые из них. Выберем из таблицы phoneList только значения поля 'email' для, предположим, формирования списка обезличенной почтовой рассылки.

14. Проверьте соответствующую команду выборки средствами phpMyAdmin (во вкладке «SQL» базы данных PHP7):

```
SELECT email
FROM phoneList;
```

Дополните предыдущий запрос, чтобы в выводе, кроме почтового адреса, были указаны также имена. Экспортируйте результат запроса в файл CSV.

Проверьте команду выборки записей из таблицы для определенной фамилии:

```
SELECT *
FROM phoneList
WHERE lastName = 'Moroz';
```

Проверьте команду выборки из таблицы людей, родившихся раньше определенного года:

```
SELECT *
FROM phoneList
WHERE YEAR(birthDay) < 2000;
```

Проверяемое условие может формироваться с использованием знаков сравнения <, >, <=, >=, =, !=.

Примечание: в общем случае SQL не чувствителен к регистру букв.

Проверьте команду выборки из таблицы людей, чья фамилия начинается с буквы Р:

```
SELECT *
FROM phoneList
WHERE lastName LIKE 'P%';
```

Проверьте команду выборки из таблицы людей, чья фамилия начинается с буквы Р и почтовый ящик расположен на mail.ru:

```
SELECT *
FROM phoneList
WHERE lastName LIKE 'P%'
AND email LIKE '%mail.ru';
```

Кроме AND («и»), множественные условия могут формироваться также с помощью команд OR («или») и NOT («не»).

Проверьте команду выборки записей из таблицы сортировкой по определенному полю по возрастанию:

```
SELECT *
FROM phoneList
ORDER BY lastName;
```

Проверьте команду выборки записей из таблицы сортировкой по определенному полю по убыванию:

```
SELECT *
FROM phoneList
ORDER BY birthDay DESC;
```

Проверьте команду изменения значения поля в определенной записи:



```
UPDATE phoneList  
SET email=NULL  
WHERE YEAR(birthDay) >= 2010 ;
```

**Литература:**

1. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство, 2013.
2. Суэринг С. и др. PHP 6 и MySQL 6. Библия программиста, 2010.
3. Янк К. PHP и MySQL. От новичка к профессионалу, 2013.
4. Справочное руководство по MySQL [Электронный ресурс]. URL: <http://www.mysql.ru/docs/man/> (дата обращения 23.05.2016).
5. phpMyAdmin по-русски – сайт для русскоязычных пользователей [Электронный ресурс]. URL: <http://php-myadmin.ru/> (дата обращения 24.05.2016).
6. Корректная настройка MySQL для работы с UTF8 [Электронный ресурс]. URL: <http://gahcep.github.io/blog/2013/01/05/mysql-utf8/> (дата обращения 24.05.2016).