

Создание web-приложений,
исполняемых на стороне сервера при помощи
языка программирования PHP
и технологии AJAX

Урок №3

Базы данных и работа с MySQL

Содержание

Введение	4
Разработка сайта	6
Создание структуры сайта	8
Создание базы данных	12
Обработка SQL запросов в PHP	19
Создание таблиц в базе данных	26
Страница регистрации.....	31
Страница добавления данных	36
Форма добавления и удаления стран	38
Форма добавления и удаления городов.....	44
Форма добавления и удаления отелей	49
Форма добавления картинок для отелей.....	54

Страница выбора тура	57
Страница описания отеля	64
Аутентификация и авторизация на сайте	68
Создание администраторов сайта	74
Подведение итогов	81
Домашнее задание	83

Введение

Когда мы разрабатывали наш первый сайт, нам надо было хранить список зарегистрированных пользователей, чтобы иметь возможность выполнять аутентификацию. В качестве постоянного хранилища для этого списка мы использовали файл на диске. Однако в реальных приложениях файл не является самым эффективным способом хранения данных. Недостатки использования файла известны: медленный доступ и слабая защищенность данных. Альтернативой файлу для хранения данных является использование баз данных (БД). Рассмотрим работу с БД.

Сначала вспомним несколько важных понятий. Мы будем работать с реляционными БД. Реляционная база данных – это набор связанных нормализованных таблиц, в которых располагаются хранимые данные. Реляционные базы данных содержатся и обслуживаются в специальных программных средах, называемых системами управления базами данных (СУБД) или же просто – серверами БД.

Сначала мы будем работать с СУБД MySQL, которая уже давно зарекомендовала себя как надежный инструмент для хранения и обслуживания данных веб-приложений. Потом мы рассмотрим работу с некоторыми другими СУБД.

Наша среда разработки, OpenServer, уже включает в себя установленный экземпляр MySQL, поэтому нам не

надо устанавливать никаких новых инструментов для использования БД.

Как правило, СУБД являются консольными продуктами. Многим пользователям, не привычным к символьному интерфейсу, работать с такими приложениями сложно. Поэтому для предоставления пользователям графического интерфейса для каждой СУБД применяются различные клиентские приложения. В паре с MySQL для этих целей чаще всего используют специальное приложение с веб-интерфейсом, называемое PhpMyAdmin. OpenServer включает в себя также и установленное приложение PhpMyAdmin.

Разработка сайта

Чтобы разобраться с основными вопросами, возникающими при работе с БД, мы начнем создавать новый сайт. В этот раз это не будет абстрактный сайт без пользовательской логики, как наш сайт во втором уроке. Сейчас мы создадим сайт для вымышленного турагентства. Он будет предоставлять пользователю возможность выбрать страну для посещения, затем – город в этой стране, и, наконец, подобрать отель в выбранном городе. Пользователи также смогут оставлять свои отзывы об отелях. Кроме этого, сайт будет предоставлять возможность администратору добавлять, изменять и удалять списки стран, городов и отелей.

Прежде всего, нам надо спроектировать БД для создаваемого сайта. Мы должны описать структуру всех таблиц, которые будут входить в базу данных, связи между этими таблицами, имена и типы полей, ограничения и многое другое. К проектированию БД надо относиться очень внимательно, так как база данных является фундаментом, на котором строится веб-приложение.

Наша БД будет включать в себя такие таблицы со столбцами:

- Countries (id, country);
- Cities (id, city, countryid, ucity);
- Hotels (id, hotel, countryid, cityid, stars, cost, info);
- Images (id, hotelid, imagepath);

- Roles (id, role);
- Users (id, login, pass, email, roleid, discount, avatar).

Хотя приведенное описание таблиц вполне характеризует БД, будет полезно сказать несколько слов дополнительно. В таблицу Hotels добавлены два поля – countryid и cityid. Это внешние ключи для связи с таблицами Countries и Cities, соответственно. Можно видеть, что достаточно было бы использовать только связь с таблицей Cities и через нее выходить на связанную страну, но в этом случае нам пришлось бы использовать многотабличные запросы. Пусть лучше у нас в таблице будет дополнительное поле countryid, зато мы будем быстрее делать выборки отелей по странам. А такие выборки надо будет делать часто. Поле country в таблице Countries надо будет сделать уникальным. Названия же городов и отелей уникальными быть не должны, так как существуют города с одинаковыми названиями в разных странах, и отели с одинаковыми названиями существуют тоже. В таблице Cities мы сделаем уникальным индексом поле ucity, представляющее собой конкатенацию полей city и countryid. Вы видите, что в таблице Hotels нет поля для описания картинки. Вместо этого у нас есть отдельная таблица Images для хранения картинок, что позволит нам для каждого отеля хранить произвольное количество картинок. В таблице Images мы храним пути к картинкам, которые будут располагаться в папке images нашего сайта.

В таблице Users создано поле avatar типа mediumblob для хранения фото администраторов сайта. От пользователей мы не будем требовать обязательно загружать

фотографию, а от администратора – будем. Таким образом, мы рассмотрим работу с изображениями в БД двумя способами: когда в таблице хранится путь к картинке, и когда в таблице хранится сама картинка в поле типа `mediumblob`.

Зачем пользователям регистрироваться на нашем сайте? Особо отличившимся туристам мы будем предоставлять скидки!

Создание структуры сайта

Создайте рядом с папкой `Site1` новую папку с именем `Site2`. В этой папке создайте такие вложенные папки: `pages`, `images`, `css` и `js`. Рядом с этими вложенными папками создайте файл `index.php` – это будет главная страница нашего сайта. Узнаете? Чтобы не повторять уже выполненную работу, можете скопировать целиком всю папку `Site1` и переименовать ее в `Site2`. Лишнее мы уберем, недостающее добавим. Вы должны получить снова такую структуру папок:

 <code>css</code>	30.10.2016 14:59
 <code>images</code>	30.10.2016 14:59
 <code>js</code>	30.10.2016 15:00
 <code>pages</code>	30.10.2016 14:59
 <code>index.php</code>	30.10.2016 15:00

Рис. 1. Структура папок для нового сайта

Для того чтобы создать каркас нового сайта, повторим действия из предыдущего урока. У нас будут такие пункты меню:

- Tours – пользовательский пункт меню для выбора тура;
- Comments – пользовательский пункт меню для комментариев об отелях;
- Registration – пользовательский пункт меню для регистрации;
- Admin Forms – пункт меню администратора сайта для обслуживания наших справочников.

Позаботьтесь о том, чтобы в папке pages нашего сайта были такие файлы (пока пустые):

menu.php
functions.php
Tours.php
Admin.php
Registration.php
Comments.php

Мы уже делали все это при разработке первого сайта, поэтому сейчас будем акцентировать внимание только на новых моментах. Приведите файл menu.php к такому виду:

```
<ul class="nav nav-tabs nav-justified">
<li <?php echo ($page==1) ? "class='active'" :""
    ?>>
    <a href="index.php?page=1">Tours</a></li>
<li <?php echo ($page==2) ? "class='active'" :""
    ?>>
    <a href="index.php?page=2">Comments</a></li>
<li <?php echo ($page==3) ? "class='active'" :""
    ?>>
    <a href="index.php?page=3">Registration</a></li>
<li <?php echo ($page==4) ? "class='active'" :""
    ?>>
    <a href="index.php?page=4">Admin Forms</a></li>
</ul>
>
```

Здесь используется тернарный оператор для того, чтобы выделенному пункту меню присвоить класс active.

Теперь создадим разметку главной страницы нашего сайта – файла index.php:

```
<?php
session _ start();
include _ once ("pages/functions.php");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Travel Agency</title>
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css"
                      rel="stylesheet">
    <link href="css/style1.css" rel="stylesheet">
</head>
<body>
<div class="container" >
    <div class="row">
        <header class="col-sm-12 col-md-12 col-lg-12">
        </header>
    </div>
    <div class="row">
        <nav class="col-sm-12 col-md-12 col-lg-12">
            <?php
                include _ once('pages/menu.php');
            ?>
        </nav>
    </div>
    <div class="row">
        <section class="col-sm-12 col-md-12 col-lg-12">
            <?php
                if(isset($ _ GET['page']))
                {

```

```
$page=$_GET['page'];
if($page==1)
    include __once("pages/tours.php");
    if($page==2)
        include __once("pages/comments.php");
        if($page==3)
            include __once("pages/registration.php");
            if($page==4)
                include __once("pages/admin.php");
}
?>
</section>
</div>
<div class="row">
    <footer>Step Academy &copy;</footer>
</div>
</div>
<!-- jQuery (necessary for Bootstrap's JavaScript plugins) --&gt;
&lt;script src="https://ajax.googleapis.com/ajax/libs/
jquery/1.12.4/jquery.min.js"&gt;
&lt;/script&gt;
<!-- Include all compiled plugins (below), or include
individual files as needed --&gt;
&lt;script src="js/bootstrap.min.js"&gt;&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

Эта разметка практически совпадает с разметкой главной страницы нашего предыдущего сайта. Но есть несколько особенностей. Обратите внимание, что в самом начале файла мы открываем сессию, которая, конечно же, понадобится нам в дальнейшем, и здесь же мы подгружаем файл с функциями.

Создание базы данных

Пришло время создать для нашего сайта базу данных и таблицы в ней. Чтобы выполнить эту работу, будет полезно познакомиться с программой PhpMyAdmin, являющейся клиентом для СУБД MySQL и предоставляющей пользователю графический интерфейс для работы с СУБД.

Однако до того, как мы познакомимся с клиентским приложением PhpMyAdmin, давайте помотрим как выглядит работа с MySQL в консоли. Убедитесь, что OpenServer запущен и на вкладке Modules меню Settings посмотрите, какая версия MySQL у вас установлена. Скорее всего, это версия MySQL-5.5. Теперь откройте в проводнике папку, в которой установлен OpenServer и пройдите по такому пути, к папке, в которой расположена наша СУБД MySQL. У меня это такой путь: C:\openserver\modules\database\MySQL-5.5\bin. Скопируйте этот путь в клипбоард, он вам сейчас понадобиться.

Нажмите Win+R и в появившемся окне занесите команду cmd, затем нажмите кнопку OK. Вы запустите консольное окно, с помощью которого мы сейчас зайдем в гости к MySQL. Перейдите в консольное окно и занесите в него команду смены папки cd (change directory), после которой через пробел укажите свой путь из клипборда. Если в пути присутствуют пробелы, то его лучше взять в кавычки. После нажатия Enter вы перейдете в эту папку.

```
C:\> cd "C:\openservert\modules\database\MySQL-5.5\bin"
```

Рис. 2. Смена текущей папки в консольном окне

Теперь можно вызвать MySQL, например, так:

mysql -u root -p

После ввода этой команды надо будет еще ввести пароль:

```
C:\> cd "C:\openservert\modules\database\MySQL-5.5\bin"
C:\openservert\modules\database\MySQL-5.5\bin>mysql -u root -p
Enter password: *****
```

Рис. 3. Вход на MySQL сервер в консольном окне

Если вы ввели правильный пароль, то войдете на сервер MySQL, о чем вам подскажет внешний вид подсказки:

```
C:\> cd "C:\openservert\modules\database\MySQL-5.5\bin"
C:\openservert\modules\database\MySQL-5.5\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.50 MySQL Community Server (GPL)

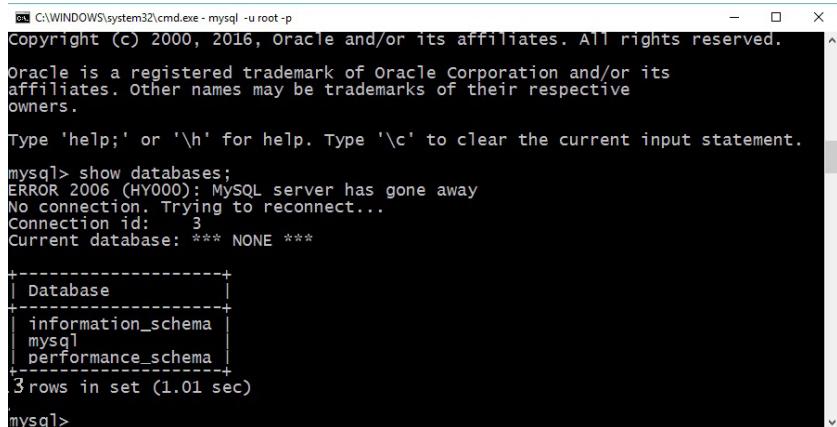
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

Рис. 4. Вид консольного окна при входе на сервер

С помощью команды show databases можно посмотреть список уже существующих на сервере баз данных. Введите такую команду, только в конце каждой команды в консольном окне не забывайте ставить символ ';'.



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

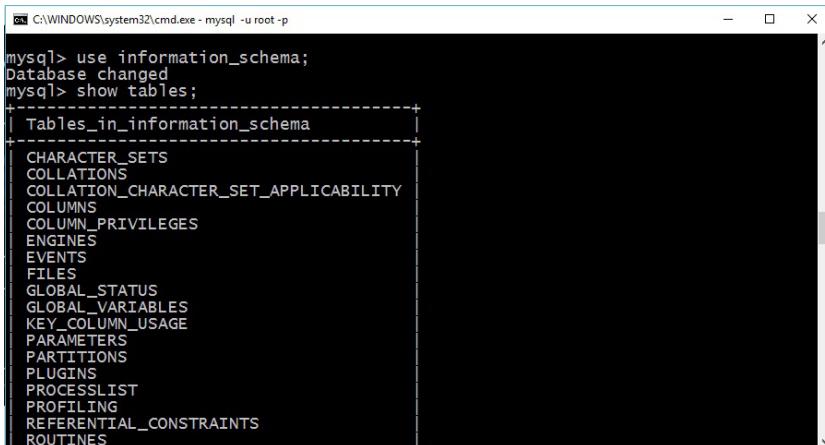
mysql> show databases;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id:      3
Current database: *** NONE ***

+-----+
| Database
+-----+
| information_schema
| mysql
| performance_schema
+-----+
3 rows in set (1.01 sec)

mysql>
```

Рис. 5. Список системных баз данных

Вы увидите три уже существующие системные БД. Кстати, обратите внимание, что пока мы с вами отвлеклись, MySQL уже выгрузился из памяти, но когда получил от нас новую команду, то снова активировался и выполнил нашу команду. Это можно проследить по выводу в окне: MySQL has gone away и Trying to reconnect. MySQL выполняется как служба, а служба переходит в режим ожидания, если не получает запросов на протяжении определенного времени. Давайте зайдем в БД information_schema, введя в консольное окно команду use information_schema; Потом попросим показать нам все таблицы в этой БД, введу команду show tables;



```
cmd C:\WINDOWS\system32\cmd.exe - mysql -u root -p

mysql> use information_schema;
Database changed
mysql> show tables;
+-----+
| Tables_in_information_schema |
+-----+
| CHARACTER_SETS
| COLLATIONS
| COLLATION_CHARACTER_SET_APPLICABILITY
| COLUMNS
| COLUMN_PRIVILEGES
| ENGINES
| EVENTS
| FILES
| GLOBAL_STATUS
| GLOBAL_VARIABLES
| KEY_COLUMN_USAGE
| PARAMETERS
| PARTITIONS
| PLUGINS
| PROCESSLIST
| PROFILING
| REFERENTIAL_CONSTRAINTS
| ROUTINES |
+-----+
```

Рис. 6. Список таблиц в базе данных information_schemas

Зная имена таблиц можно в консольном окне создавать любые запросы. Однако такой стиль работы применяется только в отладочных или тестовых целях. Давайте отключимся от сервера командой exit; и перейдем к знакомству с PhpMyAdmin.

Мы уже упоминали, что PhpMyAdmin включен в установку OpenServer, поэтому нам надо просто запустить эту программу из главного меню OpenServer. Выберите в трее пиктограмму OpenServer и из контекстного меню выберите пункт Advanced, а затем – PhpMyAdmin (Рис. 7).

После запуска PhpMyAdmin в браузере откроется окно входа на сервер MySQL. Этот сервер всегда создается со встроенной учетной записью администратора с именем root. Пароля изначально нет. Мы зададим свой пароль после первого входа (Рис. 8).

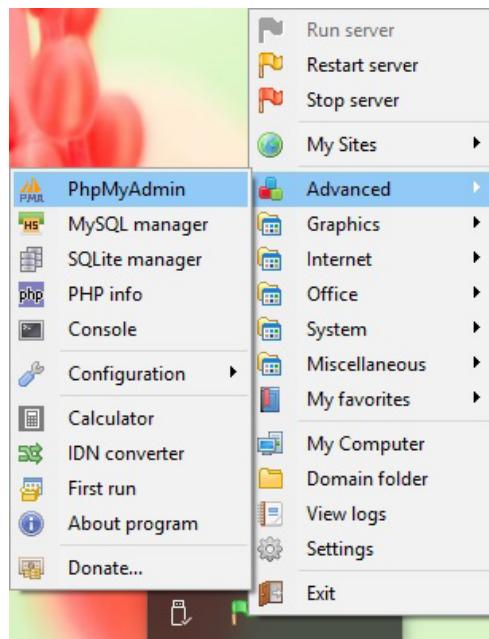


Рис. 7. Запуск PhpMyAdmin

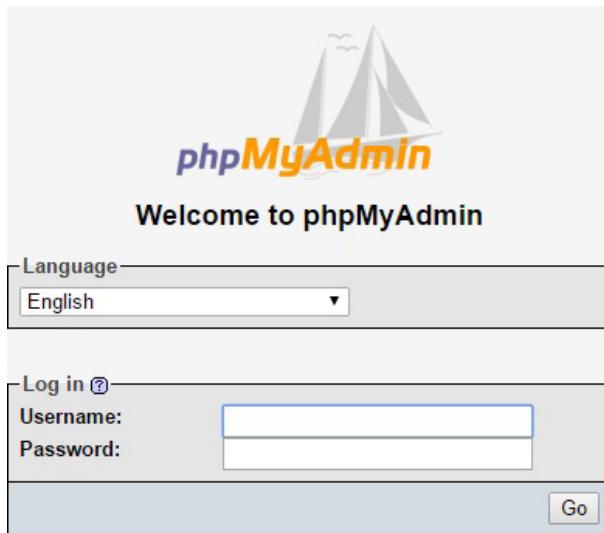


Рис. 8. Окно входа в PhpMyAdmin

Для первого входа введите в окно имя root, поле пароля оставьте пустым и нажмите кнопку Go. Вы увидите главное окно PhpMyAdmin:

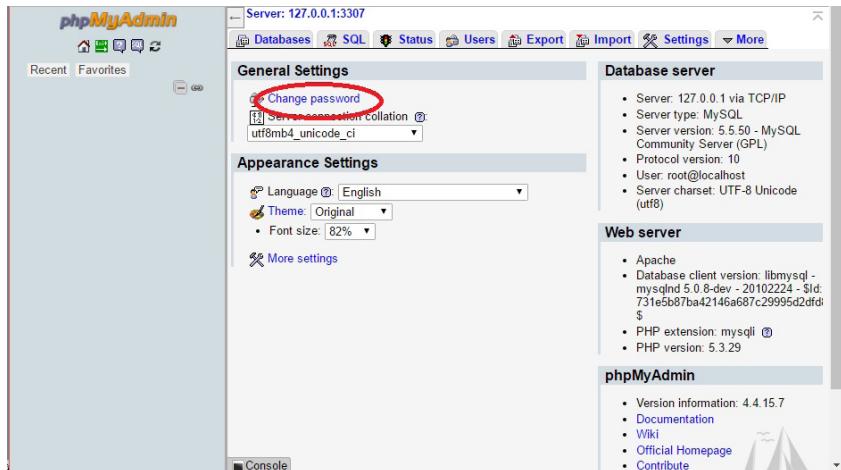


Рис. 9. Главное окно PhpMyAdmin

Давайте сейчас зададим пароль для доступа к MySQL. Для этого кликните по обведенной ссылке Change password. Затем в появившемся окне смены пароля введите свой пароль и нажмите кнопку Go.

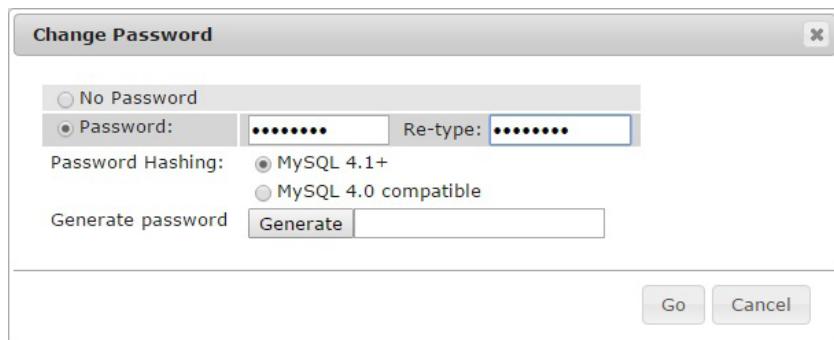


Рис. 10. Окно смены пароля

Вы получите сообщение о том, что пароль изменен. Дальше в уроке будем считать, что наш пароль – «123456». Я уверен, вы понимаете важность создания надежного пароля, поэтому ваш пароль будет не короче 8 символов и будет включать не только буквы и цифры, но и разделители.

Давайте с помощью PhpMyAdmin создадим базу данных для нашего сайта, а потом, с помощью PHP, создадим таблицы в этой базе данных. Активируйте первый пункт меню Databases, и вы увидите форму для создания новой БД. В первое поле занесите имя создаваемой БД, во второе поле занесите значение utf8_general_ci, чтобы БД поддерживала кодировку utf8. После этого нажмите кнопку Create.

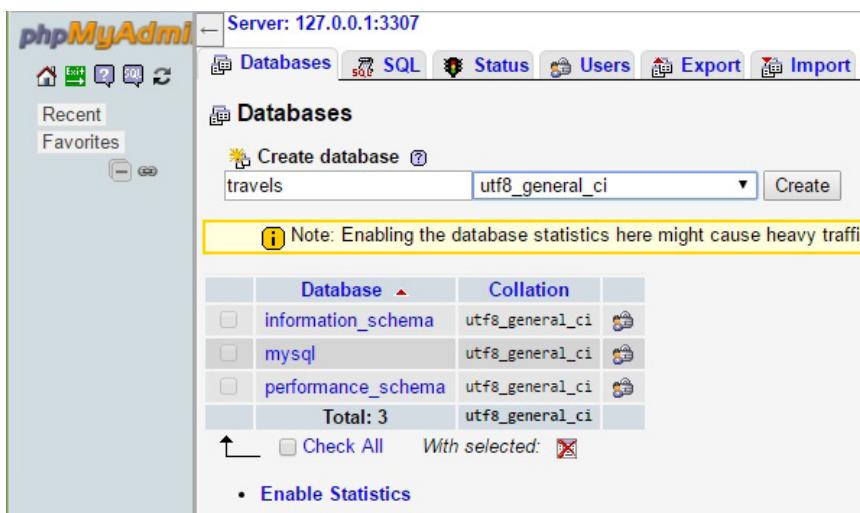


Рис. 11. Создание базы данных

Чтобы убедиться в том, что БД создана, посмотрите на левую панель в главном окне. Вы увидите там свою

БД рядом с тремя системными базами данных. Раскройте кнопку + слева от имени своей БД, чтобы убедиться, что созданная БД пока пуста.

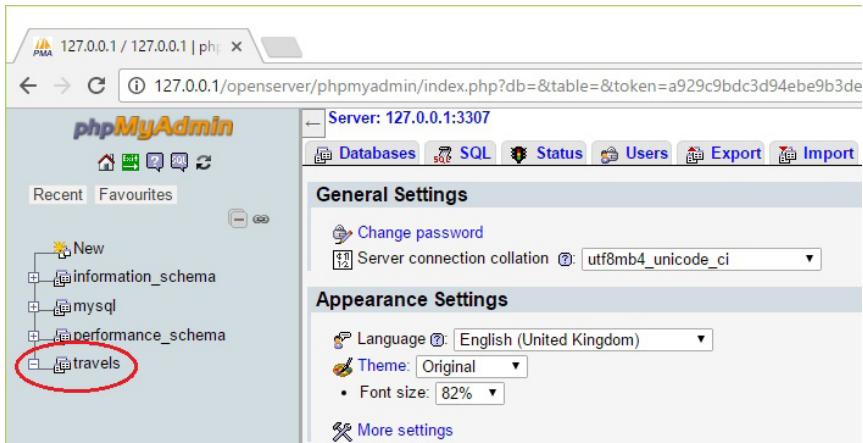


Рис. 12. Созданная база данных

Мы с вами часто будем обращаться к приложению PhpMyAdmin, и будем знакомиться с другими его возможностями по мере необходимости. А сейчас займемся созданием необходимых таблиц в БД *travels*.

Сейчас мы сделаем небольшое отступление. Мы собираемся написать свою функцию, которая будет взаимодействовать с MySQL. Значит, в этой функции нам надо будет использовать возможности PHP, позволяющие реализовать такое взаимодействие. Вот об этих возможностях и поговорим.

Обработка SQL запросов в PHP

Одним из ключевых вопросов в работе с БД является обработка SQL запросов. Давайте посмотрим, как PHP работает с запросами к MySQL.

PHP предлагает несколько способов обработки SQL запросов. Один из них, с которого мы и начнем, это применение функций, имена которых начинаются с `mysql_`.

Из каких действий складывается любой сеанс работы с БД? Прежде всего надо подключиться к серверу (или СУБД), где хостится необходимая БД. Поскольку баз данных на сервере может быть много, то после подключения к серверу надо указать, какая БД нужна нам для работы.

Для подключения к серверу MySQL в PHP есть функция `mysql_connect()`. В случае удачного подключения функция `mysql_connect()` возвращает ссылку на созданное подключение. При вызове этой функции ей надо передать как минимум, адрес MySQL сервера, имя пользователя, зарегистрированного на сервере, и пароль этого пользователя.

```
$link = mysql_connect('localhost', 'root', '123456');
```

Здесь `$link` – это и есть ссылка на созданное подключение. Для нашего сайта, который будет использовать только одну БД, такая ссылка не обязательна, поэтому мы сможем вызывать функцию `mysql_connect()`, не сохраняя ее возвращаемое значение. В случае подключения к нескольким БД надо обязательно сохранять ссылку на созданное подключение. О других особенностях этой функции смотрите здесь:

<http://php.net/manual/en/function.mysql-connect.php>.

Подключившись к серверу, надо выбрать необходимую БД и открыть ее. Для этого существует функция `mysql_select_db()`. При вызове этой функции надо пере-

дать ей строковое имя требуемой БД, и mysql_select_db() откроет доступ к этой БД.

```
$db_selected = mysql_select_db('travels', $link);
```

Здесь \$link – это ссылка на подключение, созданная вызовом функции mysql_connect(). Если вы не используете эту ссылку, то вызов mysql_select_db() может быть таким:

```
$db_selected = mysql_select_db('travels');
```

Возвращаемое значение \$db_selected равно true при успешном входе в БД, и false – в случае ошибки. О других особенностях этой функции смотрите здесь:

<http://php.net/manual/en/function.mysql-select-db.php>

Когда вы подключились к СУБД и выбрали необходимую вам БД, вы можете начинать работу с ней. Такая работа сводится к выполнению SQL запросов. Вот здесь на помощь приходит функция mysql_query(). Это одна из наиболее часто используемых функций среди всего семейства mysql_. Сейчас мы подробно разберем использование функции mysql_query().

При вызове этой функции необходимо в первом параметре передать тело SQL запроса, который надо выполнить. Вторым параметром можно указывать ссылку на конкретное подключение:

```
mysql_query('INSERT INTO CITIES (city) VALUES  
            ("London")', $link);
```

При работе с одним подключением второй параметр можно не указывать:

```
mysql_query('INSERT INTO CITIES (city)  
VALUES ("London")');
```

Иногда бывает удобнее занести тело запроса в переменную и уже эту переменную передавать функции mysql_query() при вызове:

```
$ins='INSERT INTO CITIES (city) VALUES ("London")';  
mysql_query($ins);
```

Каждый из приведенных вызовов mysql_query() приведет к тому, что будет выполнен указанный запрос INSERT и в таблицу Cities будет добавлена одна запись.

Функции mysql_query() можно передавать для выполнения любые SQL запросы, поддерживаемые MySQL. Однако надо отдельно рассмотреть, как использовать эту функцию при обработке запросов типа SELECT, когда необходимо получать результат выполнения запроса. В этом случае обязательно сохранять возвращаемое значение функции mysql_query(). Запомните, что при выполнении запросов, которые не возвращают значения (INSERT, UPDATE, DELETE и др.), функция mysql_query() возвращает true при успешном завершении и false – в случае ошибки. Однако при выполнении запросов, возвращающих результат (SELECT, SHOW и др.), ситуация другая:

```
$resource=mysql_query('SELECT * FROM COUNTRIES');
```

Здесь переменная \$resource представляет собой специальный тип данных PHP, называемый Resource. По своей природе переменная типа Resource – это ссылка на какие-то внешние по отношению к PHP ресурсы. В нашем случае – указание на набор строк, возвращенных запросом SELECT. Чтобы выбрать данные из такого набора строк, надо использовать другие PHP функции семейства mysql_.

Рассмотрим самый популярный способ излечения данных из ресурса – использование функции mysql_fetch_array(). Эта функция принимает ресурс, полученный в результате выполнения запроса, и читает из этого ресурса строку, возвращая прочитанную строку в виде массива:

```
$resource=mysql_query('SELECT * FROM COUNTRIES');
while($row=mysql_fetch_array($resource))
{
    echo $row[0]. ' ' . $row[1]. '<br/>';
}
```

Предположим, что в таблице Countries есть три строки такого вида:

1	Argentina
2	Belgium
3	Canada

Как выполнится приведенный выше код? Сначала вызов mysql_query() создаст ресурс \$resource, в котором будет находиться прочитанная из таблицы информация. Затем цикл while начнет вызывать функцию mysql_fetch_array(). После первого вызова эта функ-

ция создаст массив \$row из двух элементов. В этом массиве будет два элемента, потому что выполненный запрос SELECT возвращает два поля. В первом элементе (\$row[0]) будет значение '1', во втором (\$row[1]) – 'Argentina'. Эти два значения будут выведены в теле цикла. После этого функция mysql_fetch_array() будет вызвана во второй раз и занесет в массив \$row значения '2' и 'Belgium' и т.д. Когда вся таблица будет прочитана, mysql_fetch_array() занесет в \$row значение false, и цикл прекратит работу. На первый взгляд, все просто и понятно. Однако у функции mysql_fetch_array() есть одна интересная особенность. Если эту функцию вызвать, как в приведенном примере, т.е., указав ей только один параметр – ресурс \$resource, то она вернет не один массив \$row, а два массива с именем \$row. Один из этих массивов будет обычным, а второй – ассоциативным.

Если бы вы написали код вот так:

```
$resource=mysql_query('SELECT * FROM COUNTRIES');
while($row=mysql_fetch_array($resource))
{
    echo $row[0] . ' ' . $row['country'] . '<br/>';
}
```

он бы выполнился точно так же, как и в предыдущем случае. Но при этом значение идентификатора извлекалось бы из обычного массива \$row по индексу 0, а название страны – из ассоциативного массива \$row по индексу 'country'. Использование сразу двух массивов может быть избыточным, поэтому вы можете указать функции mysql_fetch_array(), какой массив вы хотите от нее получить: индексный или ассоциативный. Это можно

сделать, передавая этой функции второй параметр, который принимает одну из трех предопределенных констант:

- MYSQL_NUM** – чтобы получить индексный массив;
- MYSQL_ASSOC** – чтобы получить ассоциативный массив;
- MYSQL_BOTH** – чтобы получить оба массива (это значение по умолчанию).

Например, при таком вызове, в цикле можно использовать только ассоциативный массив \$row:

```
$resource=mysql_query('SELECT * FROM COUNTRIES');
while($row=mysql_fetch_array($resource, MYSQL_ASSOC))
{
    echo $row['id']. ' '. $row['country']. 
'<br/>';
}
```

Вы можете посмотреть более подробную информацию об этой функции по ссылке:

<http://php.net/manual/en/function.mysql-query.php>

Функция mysql_fetch_array() не единственная в составе PHP позволяет выбирать данные из ресурса. Есть еще похожая функция mysql_fetch_assoc(). Как можно догадаться по имени функции, она возвращает только ассоциативный массив. В остальном ведет себя так же, как и mysql_fetch_array().

Вы должны понимать, что не всякая команда адресованная СУБД обязательно успешно завершиться. По ходу выполнения любой команды может возникнуть возникнуть много различных препятствий. Поэтому в PHP есть специальная функция mysql_errno(). Ее можно

вызывать после любого запроса, направленного серверу. Если выполнение запроса прошло успешно эта функция вернет 0, если же возникнет ошибка – эта функция вернет код этой ошибки. По коду ошибки можно определить, что пошло не так.

Вы можете посмотреть более подробную информацию об этой функции по ссылке:

<http://php.net/manual/en/function.mysql-errno.php>

Этой информации о семействе функций mysql_ вам достаточно, чтобы понять код, который сейчас будет написан. Наше отступление завершено, и мы возвращаемся к созданию таблиц.

Создание таблиц в базе данных

Мы уже определились с тем, какие таблицы нам надо создать, теперь давайте обсудим, каким способом лучше создать их. Конечно же, можно создать все таблицы с помощью PhpMyAdmin. Но это не будет оптимальным решением. Когда вы вернетесь домой, вам надо будет создать такую же БД на домашнем компьютере. И что, снова все делать вручную? Мы поступим иначе: напишем скрипт на PHP, который будет создавать таблицы для нашей БД. Достаточно будет выполнить этот скрипт на любом другом компьютере, и вы получите все таблицы для БД.

Мы остановились на том, что нам надо написать свою функцию, которая будет выполнять взаимодействие с MySQL.

Вы же помните, что все функции для сайта мы описываем в файле functions.php. Откройте этот файл в блокноте и добавьте в него такое определение функции:

```

function connect(
    $host='localhost',
    $user='root',
    $pass='123456',
    $dbname='travels')
{
    $link=mysql_connect($host,$user,$pass) or
        die('connection error');
    mysql_select_db($dbname) or die('DB open
        error');
    mysql_query("set names 'utf8'");
}

```

У параметров этой функции указаны значения по умолчанию, поэтому мы сможем вызывать ее без указания параметров. Это немного упростит код.

Обратите внимание на конструкцию `or die()` после вызова `mysql_connect()`. При неудачной попытке подключиться к серверу, эта конструкция приведет к прекращению работы нашей функции `connect()`, выведя при этом в браузер сообщение «`connection error`». Это не самый лучший способ обработки ошибки, но вы должны увидеть и такой синтаксис, поскольку он очень широко использовался ранее. В дальнейшем мы будем обрабатывать ошибки более тонко. Аналогичная конструкция используется и после вызова функции `mysql_select_db()`.

Созданную функцию `connect()` мы будем вызывать всякий раз, когда нам надо будет подключаться к БД.

Теперь мы можем перейти к созданию таблиц. Добавьте в папку файл с именем `createdb.php`. Этот файл не будет обслуживать какой-нибудь пункт меню нашего сайта. Он будет запускаться один раз в жизненном цикле сайта для создания таблиц в БД.

Добавьте в этот файл такой код:

```
<?php
include_once('functions.php');
connect();
$ct1='create table countries(
    id int not null auto_increment primary key,
    country varchar(64) unique
)default charset="utf8"';  
  

$ct2='create table cities(
    id int not null auto_increment primary key,
    city varchar(64),
    countryid int,
    foreign key(countryid) references countries(id)
    on delete cascade,
    ucity varchar(128),
    unique index ucity(city, countryid))default
    charset="utf8"';  
  

$ct3='create table hotels(
    id int not null auto_increment primary key,
    hotel varchar(64),
    cityid int,
    foreign key(cityid) references cities(id) on
        delete cascade,
    countryid int,
    foreign key(countryid) references countries(id)
        on delete cascade,
    stars int,
    cost int,
    info varchar(1024))default charset="utf8"';  
  

$ct4='create table images(
    id int not null auto_increment primary key,
    imagepath varchar(255),
    hotelid int,
    foreign key(hotelid) references hotels(id) on
    delete cascade)
    default charset="utf8"';
```

```
$ct5='create table roles(
    id int not null auto_increment primary key,
    role varchar(32))default charset="utf8"';  
  
$ct6='create table users(
    id int not null auto_increment primary key,
    login varchar(32) unique,
    pass varchar(128),
    email varchar(128),
    discount int,
    roleid int,
    foreign key(roleid) references roles(id) on delete
    cascade,
    avatar mediumblob
)default charset="utf8";  
  
mysql_query($ct1);
$err=mysql_errno();
if ($err){
    echo 'Error code 1:'.$err.'<br>';
    exit();
}
mysql_query($ct2);
$err=mysql_errno();
if ($err){
    echo 'Error code 2:'.$err.'<br>';
    exit();
}
mysql_query($ct3);
$err=mysql_errno();
if ($err){
    echo 'Error code 3:'.$err.'<br>';
    exit();
}
mysql_query($ct4);
$err=mysql_errno();
if ($err){
    echo 'Error code 4:'.$err.'<br>';
    exit();
}
```

```
mysql_query($ct5);
$err=mysql_errno();
if ($err){
    echo 'Error code 5:'.$err.'<br>';
    exit();
}
mysql_query($ct6);
$err=mysql_errno();
if ($err){
    echo 'Error code 6:'.$err.'<br>';
    exit();
}
```

Обратите внимание на вызов нашей функции `connect()`. Далее в этом коде создано шесть переменных, в каждую из которых занесен SQL запрос, создающий одну из таблиц нашей БД. Каждая из этих переменных передается в вызов функции `mysql_query()`, которая выполняет эти запросы и таким образом создает таблицы. В результате этого в нашей базе данных создаются необходимые нам таблицы.

Обратите внимание на вызов функции `mysql_errno()` после каждого вызова функции `mysql_query()`. Функция `mysql_errno()` позволяет определить код завершения запроса. В случае возникновения какой-либо ошибки, эта функция возвращает код ошибки, по которому можно понять, где возникла проблема. Поэтому мы выводим эти коды. Это и есть более удобный способ обработки ошибок, о котором мы говорили ранее. Удобство заключается в том, что мы не просто фиксируем факт возникновения ошибки, а при этом еще и предоставляем разработчику код этой ошибки.

Активируйте файл `createdb.php`, введя в браузер такой адрес: <http://localhost/site2/pages/createdb.php>

Затем перейдите в PhpMyAdmin и убедитесь, что в нашей БД появились созданные таблицы. Теперь вы сможете выполнить этот файл на другом компьютере, например дома, и там будут созданы эти же таблицы. Правда, перед этим вы должны будете создать БД, но это вы уже делать умеете.

Страница регистрации

Перейдем к созданию страницы регистрации registration.php. В этот раз мы будем делить посетителей нашего сайта на три категории: администраторы сайта, клиенты и просто гости. Клиенты – это такие же посетители сайта, как и гости. Они отличаются от гостей тем, что являются зарегистрированными посетителями и могут получать скидки. Но регистрация совсем не подразумевает присваивание клиенту скидки. С точки зрения доступа к ресурсам сайта, у гостей и клиентов никакой разницы не будет. У клиентов и гостей не будет доступа к меню Admin Forms.

Для такого деления клиентов у нас есть таблица Roles, в которой изначально будут роли Admin и Customer. По умолчанию зарегистрированному посетителю присваивается роль Customer. Изменять роли может только администратор сайта. Вы уже догадались, что сейчас зарегистрированных пользователей мы будем хранить в БД.

Чтобы двигаться дальше, занесем в таблицу Roles упоминавшиеся две роли: Admin и Customer. Это можно сделать в PhpMyAdmin. Выделите в левой части главной страницы PhpMyAdmin таблицу Roles и активируйте пункт меню Insert:

Урок № 3

The screenshot shows the phpMyAdmin interface for the 'travels' database. The left sidebar lists databases: information_schema, mysql, performance_schema, and travels. Under 'travels', there are tables: cities, countries, hotels, images, roles (which is selected and highlighted in purple), and users. The main area shows the 'roles' table with two rows: 'Admin' (id: 1) and 'Customer' (id: 2). Below the table is a 'Query results operations' section with links for Print view, Export, Display chart, and Create view.

Рис. 13. Выбор таблицы для добавления данных

На появившейся после активации меню Insert странице заполните две предложенные формы и нажмите кнопку Go.

The screenshot shows the 'roles' table in insert mode. The first row has 'id' set to 'int(11)' and 'role' set to 'Admin'. The second row has 'id' set to 'int(11)' and 'role' set to 'Customer'. Both rows have their 'Value' fields circled in red. Below the table is an 'Ignore' checkbox. At the bottom are 'Insert as new row' and 'Go' buttons, with the 'Go' button circled in red.

Рис. 14. Добавление ролей

Загрузите в блокнот файл registration.php и вставьте в него форму регистрации из нашего первого сайта:

```
<h3>Registration Form</h3>
<?php
if(!isset($_POST['regbtn']))
{
?>
<form action="index.php?page=3" method="post">
<div class="form-group">
    <label for="login">Login:</label>
    <input type="text" class="form-control"
           name="login">
</div>
<div class="form-group">
    <label for="pass1">Password:</label>
    <input type="password" class="form-control"
           name="pass1">
</div>
<div class="form-group">
    <label for="pass2">Confirm Password:</label>
    <input type="password" class="form-control"
           name="pass2">
</div>
<div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control"
           name="email">
</div>
<button type="submit" class="btn btn-primary"
           name="regbtn">Register</button>
</form>
<?php
}
else
{
    if(register($_POST['login'], $_POST['pass1'],
    $_POST['email']))
    {
        echo "<h3/><span style='color:green;'>
New User Added!</span><h3/>";
    }
}
?>
```

Активируйте наш сайт и кликните по меню Registration. Вы должны увидеть уже знакомую вам форму:

The screenshot shows a registration form titled "Registration Form". At the top, there is a navigation bar with four items: "Tours", "Comments", "Registration" (which is highlighted in blue), and "Admin Forms". Below the title, there are four input fields labeled "Login:", "Password:", "Confirm Password:", and "Email address:". Each field has a corresponding empty input box below it. At the bottom left of the form area, there is a blue "Register" button. At the very bottom of the page, outside the main form area, is a small copyright notice: "Step Academy ©".

Рис. 15. Форма регистрации

Обратите внимание, что сейчас активированный пункт меню выделяется заливкой. Вы, конечно же, заметили, что в обработчике этой формы указан вызов функции `register()`. Сейчас мы займемся созданием этой функции, которая должна будет заносить новых пользователей в таблицу `Users` нашей БД. Откройте файл `functions.php` и добавьте туда такой код:

```

function register($name,$pass,$email) {
    $name=trim(htmlspecialchars($name));
    $pass=trim(htmlspecialchars($pass));
    $email=trim(htmlspecialchars($email));
    if ($name=="" || $pass=="" || $email=="") {
        echo "<h3/><span style='color:red;'>
                Fill All Required Fields!
            </span><h3/>";
        return false;
    }
    if (strlen($name)<3 || strlen($name)>30 || 
        strlen($pass)<3 ||
        strlen($pass)>30) {
        echo "<h3/><span style='color:red;'>
                Values Length Must Be Between 3
                And 30!
            </span><h3/>";
        return false;
    }
    $ins='insert into users
(login,pass,email,roleid)
values ("' . $name . '", "' .
        md5($pass) . '", "' . $email . '", 2)';
    connect();
    mysql_query($ins);
    $err=mysql_errno();
    if ($err){
        if($err==1062)
            echo "<h3/><span style='color:red;'>
                    This Login Is Already Taken!
            </span><h3/>";
        else
            echo "<h3/><span style='color:red;'>
                    Error code:". $err ."!</
            span><h3/>";
        return false;
    }
    return true;
}

```

В этой функции многое вам уже знакомо, например – валидация данных. Новыми являются только выделенные строки. Поговорим об этих строках.

В переменной \$ins создается SQL запрос добавления новой записи в таблицу Users. Обратите внимание, что мы явно не проверяем уникальность имени входящего пользователя. Это потому, что поле login в таблице Users мы сделали уникальным, переложив, таким образом, заботу об уникальности на MySQL. Машинный труд – на плечи машин. С помощью функции mysql_errno() мы проверяем, как завершился запрос INSERT. Если возникла ошибка, и код этой ошибки 1062 – мы сообщаем пользователю о необходимости заменить login. Если возникла другая ошибка – мы выводим ее код.

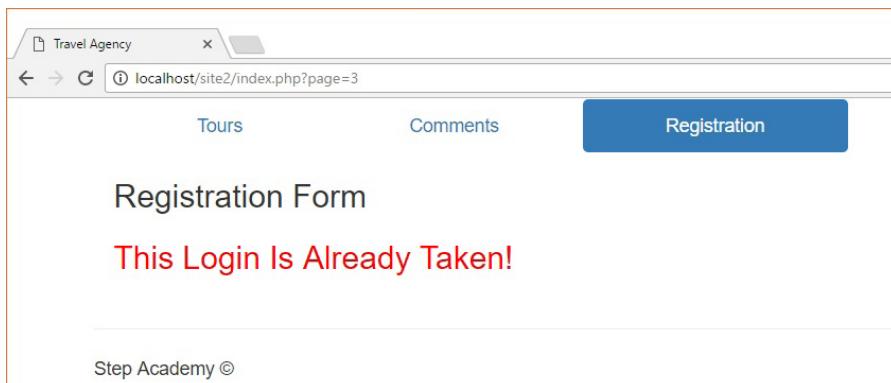


Рис. 16. Нарушение уникальности login

Страница добавления данных

Чтобы заполнять таблицы БД информацией о странах, городах и отелях, на нашем сайте предусмотрена страница AdminForms. К этой странице доступ будет

только у администраторов – у зарегистрированных пользователей с ролью Admin. Переидем к созданию этой страницы и в процессе работы рассмотрим еще несколько особенностей при работе с БД.

Нам надо создать формы для работы с четырьмя таблицами: Countries, Cities, Hotels и Images. Каждая форма должна позволять добавлять и удалять данные из таблицы. Разобьем страницу на две строки. В верхней строке создадим две колонки и разместим там формы для работы с таблицами Countries и Cities. В нижней – расположим формы для работы с таблицами Hotels и Images.

Откройте в блокноте страницу admin.php и занесите туда такую разметку:

```
<div class="row">
    <div class="col-sm-6 col-md-6 col-lg-6 left">
        <!-- section A: for form Countries -->
    </div>
    <div class="col-sm-6 col-md-6 col-lg-6 right">
        <!-- section B: for form Cities -->
    </div>
<hr/>
<div class="row">
    <div class=" col-sm-6 col-md-6 col-lg-6 left ">
        <!-- section C: for form Hotels -->
    </div>
    <div class=" col-sm-6 col-md-6 col-lg-6 right ">
        <!-- section D: for form Images -->
    </div>
</div>
```

Сразу хочу сделать одну важную оговорку. Сейчас мы с вами создадим на одной странице четыре формы. Конечно же, это перебор. Полученная страница не будет образцом стиля. Даже несмотря на использование Bootstrap 4. Тем не менее, мы сделаем эту страницу именно такой по следующим соображениям. Во-первых, с этой страницей будет работать администратор сайта, пользователи ее видеть не будут. А администраторы – люди закаленные и ко всему привычные. Во-вторых, дизайн должны разрабатывать дизайнеры, а не программисты. В-третьих, эта страница просто служит руководством по созданию разных форм и их обработчиков. Рассмотрим процесс создания разметки и PHP кода для каждой из четырех областей, выделенных на этой странице.

Форма добавления и удаления стран

Сначала мы будем выводить в таблице список уже занесенных в БД стран, а под ним – форму для добавления новой страны. В таблице для каждой строки будем выводить чекбокс. Если отметить этот чекбокс для каких-нибудь стран, а потом нажать кнопку Delete под таблицей, то отмеченные страны будут удалены. Возможно, такой интерфейс не является наилучшим с точки зрения юзабилити. Но он является очень полезным с точки зрения обучения. Вы рассмотрите целый ряд действий, которые часто надо выполнять в процессе веб-разработки.

Добавьте в файл admin.php такой код в область A:

```

<!-- section A: Countries form -->
<?php
connect();
$sel='select * from countries';
$res=mysql_query($sel);
echo '<form action="index.php?page=4"
method="post"
class="input-group" id="formcountry">';
echo '<table class="table table-striped">';
while($row=mysql_fetch_array($res,MYSQL_NUM)){
echo '<tr>';
echo '<td>'.$row[0].'</td>';
echo '<td>'.$row[1].'</td>';
echo '<td><input type="checkbox"
name="cb'.$row[0].'"></td>';
echo '</tr>';
}
echo '</table>';
mysql_free_result($res);
echo '<input type="text" name="country"
placeholder="Country">';
echo '<input type="submit" name="addcountry"
value="Add"
class="btn btn-sm btn-info">';
echo '<input type="submit" name="delcountry"
value="Delete"
class="btn btn-sm btn-warning">';
echo '</form>';
?>

```

Рассмотрим этот код. Сначала выполняется запрос 'select * from countries', и по результатам этого запроса строится таблица, в которой будут отображаться уже занесенные в БД страны. Затем создается простая форма добавления новой страны. В этой форме располагаются

две кнопки submit: для добавления новой страны и для удаления выделенных в таблице стран. Обратите внимание, как строится имя для чекбоксов в каждой строке таблицы. Имя каждого чекбокса начинается с символов 'cb', затем к этим символам добавляется id той страны, в строке которой стоит чекбокс. По этим именам с идентификаторами, мы будем узнавать, какие страны надо удалять. Обратите внимание на вызов еще одной функции семейства mysql – mysql_free_result(). Эту функцию надо вызывать для освобождения ресурсов, занимаемых ссылкой на результат выполнения SQL запроса. Когда ресурс вам уже не нужен, вызовите эту функцию и удалите такой ресурс. Обратите внимание на имена кнопок submit. По этим именам мы будем определять обработчики каждой из кнопок. Сохраните добавленный код, запустите наш сайт и активируйте пункт меню Admin Forms. Вы должны увидеть такую форму:

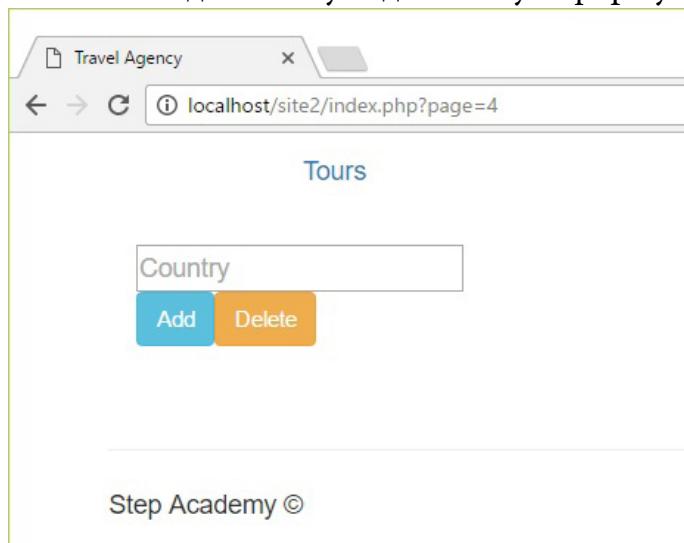


Рис. 17. Форма обработки стран

Теперь добавим к форме два обработчика для двух кнопок submit. Один обработчик будет добавлять новые страны в БД, другой будет удалять выбранные в таблице страны. Сразу после закрытия формы, перед закрытием php секции, вставьте такой код:

```

if (isset($_POST['addcountry'])) {
    $country=trim(htmlspecialchars($_POST['country']));
    if($country=="") exit();
    $ins='insert into countries(country)
          values("'.$country.'")';
    mysql_query($ins);
    echo "<script>";
    echo "window.location=document.URL;";
    echo "</script>";
}
if(isset($_POST['delcountry'])){
    foreach ($_POST as $k => $v) {
        if (substr($k,0,2)=="cb"){
            $idc=substr($k,2);
            $del='delete from countries where
                  id='.$idc;
            mysql_query($del);
        }
    }
    echo "<script>";
    echo "window.location=document.URL;";
    echo "</script>";
}

```

Этот код вам уже знаком. Проверяем, что была нажата кнопка 'addcountry'. Достаем из формы название добавляемой страны и выполняем валидацию. При успешной валидации выполняем SQL запрос INSERT, который добавляет новую страну в таблицу Countries.

Странными могут показаться строки, выводящие в браузер javascript код. Но если вы вспомните курс javascript, то увидите, что эти строки просто выполняют переадресацию на эту же страницу. Это надо для того, чтобы только что добавленная страна появилась в таблице, и вы убедились в том, что добавление выполнено успешно. Возможно, вам интересно, почему мы выполняем переадресацию с помощью javascript? Неужели в PHP нет средств для переадресации? Такие средства есть, но они не всегда являются лучшим выбором. Об этом поговорим немного позже. Добавьте несколько стран:

The screenshot shows a web browser window titled "Travel Agency". The URL in the address bar is "localhost/site2/index.php?page=4". The main content area displays a table with the heading "Tours". The table contains three rows with country names: Argentina, Belgium, and Canada. Each row has a small checkbox next to the country name. Below the table is a form with a text input field labeled "Country" and two buttons: "Add" (blue) and "Delete" (orange). At the bottom of the page, there is a copyright notice: "Step Academy ©".

Рис. 18. Добавление стран

Обработчик удаления намного интереснее предыдущего. Активация любого элемента управления нашей формы приводит к тому, что в массиве `$_POST[]` появляется новый элемент. Если пользователь отметит какой-либо чекбокс, то в массиве `$_POST[]` появится новый элемент с индексом, совпадающим со значением атрибута `name` отмеченного чекбокса. У нас имена всех чекбоксов начинаются со строки 'cb'. Поэтому мы запускаем цикл по ассоциативному массиву `$_POST[]` и ищем в нем индексы со значениями, начинающимися на 'cb'. Каждый такой индекс соответствует отмеченному чекбоксу. При этом мы должны позаботиться о том, чтобы в форме не было других элементов с именами, начинающимися на 'cb'. Если в цикле будет найден такой индекс, мы вырезаем из него подстроку, начинающуюся с третьего символа: `$idc=substr($k,2)`. Вы помните, что с третьего символа в имени чекбокса идет идентификатор соответствующей страны. Забираем полученный идентификатор в переменную `$idc` и выполняем SQL запрос `DELETE`. Проверьте, как это работает (Рис. 19).

Форма для обслуживания стран готова. Перейдем к созданию похожей формы, предназначенной для добавления и удаления городов.

The screenshot shows a web browser window titled "Travel Agency". The address bar displays the URL "localhost/site2/index.php?page=4". The main content area is titled "Tours" and lists four items:

	Country	Action
7	Argentina	<input type="checkbox"/>
8	Belgium	<input type="checkbox"/>
9	Canada	<input type="checkbox"/>
10	Narnia	<input checked="" type="checkbox"/>

Below the table is a search bar with the placeholder "Country" and two buttons: "Add" (blue) and "Delete" (orange).

At the bottom of the page, there is a copyright notice: "Step Academy ©".

Рис. 19. Удаление страны

Форма добавления и удаления городов

Эта форма будет отличаться от предыдущей наличием списка стран, занесенных в таблицу БД. Добавляя новый город, надо будет указывать, в какой стране этот город находится. Как и в предыдущем случае, будем выводить таблицу с уже добавленными в БД городами. В этой таблице будем указывать и название страны и название города, т.к. в разных странах есть города с одинаковыми названиями. Поэтому для этой таблицы мы будем использовать многотабличный запрос. Ранее мы

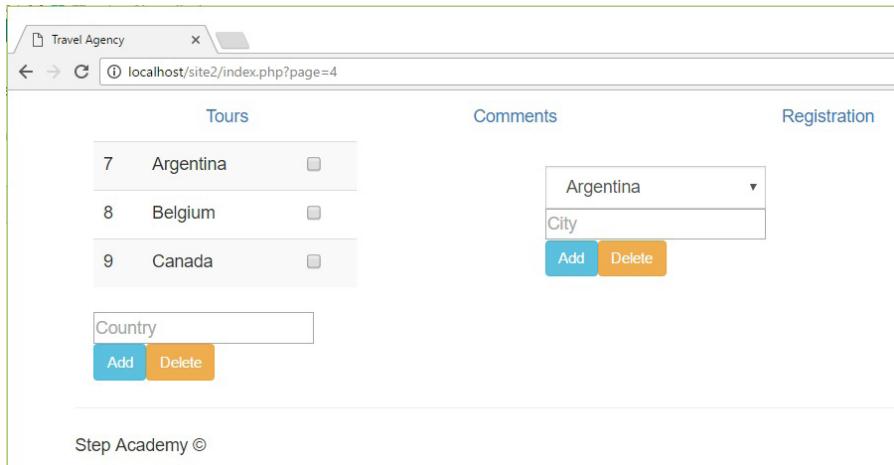
уже создавали динамический список. Сейчас рассмотрим, как создавать список на основе данных из таблицы БД.

Помните, мы разделили страницу admin.php на области: А, В, С и D? Предыдущая форма располагалась в области А. Сейчас будем работать в области В. Добавьте туда такой код для создания формы:

```
<!-- section B: Cities form -->
<?php
echo '<form action="index.php?page=4"
method="post"
class="input-group" id="formcity">';
$sel='SELECT ci.id, ci.city, co.country
from countries co, cities ci
WHERE ci.countryid=co.id';
$res=mysql_query($sel);
echo '<table class="table table-striped">';
while ($row=mysql_fetch_array($res,MYSQL_NUM))
{
    echo '<tr>';
    echo '<td>'.$row[0].'</td>';
    echo '<td>'.$row[1].'</td>';
    echo '<td>'.$row[2].'</td>';
    echo '<td><input type="checkbox"
name="ci'.$row[0].'"></td>';
    echo '</tr>';
}
echo '</table>';
mysql_free_result($res);
$res=mysql_query('select * from countries');
echo '<select name="countryname" class="form-
control">';
while ($row=mysql_fetch_array($res,MYSQL_NUM))
{
```

```
        echo '<option value="'. $row[0].'">' .  
              $row[1] . '</option>';  
    }  
echo '</select>';  
echo '<input type="text" name="city"  
                  placeholder="City">';  
echo '<input type="submit" name="addcity"  
                  value="Add"  
class="btn btn-sm btn-info">';  
echo '<input type="submit" name="delcity"  
                  value="Delete"  
class="btn btn-sm btn-warning">';  
echo '</form>';  
  
?>
```

Проверьте, как выглядит созданная форма:



The screenshot shows a web application interface for managing cities. The main content area displays a table of cities with checkboxes. To the right, there's a dropdown menu and an input field for adding new cities. Below the table, there are 'Add' and 'Delete' buttons. A separate section at the bottom left also features an 'Add' and 'Delete' button pair.

Рис. 20. Форма добавления городов

Обсудим приведенный код. Прежде всего, в глаза бросается многотабличный запрос. Этот запрос нужен для создания таблицы городов. Мы извлекаем из та-

блиц Cities и Countries три поля: идентификатор города, название страны и название города, и строим таблицу из четырех колонок. В первых трех колонках таблицы располагаются эти три поля. А в четвертой колонке мы снова создаем чекбоксы для удаления городов. Опять обратите внимание на имена создаваемых чекбоксов. Строки таблицы создаются в цикле, который вызывает функцию mysql_fetch_array().

Затем выполняется еще один простой запрос SELECT для создания списка городов. Извлекаемые в этом запросе идентификаторы пойдут в атрибуты value элементов option, а названия городов – пойдут в сами элементы option. Сам список снова строится в цикле, вызывающем функцию mysql_fetch_array().

После закрытия этой формы добавим код обработчиков:

```
if(isset($_POST['addcity'])){
    $city=trim(htmlspecialchars($_POST['city']));
    if ($city=="") exit();
    $countryid=$_POST['countryname'];
    $ins='insert into cities (city,countryid) values
        ("'.$city.'",'.$countryid.');
    mysql_query($ins);
    $err=mysql_errno();
    if ($err){
        echo 'Error code:'.$err.'<br>';
        exit();
    }
    echo "<script>";
    echo "window.location=document.URL;";
    echo "</script>";
}
```

```

if(isset($_POST['delcity'])){
    foreach ($_POST as $k => $v) {
        if (substr($k,0,2)=="ci"){
            $idc=substr($k,2);
            $del='delete from cities where
            id='.$idc;
            mysql_query($del);
        }
    }
    echo "<script>";
    echo "window.location=document.URL;";
    echo "</script>";
}

```

В этих двух обработчиках ничего нового для вас нет. Добавьте несколько городов, чтобы проверить работу созданной формы:

The screenshot shows a web page with a table of cities and a form for adding new cities.

Table Headers:

- Tours
- Comments
- Registration

Data in Table:

7	Argentina	<input type="checkbox"/>	2	Charleroi	Belgium <input type="checkbox"/>
8	Belgium	<input type="checkbox"/>	3	Ghent	Belgium <input type="checkbox"/>
9	Canada	<input type="checkbox"/>	1	Toronto	Canada <input type="checkbox"/>

Form Elements:

- Country:
- Add:
- Delete:
- City:
- Add:
- Delete:

Step Academy ©

Рис. 21. Добавление городов

Аналогичным образом проверьте удаление городов:

The screenshot shows a web page with three main sections: 'Tours', 'Comments', and 'Registration'. The 'Tours' section lists cities with checkboxes for deletion. The 'Comments' section lists cities with checkboxes for deletion. The 'Registration' section lists cities with checkboxes for deletion. Below these sections are two input fields: 'Country' and 'City', each with an 'Add' button and a 'Delete' button.

	Tours	Comments	Registration
7	Argentina	<input type="checkbox"/>	4 Parador Argentina <input checked="" type="checkbox"/>
8	Belgium	<input type="checkbox"/>	2 Charleroi Belgium <input type="checkbox"/>
9	Canada	<input type="checkbox"/>	3 Ghent Belgium <input type="checkbox"/>
Country		City	
<input type="text"/> Add		<input type="text"/> Argentina <input type="button" value="Delete"/>	
<input type="text"/> Delete			

Рис. 22. Удаление городов

Продолжим работу с этой страницей.

Форма добавления и удаления отелей

Перейдем к созданию формы, позволяющей администратору сайта добавлять в БД информацию об отелях. Эта форма будет немного сложнее двух предыдущих, но принципиально нового в ней ничего не будет. Перейдите в область С нашей страницы и добавьте такой код для создания формы ввода информации об отелях:

```

<?php
echo '<form action="index.php?page=4"
        method="post"
        class="input-group" id="formhotel">';
$sel='SELECT ci.id, ci.city,
        ho.id, ho.hotel, ho.cityid, ho.countryid,
                                ho.stars, ho.info,
        co.id, co.country
        from cities ci, hotels ho, countries co
        WHERE ho.cityid=ci.id and ho.countryid=co.id';
$res=mysql_query($sel);
$err=mysql_errno();
echo '<table class="table" width="100%">';
while ($row=mysql_fetch_array($res,MYSQL_NUM))
{
    echo '<tr>';

    echo '<td>'.$row[2].'</td>';
    echo '<td>'.$row[1]."-".$row[9].'</td>';
    echo '<td>'.$row[3].'</td>';
    echo '<td>'.$row[6].'</td>';
    echo '<td><input type="checkbox"
                name="hb'.$row[2].'"></td>';
    echo '</tr>';
}
echo '</table>';
mysql_free_result($res);

$sel='SELECT ci.id, ci.city, co.country, co.id
        from countries co, cities ci
        WHERE ci.countryid=co.id';
$res=mysql_query($sel);
$csel=array();
echo '<select name="hcity" class="">';
while ($row=mysql_fetch_array($res,MYSQL_NUM))
{

```

```
echo '<option value="'. $row[0] .'">'. $row[1]."  
      : ". $row[2] . '</option>';  
$csel[$row[0]] = $row[3];  
}  
echo '</select>';  
echo '<input type="text" name="hotel"  
          placeholder="Hotel">';  
echo '<input type="text" name="cost"  
          placeholder="Cost">';  
echo '&nbsp;&nbsp;Stars: <input type="number"  
          name="stars"  
          min="1" max="5">';  
echo '<br><textarea name="info"  
          placeholder="Description">';  
echo '</textarea><br>';  
echo '<input type="submit" name="addhotel"  
          value="Добавить"  
          class="btn btn-sm btn-info">';  
echo '<input type="submit" name="delhotel"  
          value="Удалить"  
          class="btn btn-sm btn-warning">';  
echo '</form>';  
mysql_free_result($res);  
?>
```

Сначала мы выводим таблицу с уже добавленными в БД отелями, а затем создаем форму. Проверьте, как это выглядит:

Урок № 3

Tours	Comments	Registration
7 Argentina	2 Charleroi	Belgium
8 Belgium	3 Ghent	Belgium
9 Canada	1 Toronto	Canada

Country: City:

Add Delete

Charleroi : Belgium Hotel

Cost: Stars:
Description:

добавить удалить

Рис. 23. Форма для работы с отелями

Теперь добавим обработчики:

```
if(isset($_POST['addhotel'])){
    $hotel=trim(htmlspecialchars($_POST['hotel']));
    $cost=intval(trim(htmlspecialchars($_POST['cost'])));
    $stars=intval($_POST['stars']);
    $info=trim(htmlspecialchars($_POST['info']));
    if ($hotel=="'||$cost=="'||$stars=="")
        exit();
    $cityid=$_POST['hcity'];
    $countryid=$csel[$cityid];
    $ins='insert into hotels (hotel,cityid,coun
tryid,stars,cost,info)
values("'.$hotel.'",'.$cityid.';
$ins.=',"'.$countryid.','.$stars.','.$cost.', "'.$info.'
```

```

$ins.="";
mysql _ query($ins);
echo "<script>";
echo "window.location=document.URL;";
echo "</script>";
}
if(isset($_ POST['delhotel'])){
foreach ($_ POST as $k => $v) {
if (substr($k,0,2)==="hb"){
$idc=substr($k,2);
$del='delete from hotels where
id='.$idc;
mysql _ query($del);
if ($err){
echo 'Error code:'.$err.'<br>';
exit();
}
}
}
echo "<script>";
echo "window.location=document.URL;";
echo "</script>";
}
}

```

Снова проверим работу созданной формы:

1	Toronto-Canada	Hilton 	5	<input type="checkbox"/>
2	Ghent-Belgium	Langham Hotel	5	<input type="checkbox"/>
<input type="text" value="Charleroi : Belgium ▾"/> <input type="text" value="Hotel"/> <input type="text" value="Cost"/> Stars: <input type="text"/> <input type="text" value="Description"/> <input type="button" value="добавить"/> <input type="button" value="удалить"/>				

Рис. 24. Добавление отелей

Форма добавления картинок для отелей

На этой странице осталось выполнить последнюю часть работы – создать форму добавления картинок для отелей в области D. Эта форма будет позволять пользователю выбирать из списка определенный отель и добавлять к описанию этого отеля произвольное число картинок. Картинки будут храниться в папке images нашего сайта, а пути к картинкам – в таблице Images в поле imagepath. Мы создадим multiple upload, позволив пользователю за одну выгрузку передать на сервер несколько картинок. Когда вы разрешаете пользователю выполнять upload, вы обязательно должны обеспечить контроль размеров передаваемых файлов. Контроль надо выполнять и на стороне клиента и на стороне сервера. На стороне сервера ограничения размеров передаваемых файлов указываются в файле php.ini. В атрибуте upload_max_filesize указывается максимальный допустимый размер одного выгружаемого файла. В атрибуте max_file_uploads указывается максимальное количество файлов, допустимое для выгрузки за один раз. Допустим, у вас в php.ini установлены такие значения:

upload_max_filesize = 16Mb

max_file_uploads = 10

Значит ли это, что вы можете за один раз выгрузить на сервер 10 файлов, размером 16 Mb каждый? Ответить на этот вопрос, располагая только указанными двумя значениями невозможно. Дело в том, что существует еще один атрибут, который имеет отношение к ограничению размеров – post_max_size. Этот атрибут устанавливает максимальный размер post пакета, пересылаемо-

го за один раз. Если в нем указано значение 160M, то вы сможете выгрузить 10 файлов по 16M каждый, за один раз. Если же в post_max_size указано значение, например, 80M, то вы сможете выгрузить либо 10 файлов по 8M, либо 5 файлов по 16M, либо в другом наборе, но общим размером не больше 80M.

Вы уже привыкли, что у тега form надо указывать два атрибута: action и method. Однако не забывайте, что в некоторых ситуациях у тега форм надо обязательно указывать еще один атрибут. Это атрибут enctype со значением «multipart/form-data». Этот атрибут обязательно надо указывать в тех формах, где вы планируете использовать upload. У нас именно такая форма. Добавим в область D такую разметку:

```
<!-- section D: Images form -->
<?php
echo '<form action="index.php?page=4"
method="post"
enctype="multipart/form-data"
class="input-group">';
echo '<select name="hotelid">';
$sel='select ho.id, co.country,ci.city,ho.hotel
from countries co,cities ci, hotels ho
where co.id=ho.countryid and ci.id=ho.cityid
order by co.country';
$res=mysql_query($sel);
while($row=mysql_fetch_array($res,MYSQL_NUM)){
    echo '<option value="'.$row[0].'">';
    echo $row[1].' &nbsp;'.$row[2].'&nbsp;&n
bsp;'.$row[3].
'</option>';
}
}
```

```

mysql _ free _ result($res);
echo '<input type="file" name="file[]" multiple
accept="image/*">';
echo '<input type="submit" name="addimage"
value="Add"
class="btn btn-sm btn-info">';
echo '</select>';
echo '</form>';
if(isset($_ REQUEST['addimage'])){
    foreach($_ FILES['file']['name'] as $k => $v)
{
        if($_ FILES['file']['error'][$k] !=0){
            echo '<script>alert("Upload
file error:'.$v.'")</script>';
            continue;
        }
        if(move_uploaded_file($_ FILES['file']
['tmp_name'][$k],
'images/'.$v)){
            $ins='insert into
images(hotelid,imagepath)
values('.$_ REQUEST['hotelid'].',"images/'.$v.')';
            mysql _ query($ins);
        }
    }
}
echo '</div>';
?>
```

У элемента input с типом file, который предназначен для выполнения многофайловой выгрузки, имя должно описываться в виде массива, в нашем случае – file[]. Это необходимо потому, что в этом случае такой элемент input является контейнером для нескольких файлов. Кроме того, у такого input должен присутствовать

атрибут multiple. В обработчике массив \$_FILES теперь будет не двумерным, а трехмерным, потому что каждый элемент этого массива теперь хранит информацию о нескольких выбранных файлах. Например, \$_FILES[«file»] [«name»] теперь будет содержать имена всех выбранных пользователем файлов. Снова вспоминаем, что массив \$_FILES является ассоциативным и перебираем его элементы в цикле foreach. Вызовом функции move_uploaded_file() переносим каждый выгруженный файл из временной папки в папку images нашего сайта и с помощью SQL запроса INSERT добавляем путь к каждому из этих файлов в таблицу Images.

Вот теперь мы с вами закончили создание самой трудоемкой страницы нашего сайта. Занесите в БД несколько стран, городов и отелей, чтобы у нас была тестовая информация для дальнейшей работы. Конечно же, добавьте к отелям картинки. Переходим к созданию других страниц.

Страница выбора тура

Начнем создавать страницу для первого пункта меню нашего сайта. План у нас такой. Создаем на странице список со странами из нашей БД. Рядом создаем список с городами. Когда пользователь выберет страну и город на этой странице, мы сформируем таблицу, в которой выведем краткое описание всех отелей, расположенных в выбранном городе. В этой таблице одно поле будет кликабельным. Когда пользователь кликнет по такому полю, в соседней вкладке браузера откроется новая страница, содержащая подробное описание выбранного отеля с галереей из всех картинок.

Сейчас мы столкнемся с очень интересной проблемой. Нам надо организовать работу с двумя связанными списками: списком стран и списком городов. Конечно же, будет правильно сделать так, чтобы после выбора страны во втором списке отображались только города для выбранной страны. Мы это сделаем, но не очень красиво. Красиво мы сможем организовать взаимодействие этих двух списков, когда научимся работать с Ajax. Тогда мы обязательно вернемся к этой странице и переделаем ее. А пока мы сделаем для фиксации выбора в каждом списке две кнопки submit. Вы должны понимать: для того, чтобы сервер узнал, какая страна выбрана в первом списке, мы должны будем отправить ему этот выбор в запросе. Узнав о выбранной стране, сервер сможет прислать нам только города этой страны для второго списка. Затем, таким же образом, мы сообщим серверу, какой город мы выбрали, и он пришлет нам список отелей для этого города.

Откройте страницу tours.php и добавьте в нее такой код:

```

<h2>Select Tours</h2>
<hr>
<?php
connect();

echo '<form action="index.php?page=1" method="post">';
echo '<select name="countryid" class="col-sm-3 col-md-3
          col-lg-3">';

$res=mysql _ query("SELECT * FROM countries ORDER BY
                      country");
echo '<option value="0">Select country...</option>';
while ($row=mysql _ fetch _ array($res, MYSQL _ NUM)) {
    echo '<option value="'. $row[0].'">'. $row[1].'
</option>';
}
mysql _ free _ result($res);
echo '<input type="submit" name="selcountry"
           value="Select Country"
           class="btn btn-xs btn-primary">';
echo '</select>';

```

Проверьте, как выглядит созданная форма в браузере:

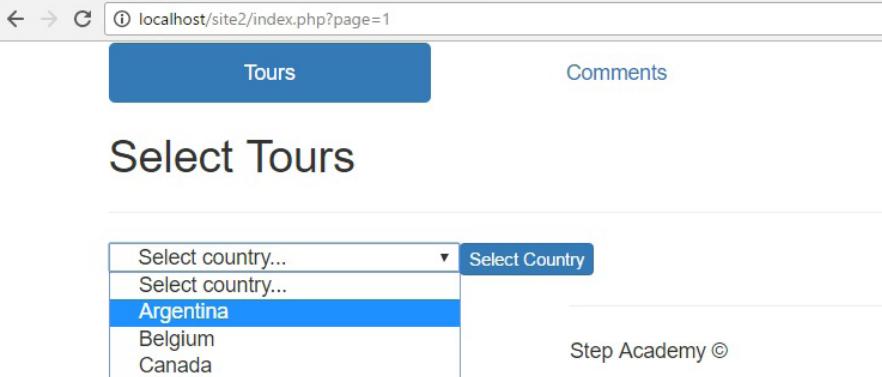


Рис. 25. Форма выбора тура (начало)

Обратите внимание, что в приведенном коде форма еще не закрыта, там не указан тег `</form>`. Это сделано вот по какой причине. Когда пользователь выберет страну и нажмет кнопку `submit`, начнет работать обработчик. Обработчик получит из формы идентификатор выбранной страны. Обратите внимание, что у наших `option` указан атрибут `value`, поэтому они отображают названия стран, но при выборе возвращают `id` выбранной страны из `value`. Так вот, обработчик получит идентификатор выбранной страны, с помощью запроса `SELECT` достанет из БД перечень городов для этой страны и вернет нам на страницу, в эту же форму, еще один список – список городов. Нам надо, чтобы этот список с городами оказался в этой же форме, поэтому мы разместим тег `</form>` после обработчика. Посмотрите, как это выглядит в коде:

localhost/site2/index.php?page=1

Tours Comments

Select Tours

Select country... ▾

Select city... ▾

Select city...

Charleroi

Ghent

Select Country

Select City

Step Academy ©

Рис. 26. Форма выбора тура (продолжение)

```

if(isset($_POST['selcountry']))
{
    echo '<br/>';
    $countryid=$_POST['countryid'];
    if($countryid == 0) exit();
    $result=mysql_query(
        "SELECT * FROM cities where countryid="
            ."$countryid." ORDER BY city");
    echo '<select name="cityid" class="col-sm-3
        col-md-3 col-lg-3">';
    echo '<option value="0">Select city...</option>';
    while ($row=mysql_fetch_array($result,
        MYSQL_NUM)) {
        echo '<option value="'. $row[0] .'">' .
            $row[1]. '</option>';
    }
    mysql_free_result($result);
    echo '</select>';
    echo '<input type="submit" name="selcity"
        value="Select City"
        class="btn btn-xs btn-primary">';
}
echo '</form>';

```

Вы видите, что мы сформировали запрос:

```
"SELECT * FROM cities where countryid="
    ."$countryid." ORDER BY city");
```

в который вставили идентификатор выбранной страны из формы. Выберите какую-нибудь страну и нажмите кнопку submit. В вашей форме должен появиться список с городами для выбранной страны (Рис. 26).

Вы видите, что рядом со списком городов мы создали еще одну кнопку submit, с помощью которой мы от-

правим на сервер информацию о том, какой город мы выбрали. Для этой кнопки будет еще один обработчик, в котором мы создадим таблицу с перечнем отелей. Добавьте этот обработчик сразу после закрытия формы:

```
if(isset($_POST['selcity']))
{
    $cityid=$_POST['cityid'];
    $sel='SELECT co.country, ci.city, ho.hotel,
           ho.cost, ho.stars, ho.id
      FROM hotels ho, cities ci, countries co
     WHERE ho.cityid=ci.id
       AND ho.countryid=co.id
       AND ho.cityid='.$cityid;
    $res=mysql_query($sel);
    echo '<table width="100%" class="table-
           striped tbtourz text-center">';
    echo '<thead style="font-weight: bold">
<td>Hotel</td><td>Country</td><td>City</td>
<td>Price</td><td>Stars</td><td>link</td></thead>';
    while ($row=mysql_fetch_array($res,MYSQL_NUM))
    {
        echo '<tr id="'.$row[1].'">';
        echo '<td>'.$row[2].'</td>
<td>'.$row[0].'</td>
<td>'.$row[1].'</td>
        <td>'.$row[3].'</td><td>'.$row[4].'</td>
        <td <a href="pages/hotelinfo.php?hotel='.$row[5].
        '" target="_blank"> more info</a></td>';
        echo '</tr>';
    }
    echo '</table><br>';
}
```

Здесь есть о чём поговорить. Сначала мы выполняем многотабличный запрос, чтобы помимо информации об отеле получить еще названия страны и города. Об-

ратите внимание на три условия в секции WHERE. Вы должны помнить, что в многотабличном запросе надо указывать условия для отсечения декартовых множеств. Количество таких условий должно быть на единицу меньше, чем количество таблиц в запросе. В нашем запросе три таблицы, поэтому указано два таких условия: ho.cityid=ci.id и ho.countryid=co.id. Третье условие (ho.cityid='\$.cityid) необходимо для того, чтобы запрос вернул только отели для выбранного города. По результатам этого запроса строится таблица. Самое интересное в ней – последняя колонка. Сейчас поговорим об этой колонке, но сначала давайте посмотрим на созданную таблицу:

Select Tours

Hotel	Country	City	Price	Stars	link
Park Plaza Westminster Bridge London	United Kingdom	London	\$250	4	more info
The Langham London	United Kingdom	London	\$340	5	more info

Рис. 27. Таблица отелей в выбранном городе

В этой таблице выводится только некоторая информация об отелях. Чтобы пользователь мог посмотреть подробное описание каждого отеля, ему надо будет кликнуть по ссылке more info для выбранного отеля. Что при этом произойдет? Давайте внимательно посмотрим на код ссылки. Вы уже и сами можете все понять по этому коду:

```
<a href="pages/hotelinfo.php?hotel='$.row[5].'"  
target=" _ blank">more info</a>
```

Эта ссылка переведет пользователя на файл с именем hotelinfo.php, расположенный в папке pages. Кроме того, этому файлу с помощью метода GET будет передано значение с именем hotel, в которое будет занесен идентификатор выбранного отеля. Указанная страница hotelinfo.php служит шаблоном для полного описания отеля. На ней мы сможем получить из элемента `$_GET['hotel']` идентификатор отеля, а по идентификатору получим всю информацию о самом отеле. Если кратко отобразить суть происходящего на странице hotelinfo.php, это будет выглядеть так:

- получить id отеля:
`$hotelid=$_GET['hotel'];`
- выполнить запрос
`'SELECT * FROM hotels WHERE id=' . $hotelid;`
- красиво расположить на странице информацию об отеле, полученную из этого запроса;
- выполнить запрос:
`'SELECT * FROM images WHERE hotelid=' . $hotelid;`
- создать из картинок, полученных из этого запроса, красивую галерею.

Этой страницы у нас пока нет, и сейчас мы начнем ее создавать.

Страница описания отеля

Все наши страницы из папки pages подключаются с помощью функции `include_once()` в страницу index.php. Поэтому ни одна из этих страниц (tours.php, admin.php и др.) не содержит структуры HTML страницы – там нет тегов `<html>`, `<head>`, `<body>` и прочих. Они вписаны

ваются в структуру файла index.php, у которого все эти теги, конечно же, есть. Со страницей hotelinfo.php, которую мы сейчас собираемся создавать, ситуация другая. Эта страница явно вызывается ссылкой и не имеет никакого отношения к index.php. Поэтому она должна содержать свою разметку HTML страницы. Создайте в папке pages файл hotelinfo.php и добавьте в него такую разметку:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Hotel Info</title>
    <link rel="stylesheet" href="../css/bootstrap.
                                min.css">
    <link rel="stylesheet" href="../css/info.css">
</head>

<body>
<?php
include _ once ("functions.php");
if(isset($ _ GET['hotel'])){
    $hotel=$ _ GET['hotel'];
    connect();
    $sel='select * from hotels where id='.$hotel;
    $res=mysql _ query($sel);
    $row=mysql _ fetch _ array($res,MYSQL _ NUM);
    $hname=$row[1];
    $hstars=$row[4];
    $hcost=$row[5];
    $hinfo=$row[6];
    mysql _ free _ result($res);

    echo '<h2 class="text-uppercase text-
center">'.$hname.'</h2>';

}

```

```
echo '<div class="row"><div class="col-md-6 text-center">';

connect();
$sel='select imagepath from images where
          hotelid='.$hotel;
$res=mysql_query($sel);

echo '<span class="label label-info">Watch our
                  pictures</span>';
echo'<ul id="gallery">';
$i=0;
while($row=mysql_fetch_array($res,MYSQL_NUM)){
    echo ' <li></li>';
}
mysql_free_result($res);
echo ' </ul>';
```

Посмотрите, как в функции include_once() указывается путь к файлу functions.php. Он указывается без папки pages. Это потому, что сейчас наш файл hotelinfo.php располагается рядом с файлом functions.php, в одной папке. Обратите внимание, что результат выполнения первого запроса SELECT мы обрабатываем без уже привычного цикла while. Догадались, почему это возможно? Потому что запрос выбирает по уникальному значению id и не может вернуть более одной строки – экономим на цикле.

Как можно описать «звездность» отеля? Поле stars в таблице имеет тип int. Мы получаем значение из этого поля и рисуем в цикле звездочку соответствующее количество раз:

```
for ($i=0; $i<$hstars; $i++)
{
    echo '<image src="../images/star.png" alt="star">';
}
```

Посмотрите, как выглядит эта страница в моем случае:



Рис. 28. Страница описания отеля

Там еще картинки в слайдере сами перелистывают-ся. Но у вас сейчас такой красоты не будет. Этот код у вас еще не завершен. А что дальше? Давайте сделаем небольшое отступление.

Я знаю, что многие программисты пренебрежительно относятся к внешнему виду своих приложений. Они считают лишним тратить усилия на красивое оформление, полагая, что это ниже их профессионального достоинства. Оправдание у этих исполнителей такое: «Главное – чтобы программа работала, а как она вы-

глядит – не важно». Такой подход неверен. Более того, в веб-разработке такой подход вообще недопустим. Каждый программист должен получать удовольствие от своей работы, чтобы ему самому нравился результат – созданное приложение, чтобы хотелось улучшать и улучшать его до бесконечности. Так вот, предлагаю вам достичь такого состояния и творчески отнестись к созданию этой страницы. Все возможности для этого есть. Используйте готовые галереи, используйте jQuery, используйте Bootstrap – и создайте собственные шедевры. Поразите своих преподавателей и коллег-студентов, да и самих себя тоже.

Аутентификация и авторизация на сайте

Мы создали форму регистрации и функцию, облучивающую процесс регистрации. У нас есть таблица ролей. Давайте доведем этот механизм до завершения: добавим на сайте возможность входа для зарегистрированных пользователей и определение ролей входящих пользователей. На главной странице нашего сайта создан раздел header, который пока не используется. В нем мы создадим форму входа. Сделаем все это снова в отдельном файле login.php и подключим этот файл в раздел header функцией include_once():

```
<header class="col-sm-12 col-md-12 col-lg-12">
    <?php include_once ("pages/login.php") ; ?>
</header>
```

Файл login.php может выглядеть так:

```

<?php
if (isset($_SESSION['ruser']))
{
    echo '<form action="index.php';
    if (isset($_GET['page'])) echo '?page='.$_
                                $_GET['page'];
    echo '" class="form-inline pull-right"
                      method="post">';
    echo '<h4>Hello, <span>' . $_SESSION['ruser'] . '</
                                span>&nbsp;';
    echo '<input type="submit" value="Logout"
                      id="ex" name="ex"
class="btn btn-default btn-xs"></h4>';
    echo '</form>';
    if (isset($_POST['ex']))
    {
        unset($_SESSION['ruser']);
        unset($_SESSION['radmin']);
        echo '<script>window.location.reload()</
                        script>';
    }
}
else
{
    if (isset($_POST['press']))
    {
        if(login($_POST['login'], $_POST['pass']))
        {
            echo '<script>window.location.
                           reload()</script>';
        }
    }
    else
    {
        echo '<form action="index.php';
        if (isset($_GET['page'])) echo
                                '?page='.$_GET['page'];

```

```
        echo '' class="input-group input-group-sm
                  pull-right"
method="post">';
        echo '<input type="text" name="login"
                  size="10" class=""
                  placeholder="login">
<input type="password" name="pass"
                  size="10" class=""
                  placeholder="password">
<input type="submit" id="press"
                  value="Login"
class="btn btn-default btn-xs" name="press">
        </form>';
    }
}
```

Этот файл создает в разделе header либо форму входа, если на сайт еще не вошел зарегистрированный пользователь, либо форму выхода, если такой пользователь уже вошел. Для обработки входа используется функция login(), которую нам предстоит написать. Эта функция при удачном входе пользователя возвращает true и при этом еще заносит в сессию в переменную \$_SESSION['ruser'] имя вошедшего пользователя. При входе в систему администратора эта функция создает в сессии еще одну переменную \$_SESSION['radmin'], куда заносит имя администратора. При выходе зарегистрированного пользователя эти переменные в сессии удаляются. У меня созданная форма выглядит так в правом верхнем углу страницы:



Select Tours

Select country...

Step Academy ©

Рис. 29. Форма входа

Теперь надо добавить в файл functions.php функцию login():

```
function login($name,$pass)
{
    $name=trim(htmlspecialchars($name));
    $pass=trim(htmlspecialchars($pass));
    if ($name=="" || $pass=="")
    {
        echo "<h3/><span style='color:red;'>
Fill All Required Fields!</span><h3/>";
        return false;
    }
    if (strlen($name)<3 || strlen($name)>30 ||
    strlen($pass)<3 || strlen($pass)>30) {
        echo "<h3/><span style='color:red;'>
Value Length Must Be Between 3 And 30!
</span><h3/>";
        return false;
    }
    connect();
    $sel='select * from users where
                    login="'. $name .'"
and pass="'. md5($pass) . '"';
    $res=mysql_query($sel);
    if($row=mysql_fetch_array($res,MYSQL_NUM)) {
        $_SESSION['ruser']=$name;
        if($row[5]==1)
        {
    }
```

```

        $_SESSION['radmin']=$name;
    }
    return true;
}
else
{
    echo "<h3/><span style='color:red;'>
No Such User!</span><h3/>";
    return false;
}
}

```

В этой функции вам все должно быть понятно. Она подобна нашей функции register(). Еще раз обратите внимание, что при удачном входе пользователя функция открывает сессию и в переменную \$_SESSION['ruser'] заносит имя вошедшего пользователя. При входе в систему администратора эта функция создает в сессии еще одну переменную \$_SESSION['radmin'], куда заносит имя администратора. Зачем надо фиксировать вход администратора? Чтобы открыть доступ к странице admin.php только для администраторов. Добавьте на страницу admin.php в самое начало кода такую проверку:

```

if (!isset($_SESSION['radmin']))
{
    echo "<h3/><span style='color:red;'>For
Administrators Only!
</span><h3/>";
    exit();
}

```

Попробуйте войти сейчас на страницу admin.php, и вы увидите такое сообщение:



For Administrators Only!

Рис. 30. Несанкционированный доступ

Зарегистрируйте на сайте нового пользователя – например, с именем traveler. Давайте сделаем этого пользователя администратором. Для этого надо сделать так, чтобы в поле roleid для этого пользователя стояло значение 1. Вы должны помнить, что в таблице Roles у нас есть две роли: Admin с идентификатором 1 и Customer с идентификатором 2. Все зарегистрированные пользователи по умолчанию получают роль Customer. Мы говорили, что менять роли может только администратор, но не создали для администратора такого интерфейса. Пока у нас нет способа создавать администраторов сайта, сменить роль для пользователя traveler нам поможет PhpMyAdmin. Запустите его и в левой панели выделите таблицу Users. Затем нажмите кнопку редактирования в строке для пользователя traveler:

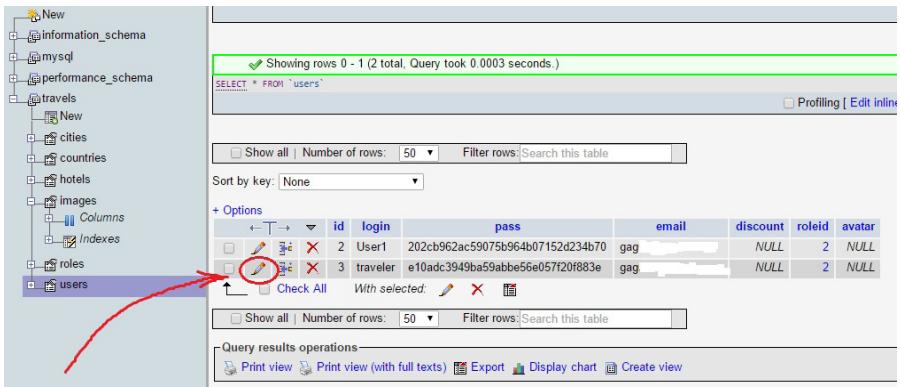


Рис. 31. Изменение роли

Откроется окно редактирования. Измените значение поля roleid на 1 и нажмите кнопку Go.

Column	Type	Function	Null	Value
id	int(11)			3
login	varchar(32)		<input type="checkbox"/>	traveler
pass	varchar(128)		<input type="checkbox"/>	e10adc3949ba59abbe56e057f20f883e
email	varchar(128)		<input type="checkbox"/>	gag
discount	int(11)		<input checked="" type="checkbox"/>	
roleid	int(11)		<input checked="" type="checkbox"/>	1
avatar	mediumblob		<input checked="" type="checkbox"/>	Binary - do not edit (0 B) Or web server upload directory: (Max: 16MB)

Рис. 32. Изменение роли (продолжение)

Теперь войдите как traveler, и вы получите доступ к странице admin.php.

Tours	Comments	Registration	Admin Forms
7 Argentina		2 Charleroi Belgium	
8 Belgium		3 Ghent Belgium	
9 Canada		1 Toronto Canada	
10 United Kingdom		4 London United Kingdom	

Country: Argentina

Add Delete

City: Charleroi

Add Delete

Рис. 33. Вход администратора

Создание администраторов сайта

Мы уже говорили о том, что на нашем сайте нет механизма создания администраторов сайта. Страница регистрации создает пользователей сайта с ролью

Customer. Чтобы создать для отладочных целей пользователя с ролью Admin, мы использовали прямое изменение записи в таблице БД с помощью PhpMyAdmin. Это не правильно. Поэтому сейчас мы создадим механизм изменения роли по умолчанию (Customer) на роль Admin. Создание этого механизма продемонстрирует вам два новых полезных навыка: скрытие пункта меню и работу с полями таблицы типа blob. Помните наш пункт меню Admin Forms? К нему нет доступа у пользователей с ролью Customer. Но этот пункт меню виден всем. Сейчас мы сделаем новый пункт меню, который будет виден и доступен только администраторам. Назовем этот пункт меню Private и создадим для него в папке pages файл private.php. В этом файле будем выводить список пользователей с ролью Customer, чтобы им можно было изменять роль пользователя со значения Customer на Admin. Кроме того, эта форма будет позволять добавлять портрет администратора, который будет храниться в таблице Users в поле avatar в виде бинарного контента. Приступим. Откройте файл и добавьте в него такой код:

```
<?php
connect();
echo '<form action="index.php?page=5" method="post"
      enctype="multipart/form-data" class="input-
      group">';
echo '<select name="userid">';
$sel='select * from users where roleid=2 order by
                  login';
$res=mysql_query($sel);
while($row=mysql_fetch_array($res,MYSQL_NUM)) {
    echo '<option value="'. $row[0] .'">' . $row[1] . '</
option>';
}
```

```

mysql_free_result($res);
echo '</select>';
echo '<input type="hidden" name="MAX_FILE_SIZE"
           value="500000"
/>';
echo '<input type="file" name="file"
accept="image/*">';
echo '<input type="submit" name="addadmin"
           value="Add"
           class="btn btn-sm btn-info">';
echo '</form>';
if(isset($_POST['addadmin']))
{
    $userid=$_POST['userid'];
    $fn=$_FILES['file']['tmp_name'];
    $file=fopen($fn, 'rb');
    $img=fread($file, filesize($fn));
    fclose($file);
    $img=addslashes($img);
    $ins='update users set avatar="'. $img .',
           roleid=1 " where
id ='. $userid;
    mysql_query($ins);
}
?>
```

Как всегда, разберем код. В форме не забываем указать атрибут enctype, т.к. мы будем выгружать картинку. Дальше, как обычно, создаем список пользователей. Переходим к обработчику.

Когда мы выполняли upload картинок для таблицы Images, мы переносили эти картинки в паку images нашего сайта. Этот перенос выполняла функция move_uploaded_file(). Сейчас мы не будем хранить картинки в папке images, поскольку мы заносим их сра-

зу в таблицу. Для этого мы открываем файл функцией fopen(), пока он находится во временной папке. Во втором параметре 'rb' мы указываем функции fopen(), что хотим читать открываемый файл как бинарный ('r' означает read, 'b' – binary). Читать бинарный файл функцией fgets(), как мы делали ранее, нельзя. Для чтения бинарных файлов надо использовать специальную функцию – fread(). Но функция fread() требует, чтобы ей указали, сколько байт она должна прочитать. Требуется знать размер читаемого файла. Для определения размера файла надо использовать функцию filesize(), которая требует только путь к файлу и возвращает его размер в байтах. Прочитанный контент заносим в переменную \$img и обрабатываем для безопасности функцией addslashes(), которая экранирует потенциально опасные символы, если такие попадутся. Теперь переменную \$img можно заносить в таблицу, что мы и делаем. Поскольку мы не создаем новую запись, а изменяем существующую, то мы выполняем SQL запрос UPDATE. В этом же запросе мы изменяем роль пользователя. Проверьте, как это работает.

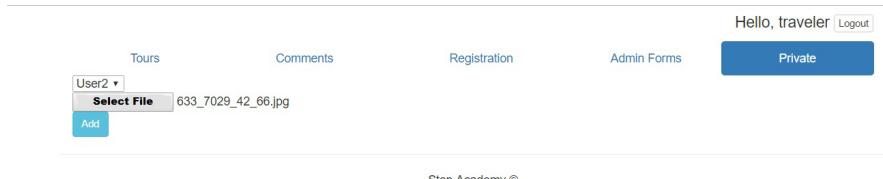


Рис. 34. Изменение роли

Теперь загляните в PhpMyAdmin, чтобы убедиться, что в таблице у выбранного пользователя изменилась роль и появился бинарный контент в поле avatar.

travels		+ Options							
	New	←	→	id	login	email	discount	role_id	avatar
				2	User1	202cb962ac59075b964b07152d234b70	gag	NULL	
				3	traveler	e10adc3949ba59abbe56e057f20f883e	gag	NULL	
				4	User2	202cb962ac59075b964b07152d234b70	aaa@aaa.ua	2	
				5	User3	202cb962ac59075b964b07152d234b70	aaa@bbb.ua	2	

Рис. 35. Изменение пользователя в таблице

Нам осталось разобраться, как отображать изображение из поля типа blob. Существует мнение, что хранить бинарные данные прямо в таблицах БД – это плохой выбор. Что лучше в таблицах хранить только пути к картинкам, а сами картинки хранить в определенной папке на диске. На самом деле оба способа работы с картинками имеют свои достоинства и недостатки. При хранении в таблице путей к рисункам мы получаем меньший размер данных в таблицах, что хорошо. Но при этом СУБД не может обеспечить целостность данных. При хранении в таблице бинарного контента, мы получаем больший размер таблиц, но данные при этом защищены гораздо лучше. Поэтому выбор надо делать применительно к конкретному случаю. Если не позволять пользователю заносить в таблицы очень большие файлы, то применение полей типа blob будет лучшим решением во многих случаях.

Чтобы научиться отображать изображения из бинарного контента мы прямо на этой странице будем выводить список всех существующих администраторов с их портретами. Добавьте в файл private.php сразу после обработчика такой код:

```
$sel='select * from users where roleid=1 order by login';
$res=mysql_query($sel);
echo '<table class="table table-striped">';
while($row=mysql_fetch_array($res,MYSQL_NUM)) {
    echo '<tr>';
    echo '<td>'.$row[0].'</td>';
    echo '<td>'.$row[1].'</td>';
    echo '<td>'.$row[3].'</td>';
    $img=base64_encode($row[6]);
    echo '<td></td>';
}
mysql_free_result($res);
echo '</table>';
```

Мы строим таблицу по результатам выборки пользователей с ролью Admin из таблицы Users. Здесь представлен способ отображения картинок из поля blob. Бинарный контент преобразуется к типу base_64, а затем подставляется в значение атрибута src элемента img. Если вы забыли, что такое формат base_64, я вам напомню. Это замечательный формат, позволяющий кодировать бинарные данные в текстовый формат. Для кодировки используются символы A-Z, a-z, 0-9, +, / и = в качестве заполняющего конечного символа. Обратите внимание, что коды всех этих символов меньше 128, а это значит, что они выглядят одинаково для всех кодировок. Как мы говорили, HTTP протокол поддерживает только текстовые форматы. А вот когда надо передать бинарные данные, как в нашем случае, на помощь приходит base_64.

У меня это выглядит так:

User1	traveler	superadmin@gmail.com	
2	User1	gag@gambler.com	
4	User2	aaa@aaa.ua	

Рис. 36. Таблица администраторов

Теперь осталось сделать так, чтобы пункт меню Private был виден только администраторам. Для этого надо изменить код в файле index.php, где мы подключаем файлы в зависимости от значения page:

```
<?php
if(isset($_GET['page']))
{
    if($page==1) include_once("pages/tours.php");
    if($page==2) include_once("pages/comments.
                                php");
    if($page==3) include_once("pages/registration.
                                php");
    if($page==4) include_once("pages/admin.php");
    if($page==6 && isset($_SESSION['radmin']))
        include_once("pages/private.php");
}
?>
```

И еще надо добавить новый пункт меню в файл menu.php:

```
<ul class="nav nav-pills nav-justified">
<li <?php echo ($page==1) ? "class='active'" : "" ?>>
<a href="index.php?page=1">Tours</a></li>
<li <?php echo ($page==2) ? "class='active'" : "" ?>>
<a href="index.php?page=2">Comments</a></li>
<li <?php echo ($page==3) ? "class='active'" :"hhh" ?>>
<a href="index.php?page=3">Registration</a></li>
<li <?php echo ($page==4) ? "class='active'" : "" ?>>
<a href="index.php?page=4">Admin Forms</a></li>
<?php
if(isset($_SESSION['radmin']))
{
    if($page==6)
        $c='active';
    else
        $c=' ';
    echo '<li class="'. $c .'>'>
<a href="index.php?page=6">Private</a></li>';
}
?>
</ul>
```

Помните, что значение page, равное 5, принадлежит файлу login? Поэтому для пункта меню Private мы используем значение 6. Вот теперь наш план выполнен.

Подведение итогов

В процессе создания нашего сайта мы рассмотрели основные приемы применения PHP при взаимодействии с БД. Однако несколько вопросов остались не рассмотренными.

Мы работали с БД только с помощью функций mysql_. В следующих уроках мы рассмотрим другие инструменты PHP для работы с БД.

Мы оставили открытый вопрос о взаимодействии двух списков с помощью Ajax. Обязательно вернемся к этому вопросу и исправим страницу выбора тура.

Вы должны были обратить внимание на важность использования таких инструментов, как Bootstrap и jQuery. Хотя PHP и является мощным и гибким инструментом для веб-разработки, он самостоятельно не может создавать приложения, которые сегодня требует современный рынок. Работа в вебе требует от программиста умения использовать и применять широкий спектр разных инструментов для создания современного приложения.

Работая с нашим сайтом, мы использовали функции `_mysql`. Вы уже привыкли к ним, и поняли логику их использования. Однако, вы должны знать, что использование этих функций – не лучший выбор на сегодняшний день. Начиная с версии PHP 5.4, эти функции признаны устаревшими и не рекомендованы к использованию. Зачем же мы их изучали? Дело в том, что на протяжении многих лет эти функции были самым распространенным способом работы с БД. Многим из вас наверняка придется сопровождать проекты, созданные с применением этих функций. В таком случае опыт из этого урока будет вам просто необходим.

Во всех наших примерах намеренно опускалась часть работы, которая обязательно должна выполняться на клиентской части приложения. Для этой работы надо применять такие инструменты, как Javascript, jQuery, jQueryUI и AngularJS. В реальной разработке эту работу надо выполнять.

Домашнее задание

Поговорим о том, что осталось сделать на этом сайте. Мы еще не создали механизм добавления пользовательских отзывов об отелях. Для этого зарезервированы второй пункт меню и страница comments.php. На этой странице надо создать форму, позволяющую выбрать конкретный отель (например, из выпадающего списка), как мы это делали раньше. Эта форма также должна содержать поле для добавления текстового комментария и кнопку submit. Добавленный комментарий должен сохраняться в таблице comments. Отображаться введенные пользователями комментарии должны на странице hotelinfo.php, внизу страницы под основной информацией об отеле.

Вы уже догадались, что ваше домашнее задание состоит в доработке страницы hotelinfo.php, как мы об этом говорили раньше, и в создании механизма комментариев. Удачи.



Урок №3

Базы данных и работа с MySQL

© Александр Геворкян
© Компьютерная Академия «Шаг»
www.itstep.org

Все права на охраняемые авторским правом фото-, аудио- и видеопрограммные изделия, фрагменты которых использованы в материале, принадлежат их законным владельцам. Фрагменты произведений используются в иллюстративных целях в объеме, оправданном поставленной задачей, в рамках учебного процесса и в учебных целях, в соответствии со ст. 1274 ч. 4 ГК РФ и ст. 21 и 23 Закона Украины «Про авторське право і суміжні права». Объем и способ цитируемых произведений соответствует принятым нормам, не наносит ущерба нормальному использованию объектов авторского права и не ущемляет законные интересы автора и правообладателей. Цитируемые фрагменты произведений на момент использования не могут быть заменены альтернативными, не охраняемыми авторским правом аналогами, и как таковые соответствуют критериям добросовестного использования и честного использования.

Все права защищены. Полное или частичное копирование материалов запрещено. Согласование использования произведений или их фрагментов производится с авторами и правообладателями. Согласованное использование материалов возможно только при указании источника.

Ответственность за несанкционированное копирование и коммерческое использование материалов определяется действующим законодательством Украины.