Лабораторная работа №2

Настройка и создания базы данных в PHPMyAdmin

**Цель работы:** Ознакомится с функциями и работой инструмента PHPMyAdmin, произвести его первоначальную настройку и создание базы данных для установки вашего будущего сайта.

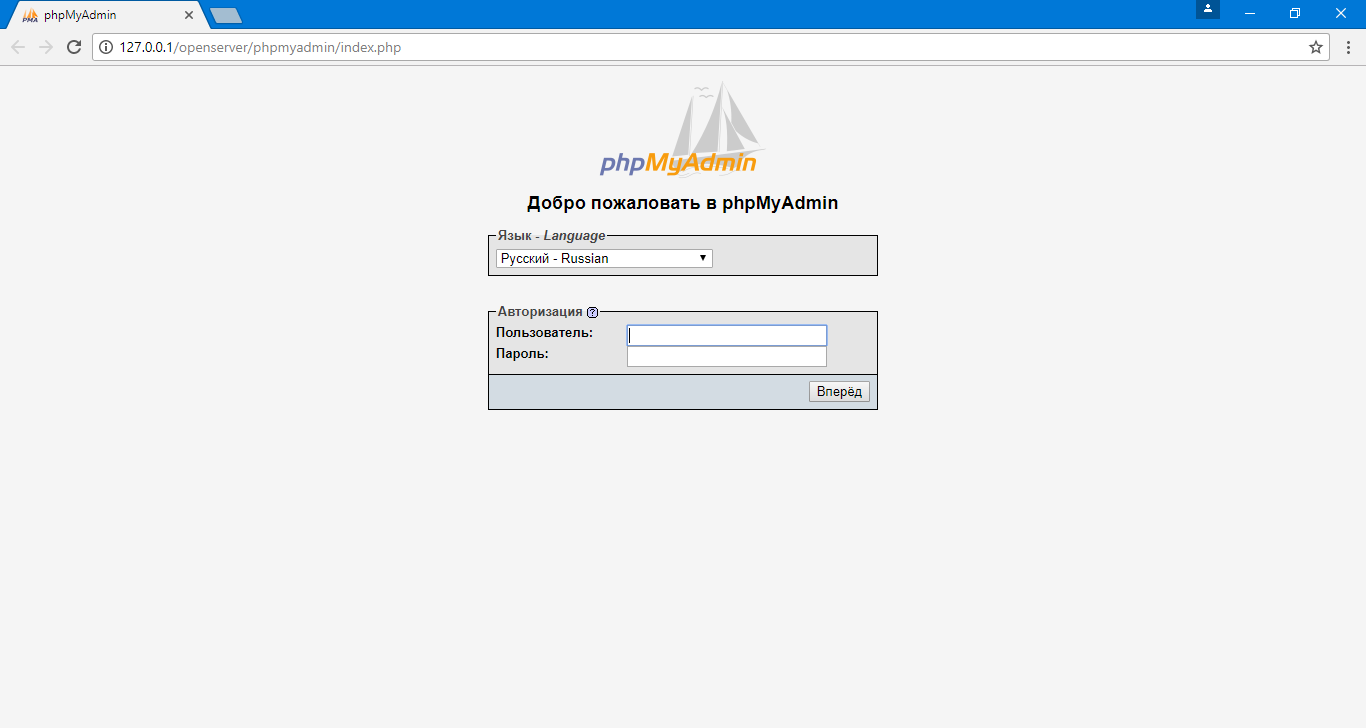
**phpMyAdmin** - это приложение написанное на PHP и обеспечивающее полноценную, в том числе удаленную, работу с базами данных MySQL через браузер. Так как phpMyAdmin позволяет во многих случаях обойтись без непосредственного ввода команд SQL, то работа с базами данных становится вполне посильной задачей даже для человека весьма поверхностно знакомого с MySQL.

Активное использование MySQL в веб-программировании обусловило его актуальность, а интуитивно понятный интерфейс в совокупности с широкой функциональностью и поддержкой более 60 языков (в т.ч. и русского) обеспечило ему популярность среди веб-разработчиков.

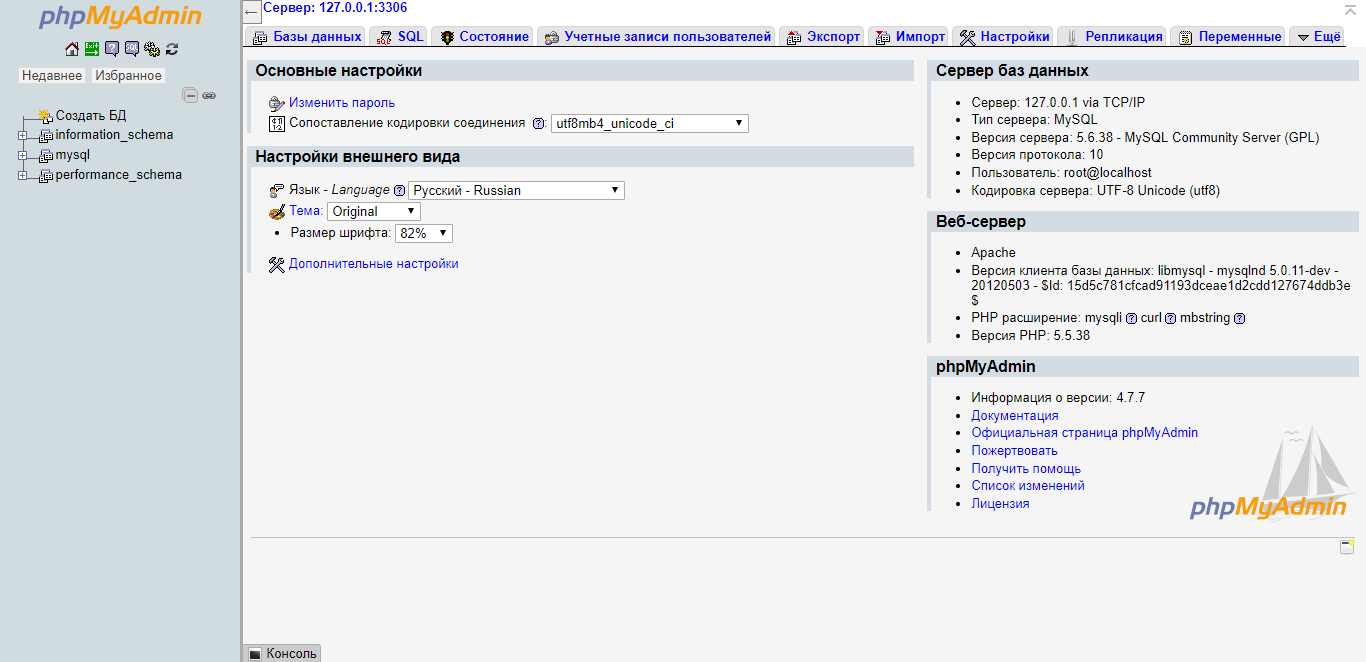
**Ход выполнения работы:**

Для изучения принципов работы и настроек PHPMyAdmin мы создадим две базы данных, первая база данных будет тестовой и носить только обучаемый характер, вторая же база данных будет напрямую привязана к нашему будущему сайту. Начинайте выполнение по порядку.

1. Запустите виртуальный сервер Open Server > перейдите в меню Дополнительно > PHPMyAdmin, после чего в браузере вы увидите открывающеюся страницу нужного нам инструмента.

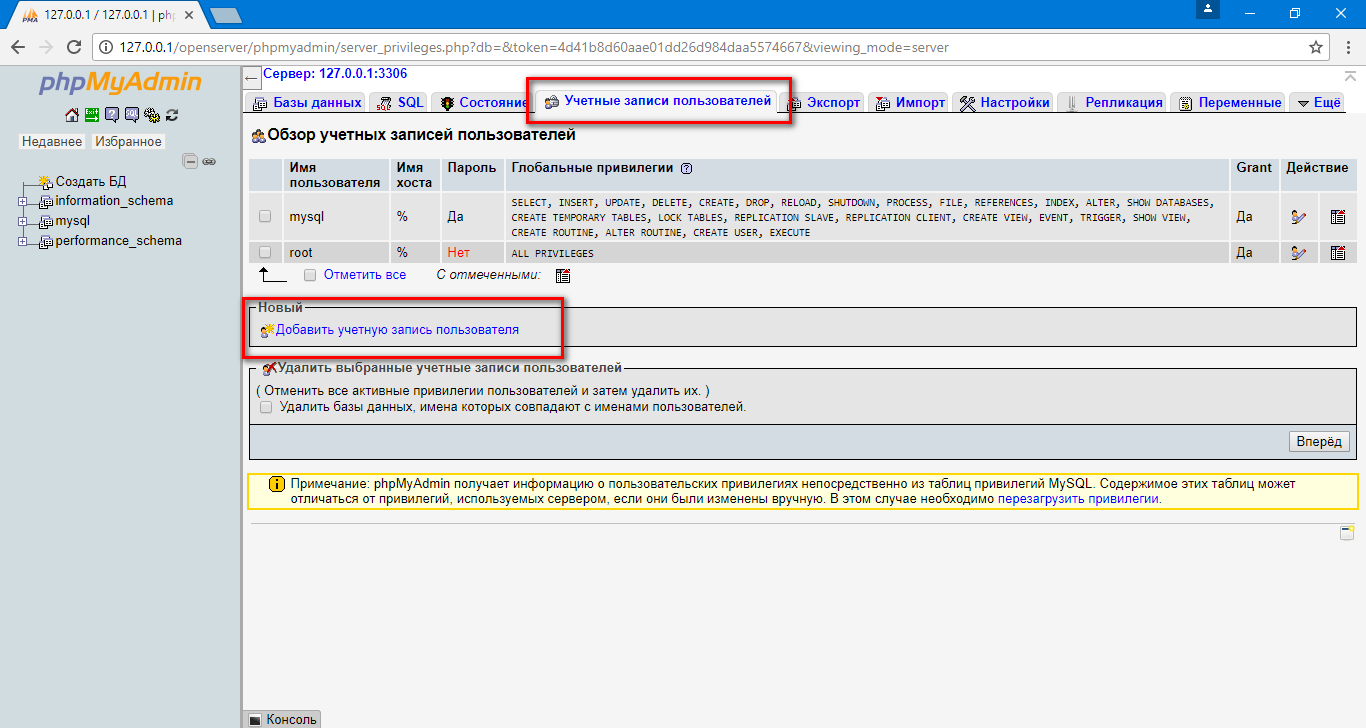


По умолчанию имя пользователя задано root, пароль отсутствует. Авторизуйтесь под данным именем пользователя. Вы увидите рабочую панель инструментов.

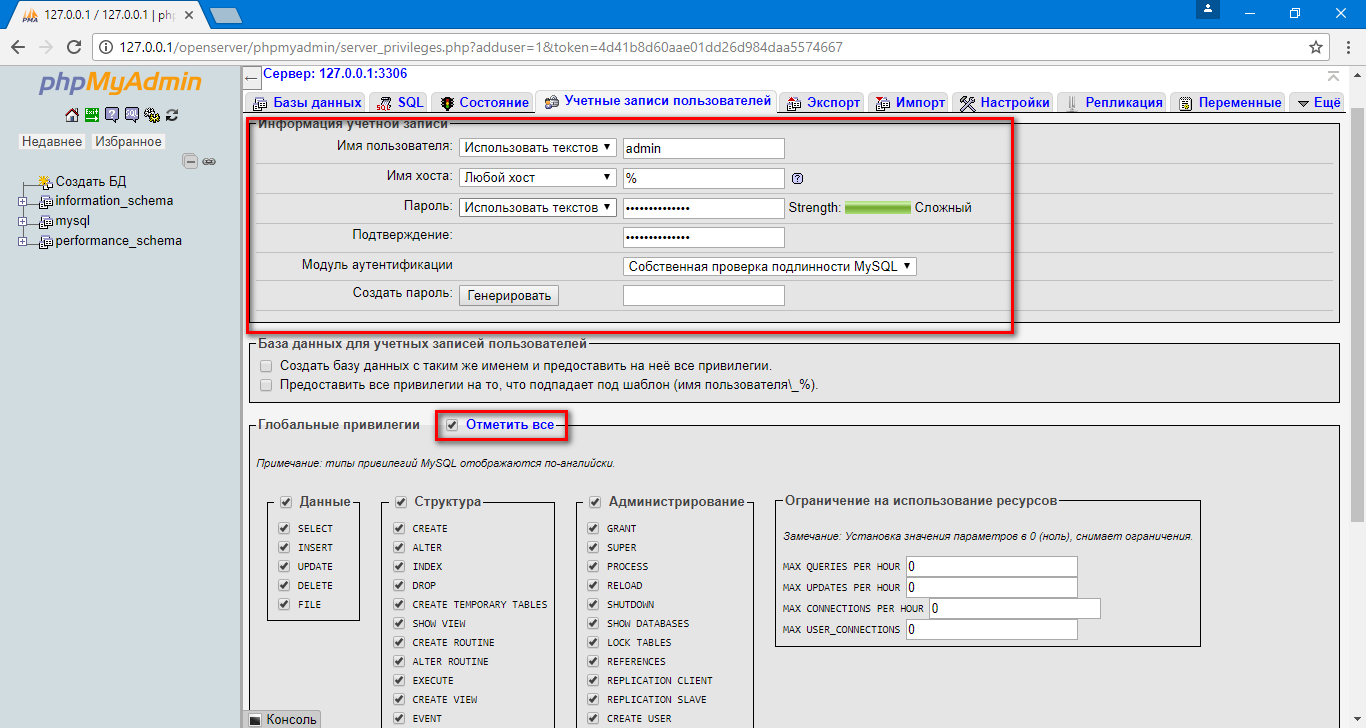


Сейчас здесь отсутствуют какие-либо базы данных за исключением системных, которые не в коем случае нельзя трогать!

Для того, чтобы у каждого пользователя базы данных в том числе и у администратора был собственный логин и пароль нам необходимо завести нового пользователя, для этого перейдите во вкладку «Учетные записи пользователей» и выберите пункт «Добавить учётную запись пользователя:



Вы будете отправлены на страницу добавления нового пользователя, здесь представлено достаточно большое количество настроек, однако вам необходимо заполнить и выделить следующие:



Таким образом вы указываете только данные для авторизации, а также даете данному пользователю в нашем случае это Администратор – все допустимые права по управлению базой данных.

Нажмите вперед, после чего вы увидите сообщение об успешном добавление нового пользователя в вашу базу данных.

1. Выйдите из панели управления PHPMyAdmin и заново авторизуйтесь под новым созданным вами пользователем. Вся дальнейшая работа в рамках этого инструмента будет проходить под новым пользователем.
2. Мы можем приступать к созданию нашей первой базы данных посредством инструмента PHPMyAdmin.

С помощью данного инструмента у пользователя есть два варианта для разработки баз данных, первый вариант это графический посредством пользовательского интерфейса и второй это разработка базы данных исключительно путем написания кода на языке SQL – чем вы и будете заниматься в дальнейшем. **Использование пользовательского интерфейса при разработке базы данных запрещена!**

Давайте теперь научимся работать с базами данных через PHP. Для этого прежде всего необходимо установить соединение с сервером базы данных.

Делается это с помощью функции 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 (на некоторых серверах он тоже может быть root). Если ваша база данных в интернете - то эти данные выдает вам хостинг.

Итак, давайте установим соединение с базой данных:

<?php

$host = 'localhost'; // имя хоста

$user = 'root'; // имя пользователя

$pass = ''; // пароль

$name = 'mydb'; // имя базы данных

$link = mysqli\_connect($host, $user, $pass, $name);

?>

Если указанные нами доступы правильные, то установится соединение к базе данных. При этом в переменную $link запишется специальный объект соединения, который мы будем использовать в дальнейшем для всех обращений к нашей базе данных.

Посылаем запросы к базе данных

После соединения с базой к ней можно отправлять запросы. Это делается с помощью функции mysqli\_query. Первым параметром эта функция принимает принимает переменную, в которую мы записали результат mysqli\_connect, а вторым - строку с SQL запросом.

К примеру, выполним запрос, который достанет все записи из таблицы users:

<?php

$result = mysqli\_query($link, 'SELECT \* FROM users');

?>

Текст запроса не обязательно писать прямо в параметре функции mysqli\_query. Давайте вынесем его в переменную:

<?php

$query = 'SELECT \* FROM users';

$result = mysqli\_query($link, $query);

?>

Отлавливаем ошибки базы данных

Как вы уже знаете, в PHP вывод ошибок на экран включается с помощью функции error\_reporting. Эта функция, однако, не включает вывод ошибок, допущенных в тексте SQL запроса.

Чтобы вывести ошибки SQL команд, следует пользоваться функцией mysqli\_error, которую необходимо добавлять к каждому запросу к БД, вот так:

<?php

$query = 'SELECT \* FROM users';

$result = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

Пока не будем разбираться с тем, как работает эта конструкция. Просто добавляйте ее и, в случае ошибочного SQL запроса вы увидите сообщение об этом в окне браузера.

Тестируем работоспособность

Давайте теперь напишем тестовый код, который вы сможете запустить у себя, чтобы убедится в том, что у вас все подключилось верно.

Для начала убедитесь, что у вас есть база данных mydb, а в ней таблица users, заполненная какими-нибудь данными.

Затем запустите у себя следующий код:

<?php

$host = 'localhost'; // имя хоста

$user = 'root'; // имя пользователя

$pass = ''; // пароль

$name = 'mydb'; // имя базы данных

$link = mysqli\_connect($host, $user, $pass, $name);

$query = 'SELECT \* FROM users';

$result = mysqli\_query($link, $query) or die(mysqli\_error($link));

var\_dump($result);

?>

Если на экране нет ошибок и вы видите результат var\_dump - значит все хорошо. Если же есть какие-то ошибки - исправьте их и попробуйте снова.

**Задания для самостоятельной работы.**

**ВНИМАНИЕ! ВСЕ ДЕЙСТВИЯ ДЕЛАТЬ ТОЛЬКО ПРИ ПОМОЩИ SQL КОМАНД БЕЗ ИСПОЛЬЗОВАНИЯ ВИЗУАЛЬНОГО РЕДАКТОРА!**

**Задание №1**

Создайте базу данных mydb.

**Задание №2**

Поменяйте кодировку для таблицы и для всей базы данных (на utf8\_general\_ci).

**Задание №3**

В ней создайте таблицу users. В этой таблице сделайте 4 поля (столбца) с указанным выше содержимым.

|  |  |  |  |
| --- | --- | --- | --- |
| id | name | age | salary |
| 1 | user1 | 23 | 400 |
| 2 | user2 | 25 | 500 |
| 3 | user3 | 23 | 500 |
| 4 | user4 | 30 | 900 |
| 5 | user5 | 27 | 500 |
| 6 | user6 | 28 | 900 |

**Задание №5**

С помощью запроса вставьте несколько строк в эту таблицу.

**Задание №6**

Проредактируйте какую-нибудь запись.

**Задание №7**

Удалите какую-нибудь запись.

**Задание №8**

Переименуйте таблицу.

**Задание №9**

Переименуйте базу данных.

**Задание №10**

<?php

$host = 'localhost'; // имя хоста

$user = ''; // имя пользователя

$pass = ''; // пароль

$name = 'mydb'; // имя базы данных

$link = mysqli\_connect($host, $user, $pass, $name);

mysqli\_query($link, "SET NAMES 'utf8'");

$query = ‘’;

$result = mysqli\_query($link, $query) or die(mysqli\_error($link));

var\_dump($result);

?>

Создайте файл PHP. Скопируйте исходный код в этот файл.

Напишите запрос для выборки всех пользователей из таблицы Users. Выполните код.