**ЛАБОРАТОРНАЯ РАБОТА № 1 «РЕАЛИЗАЦИЯ WEB-ПОРТАЛА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PHP»**

При выполнении данного задания необходимо создать web-портал, имеющего доступ к базам данных сервера MySQL. В качестве примера разработаем сайт заказа блюд ресторана. Для создания сайта нам потребуется программный пакет Open Server 4.х.х, включающего веб-сервер Apache 2, интерпретатор PHP 5, сервер баз данных MySQL 5 и PostgreSQL 9. Стандартными расширениями веб-сервера являются: PhpMyAdmin, MySQL менеджер, SQL менеджер. Управление базой данных MySQL осуществляется с помощью динамических веб-страниц расширения PhpMyAdmin **(**см. рис. 4.1**)**, а построение сайта в программе «Блокнот++».

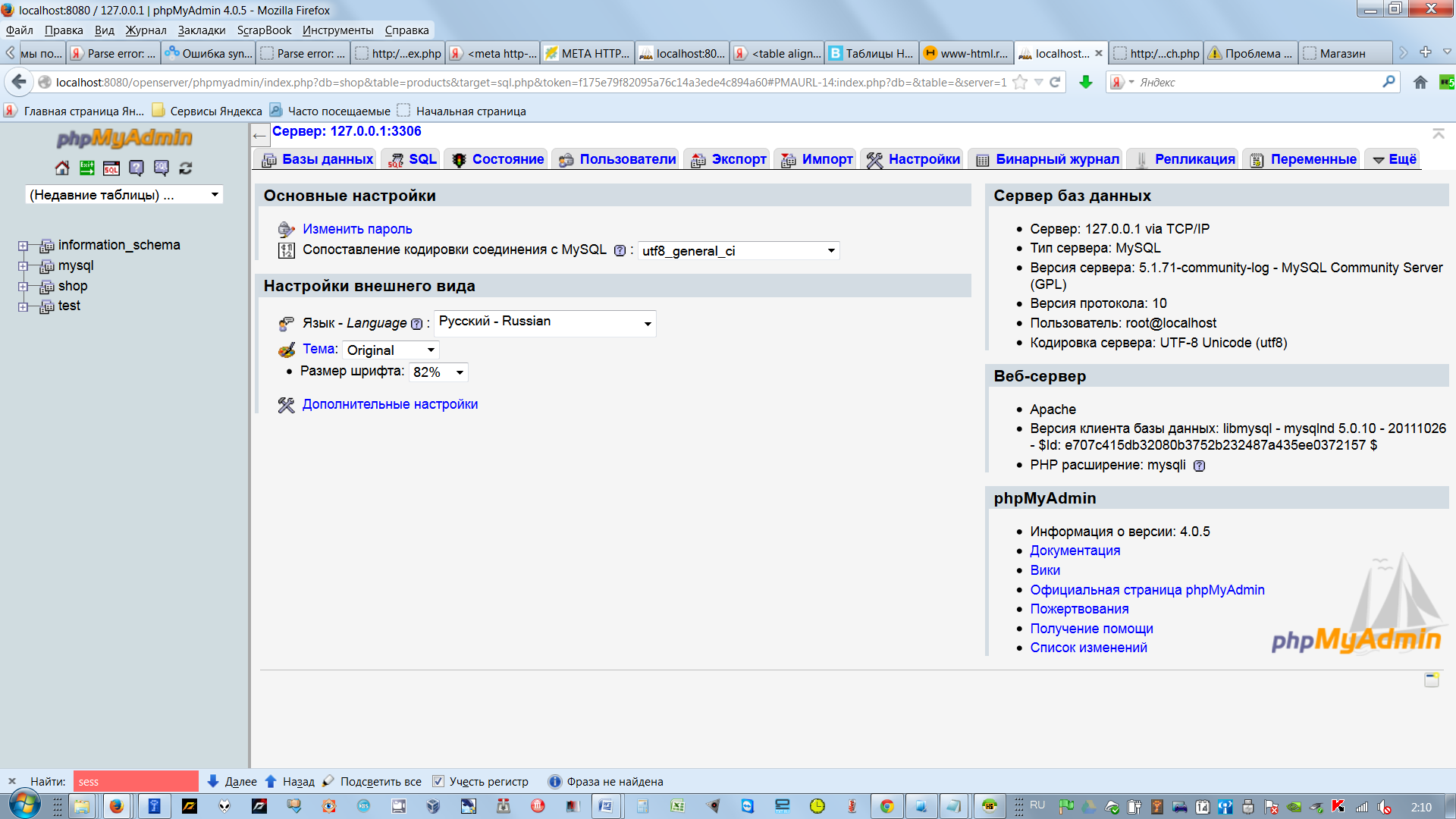


Рис. 4.1 Внешний вид расширения «PhpMyAdmin»

Создадим базу данных «Shop» с помощью вкладки «Базы данных», при этом указываются ее название и ее кодировка. Наиболее распространенной таблицей символов является глобальная кодировка utf8\_general\_ci.

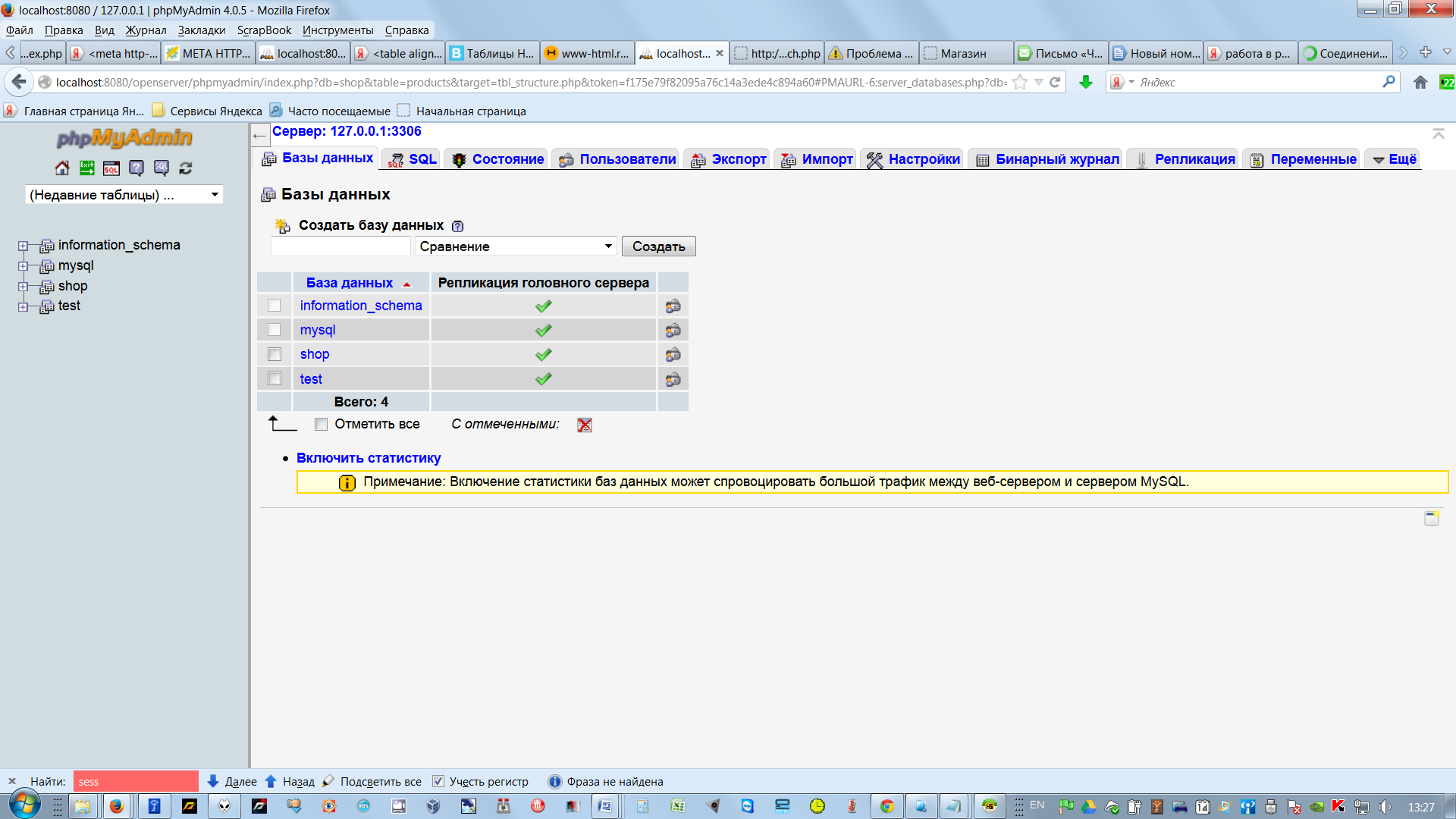


Рис. 4.2 Веб-страница «База данных»

Безопасность доступа к данным сервера базы данных настроим с помощью вкладки «Привилегии». Для этого необходимо выбрать созданную базу данных и данную вкладку и нажать кнопку добавить пользователя.

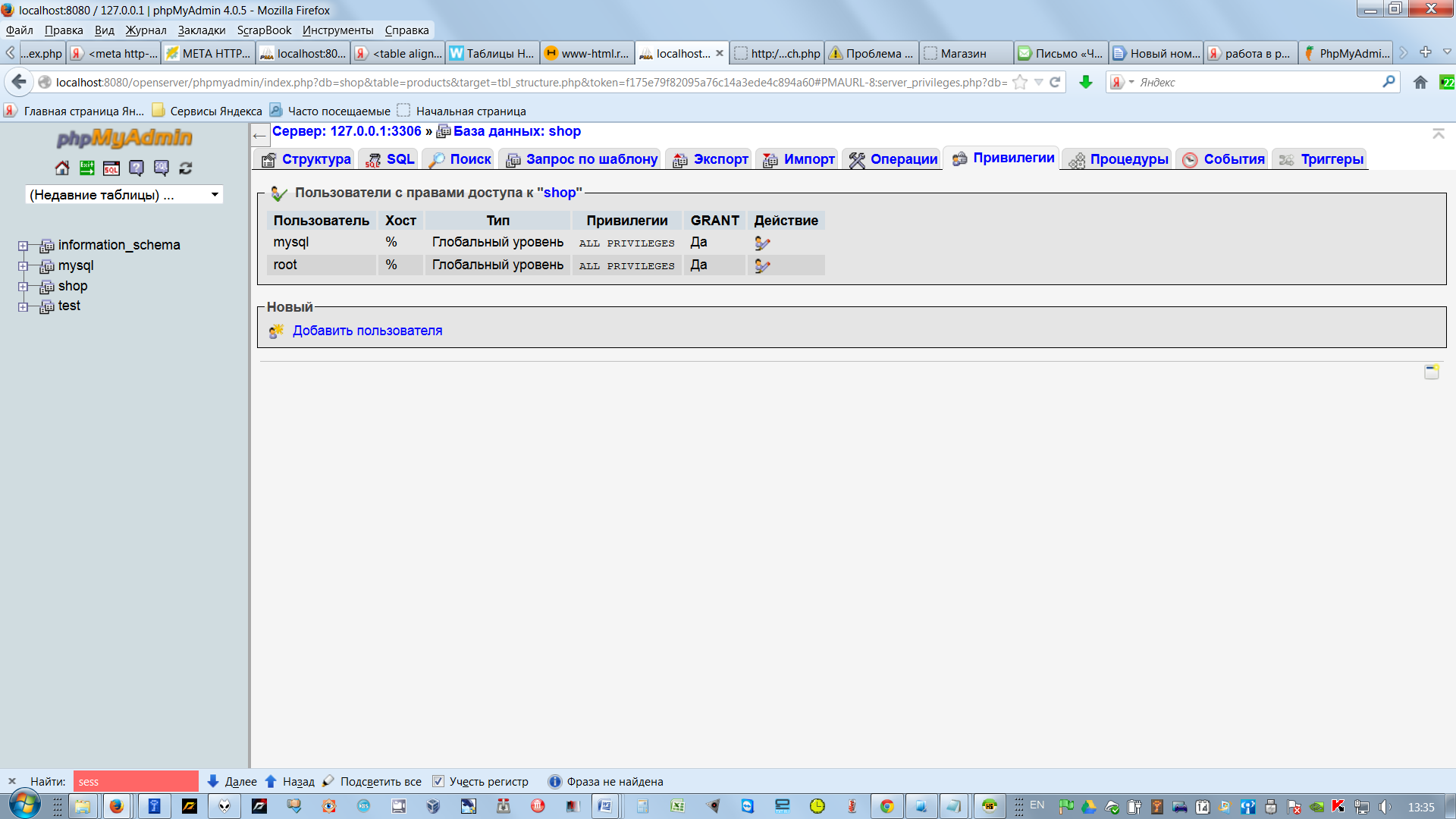


Рис. 4.3 Веб-страница «Привилегии»

Далее указывается имя пользователя, генерируется его пароль и нажатием кнопки «ОК» сохраняются данные. В поле «Хост» указываются имена рабочих станций, с которых пользователь может подключиться к серверу баз данных MySQL.

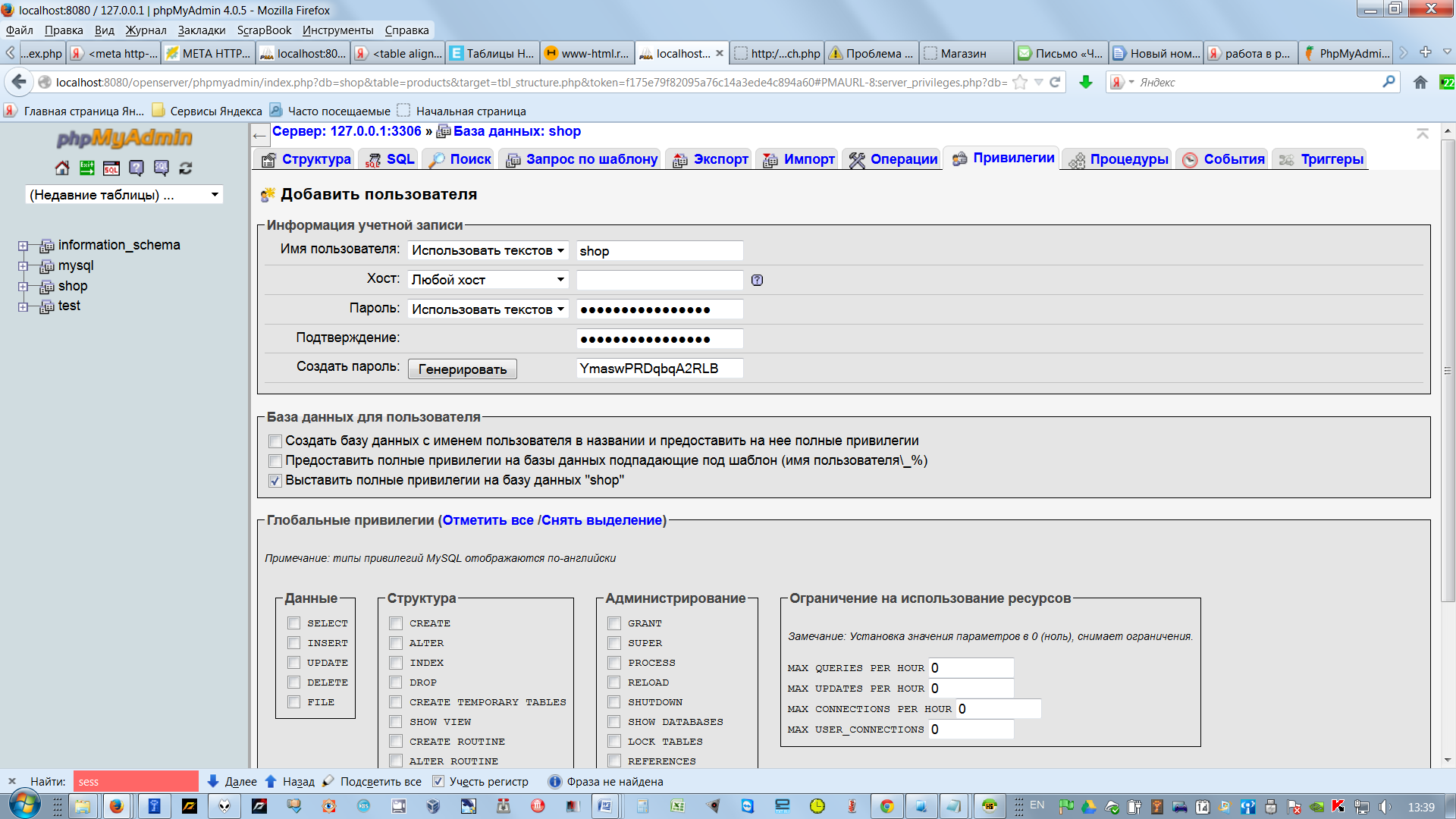


Рис. 4.4 Веб-страница «Добавить пользователя»

Предлагаемая база данных заказов ресторана («Shop») состоит из трех таблиц: products (блюда), categories (категории блюд), orders (заказы). Таблицы создаются с помощью вкладки «Структура», указываются имя и количество столбцов. Затем аналогично подтверждается действие нажатием кнопки «ОК».

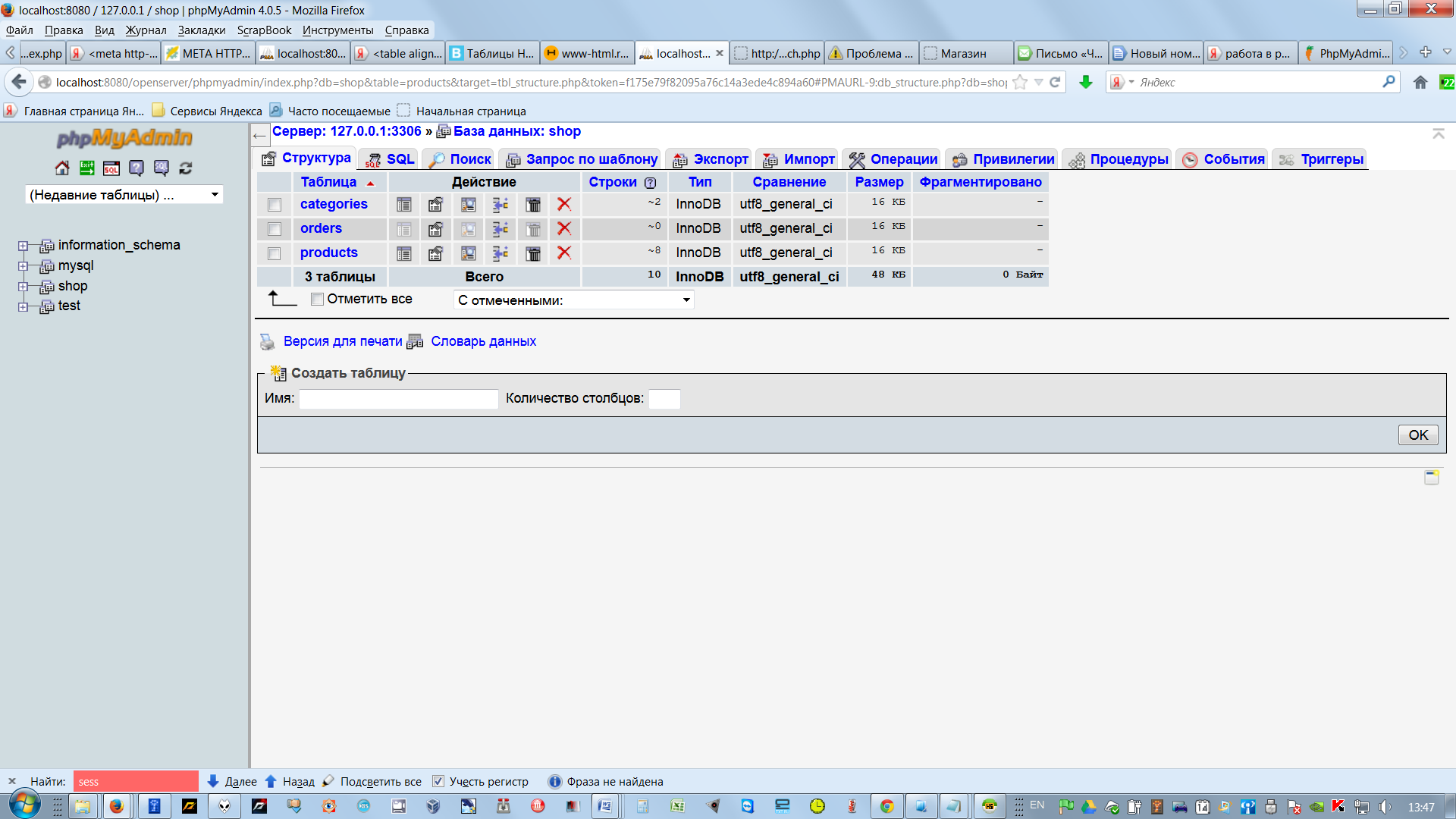


Рис. 4.5 Веб-страница «Структура»

В открывшейся веб-странице вводиться названия атрибутов таблиц, соответствующий им тип данных: целочисленный - int(11), вещественный -decimal (8,2), строка - varchar(255), текст - text, дата – date и время - time. В полях «atribut2» и «atribut3» тип кодировки указываем utf8\_general\_ci.

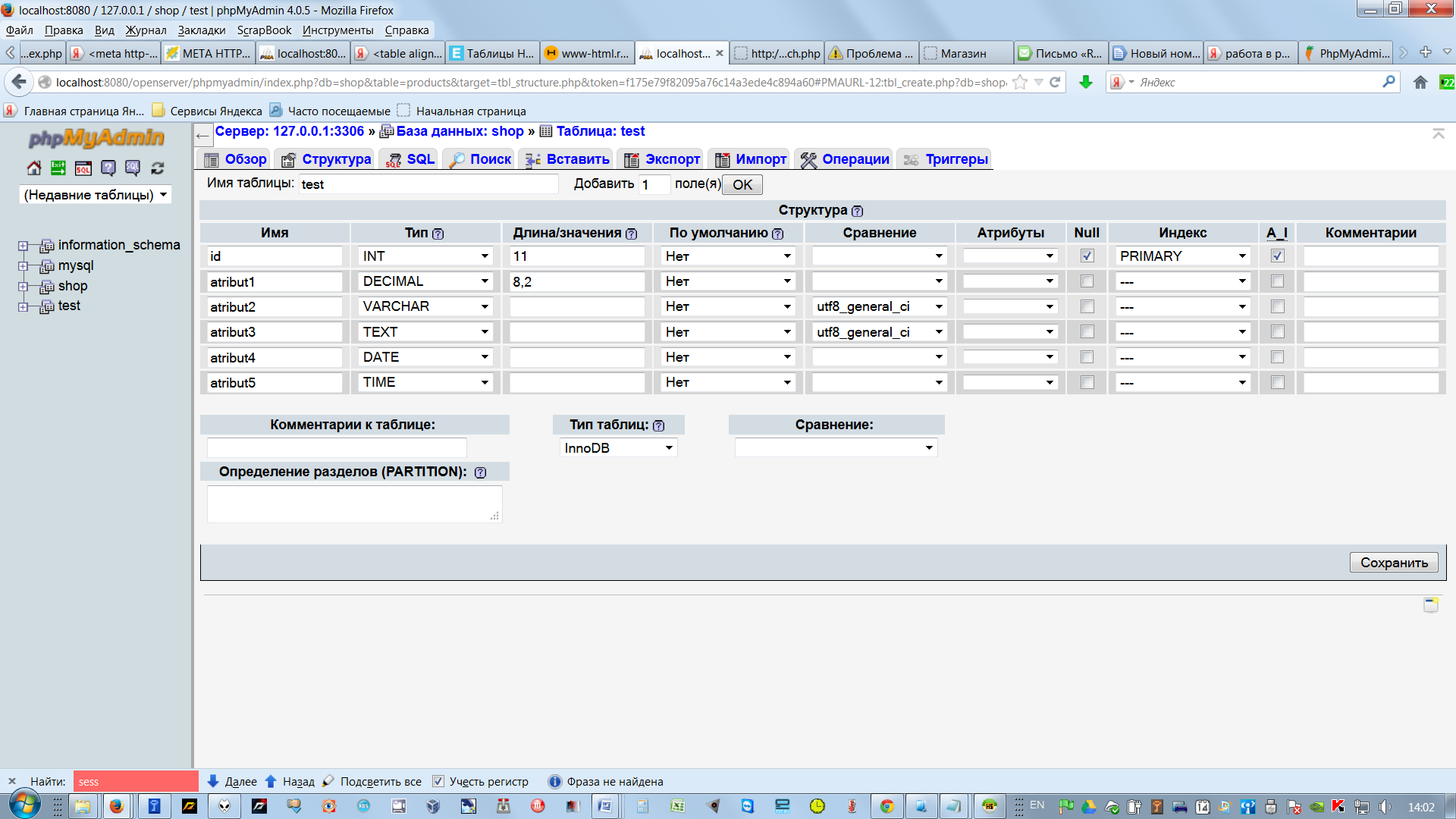


Рис. 4.6 Веб-страница «Структура таблицы»

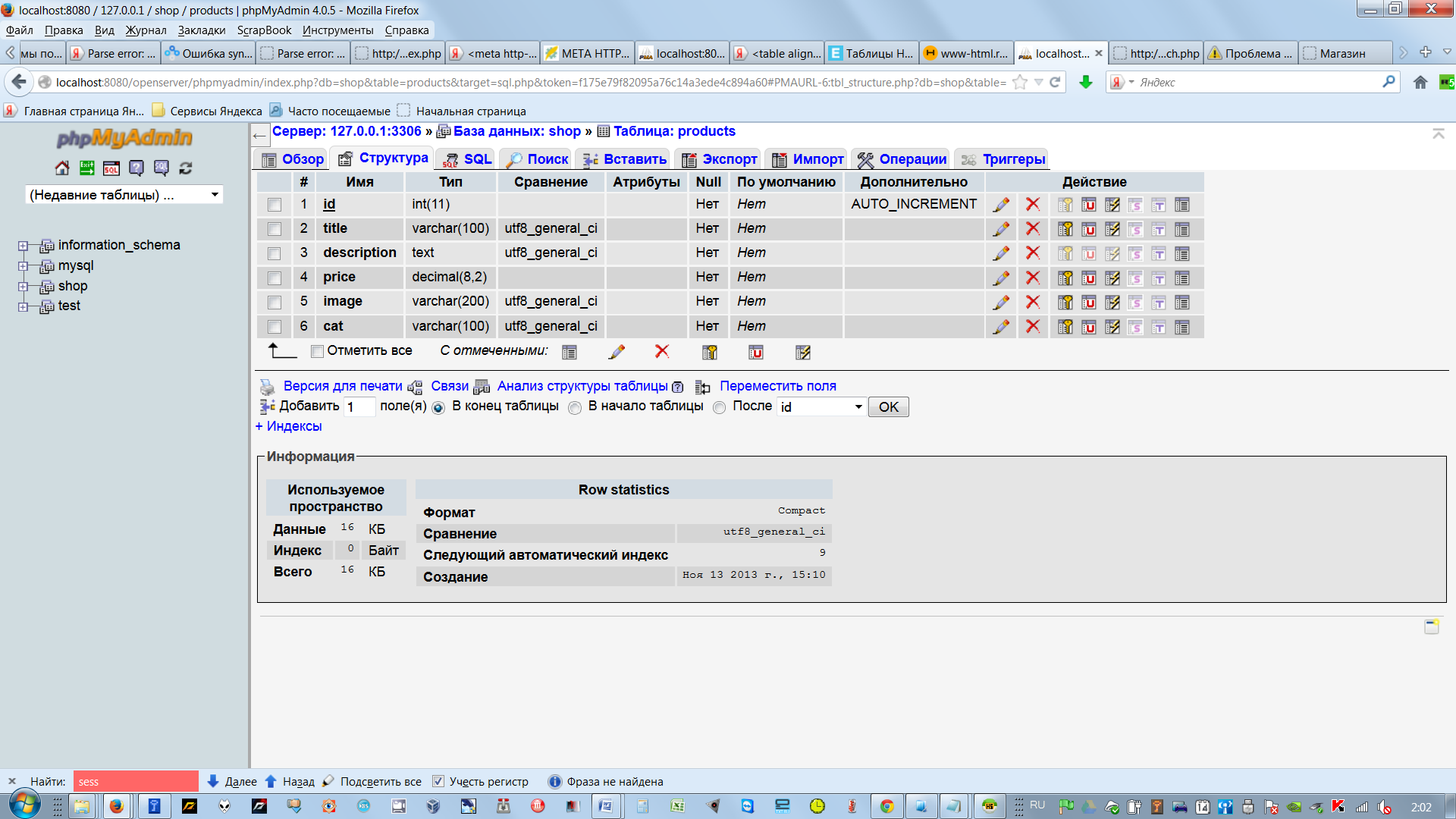
Ниже на рис. 4.7-4.9 представлены веб-страницыPhpMyAdmin, отображающие физическую структуру данных таблиц**.** Дополнительными параметрами являются «Сравнение» и «Дополнение», позволяющие указать кодировку хранения данных каждого атрибута и генерацию его значения. Первичный ключ, уникальные значения, индекс фиксируются тремя элементами управления, представленными в виде пиктограмм ****.

Таблица «products» содержит 6 атрибутов: id – код блюда, первичный ключ; title – название блюда; description – описание блюда; price – цена блюда; image – название файла с изображением блюда; cat – категория блюда.

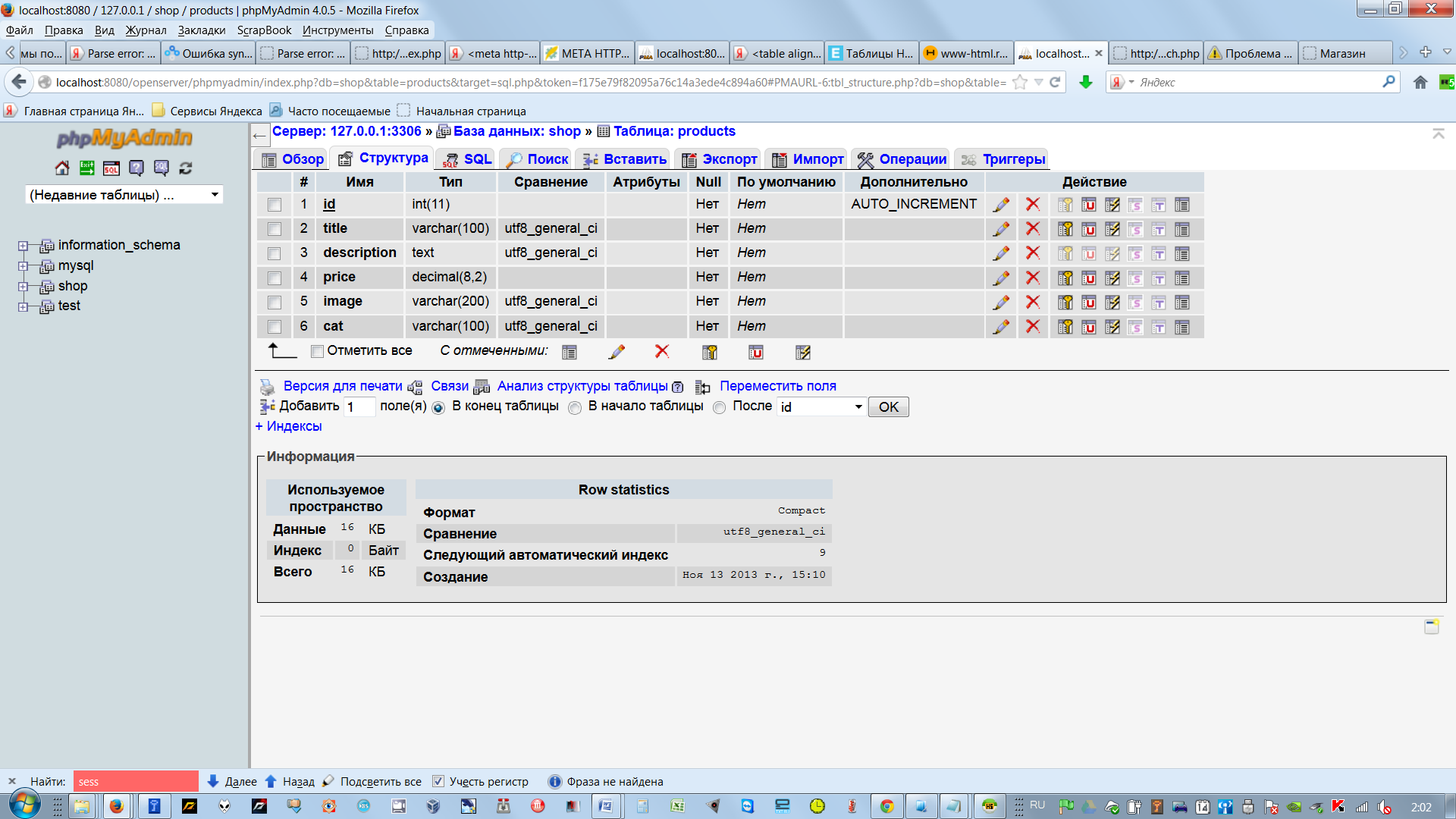
****

Рис. 4.7 Веб-страница для работы с таблицей «products»

Таблица «categories» содержит 3 атрибута: id – код категории, первичный ключ; name – заголовок категории; cat\_id – название категории.

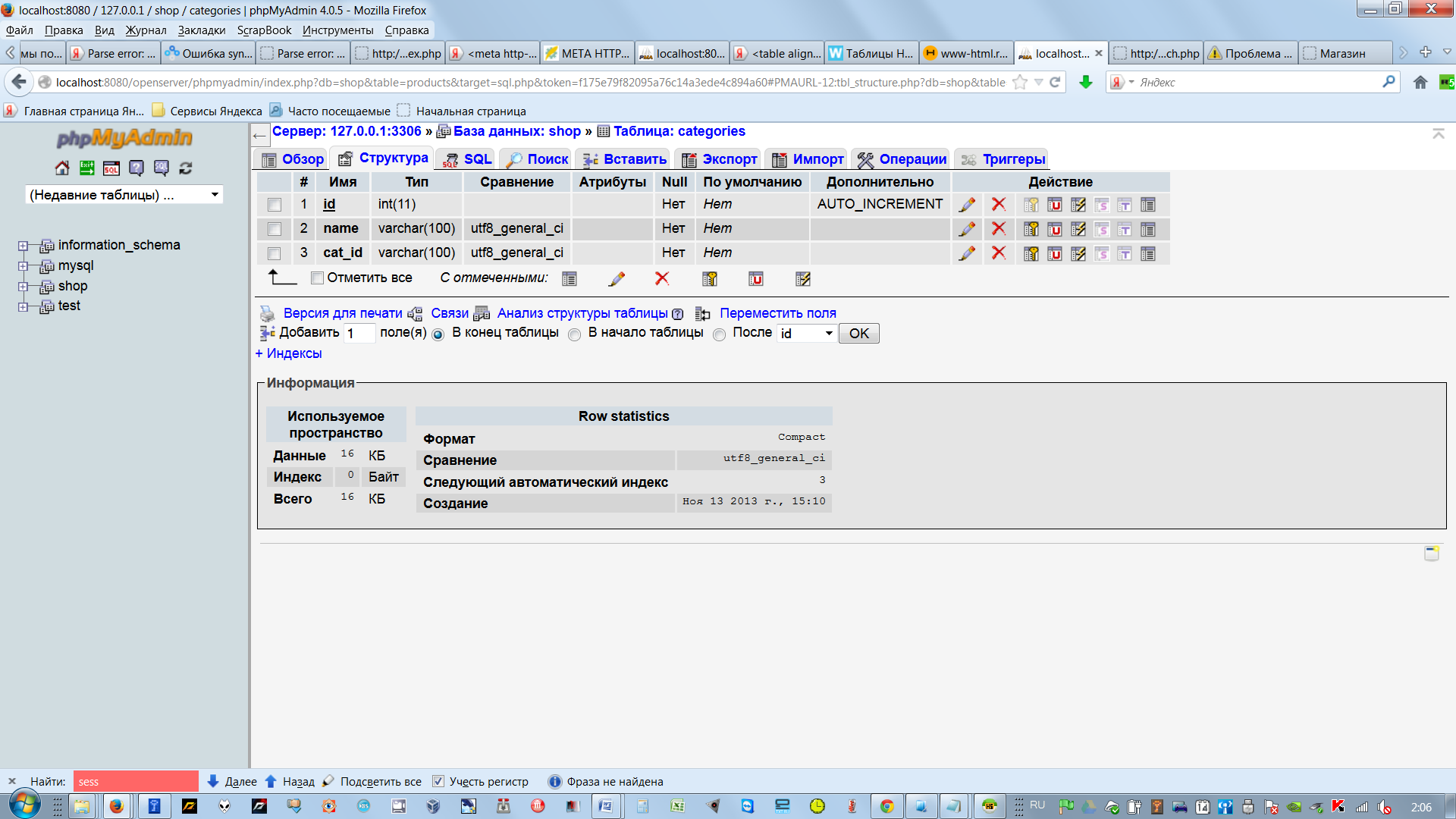
****

Рис. 4.8 Веб-страница для работы с таблицей «categories»

Таблица «orders» содержит 12 атрибутов: id – код заказа, первичный ключ; product – название блюда; prod\_id – код блюда; price – цена блюда; qty – количество блюд; name – фамилия клиента; s\_name – имя клиента; address – адрес клиента; post\_index – почтовый индекс; email – электронная почта; date и time – дата и время создания заказа.

****

Рис. 4.9 Веб-страница для работы с таблицей «orders»

Заполнение базы данных выполняется вручную только для двух таблиц «categories» и «products» с помощью веб-страницы «Вставить» расширения PhpMyAdmin, таблица «orders» заполняется скриптом на языке программирования PHP.

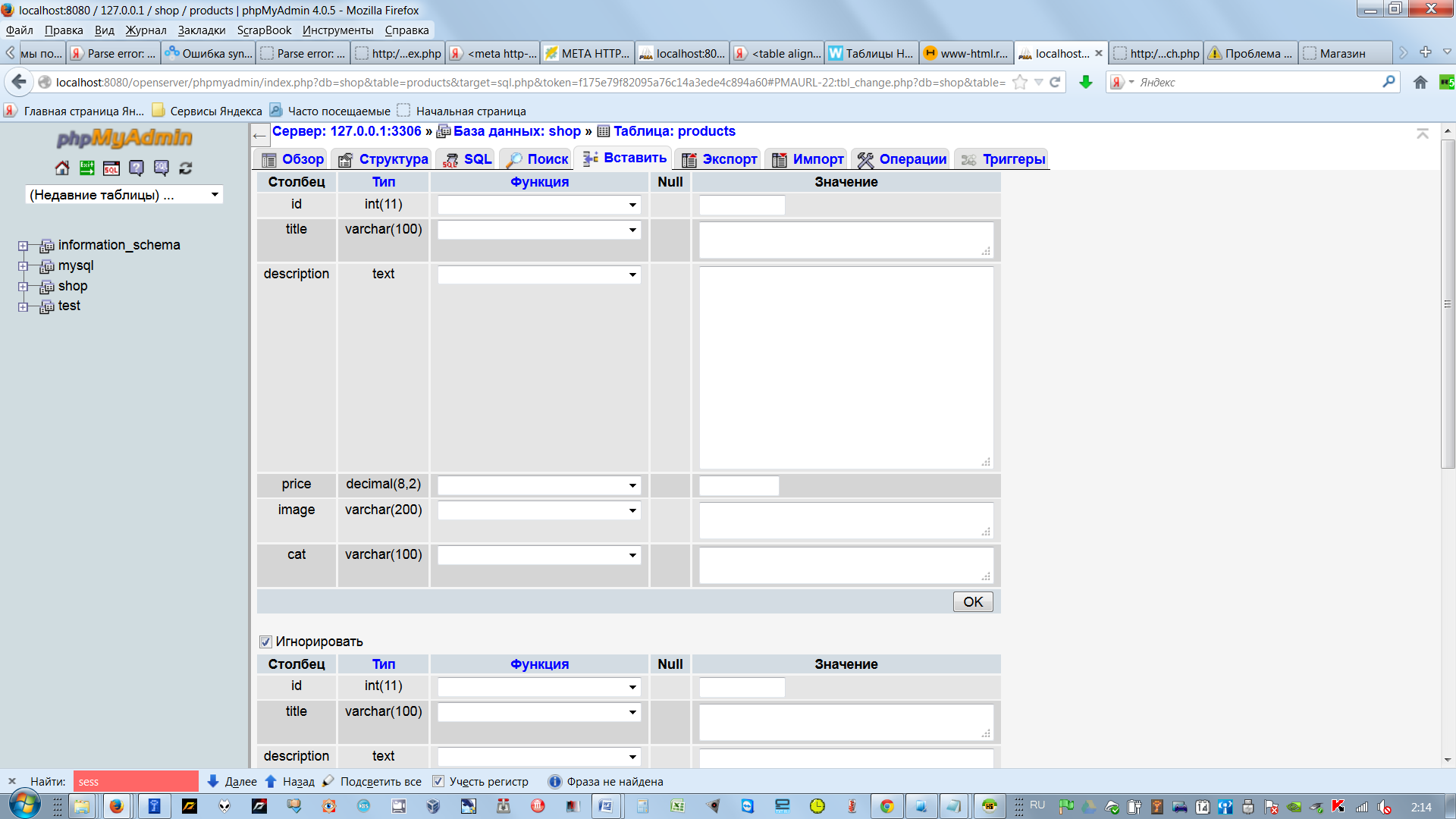
****

Рис. 4.10 Веб-страница «Вставить»

Тестирование SQL-запросов к созданным таблицам осуществляется с помощью вкладки «SQL».

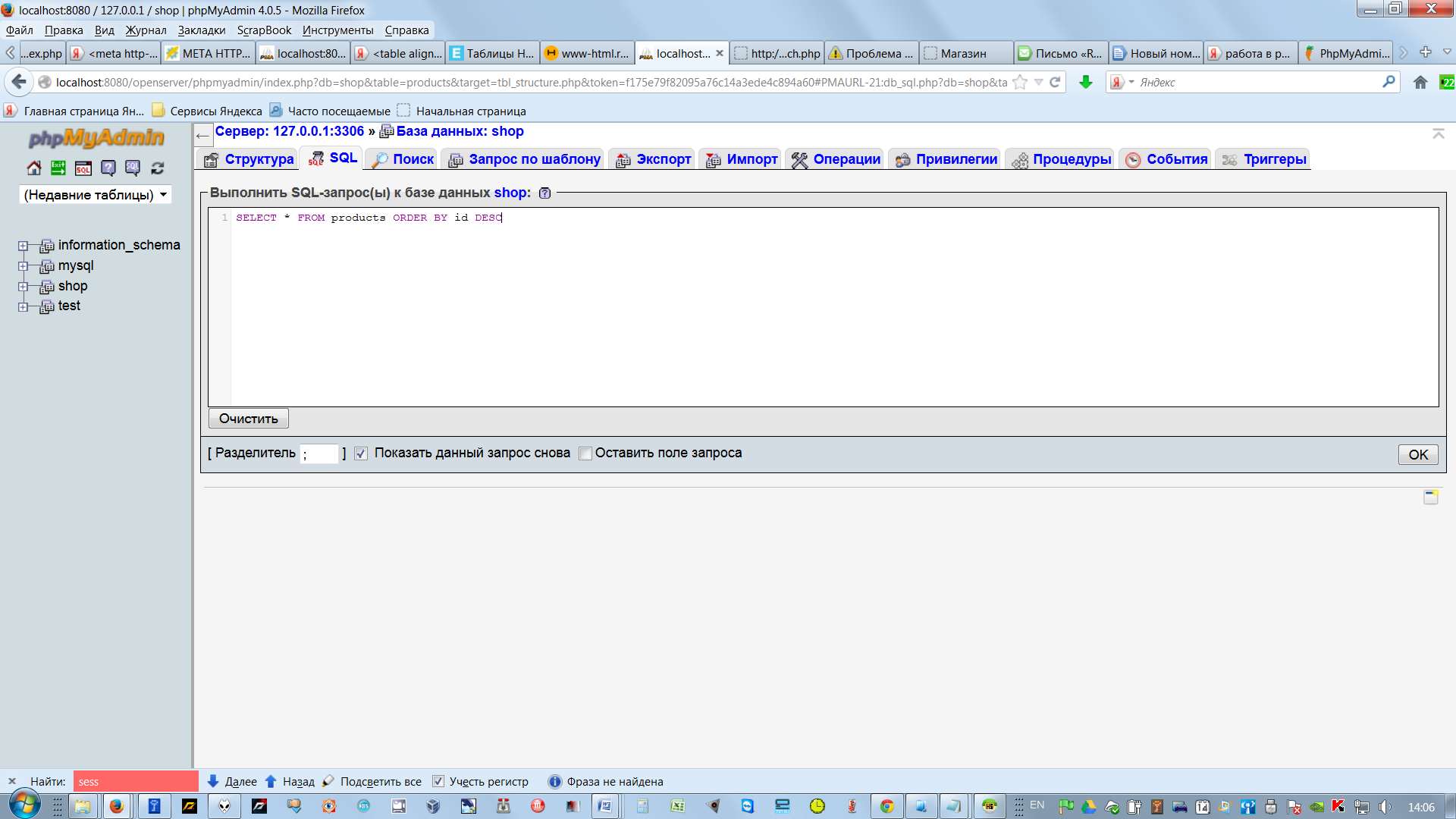


Рис. 4.11 Веб-страница «SQL»

Для разработки собственных динамических веб-страниц воспользуемся программой «Блокнот++», позволяющей оптимально создавать и проверять на скрипты и сценарии на ошибки (cм. рис. 4.5).

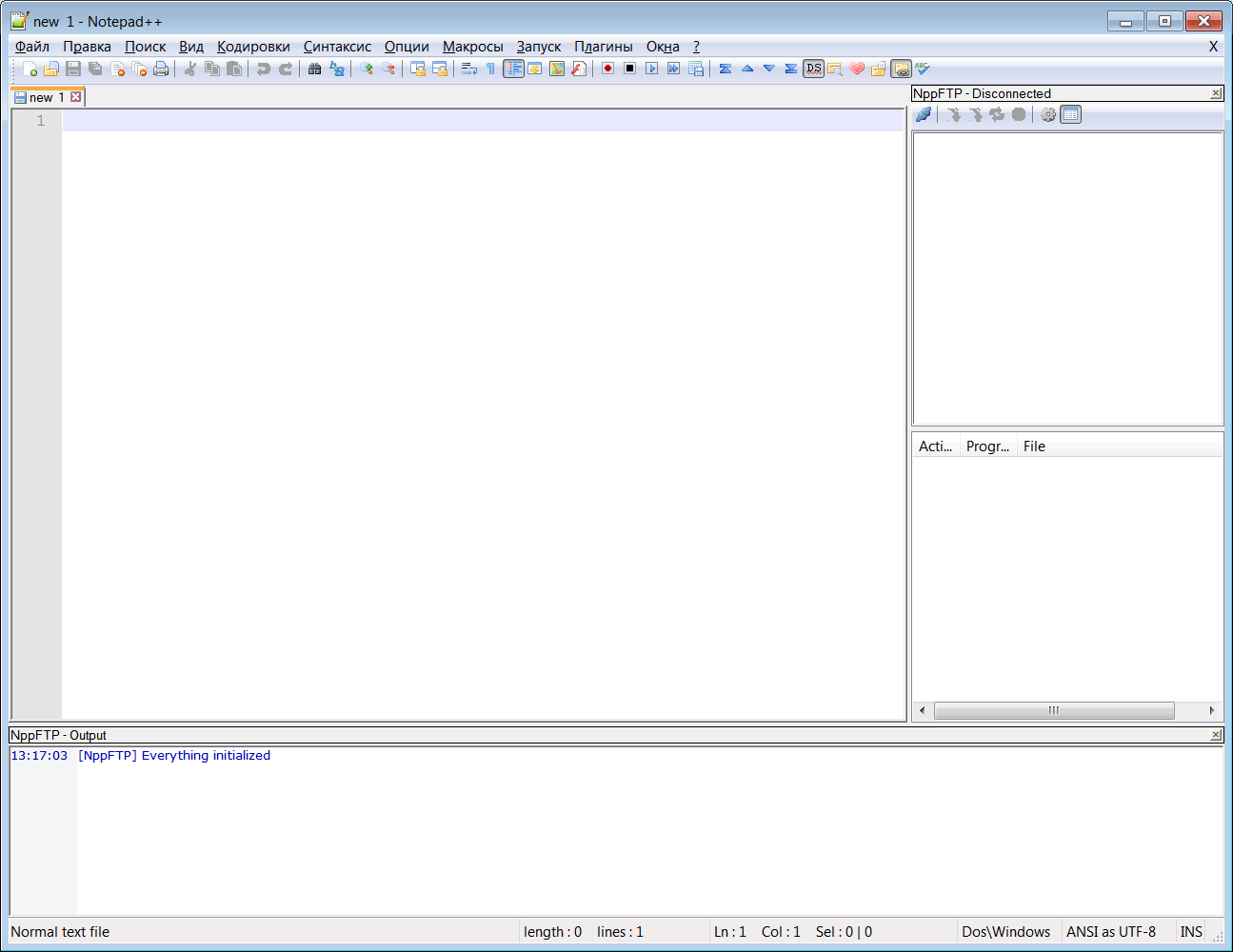
****

Рис. 4.12 Программа «Блокнот ++»

Рабочим каталогом веб-портала является папка domains/localhost программного пакета OpenServer. В данную папку сохраняются скрипты на языке программирования PHP, и вспомогательные файлы в виде изображений блюд.

Создадим главный скрипт **index.php** для контроля перемещения по динамическим страницам веб-портала и просмотра корзины выбранных блюд. Вначале скрипта с помощью оператора **include** подключаются два внешних скрипта **db\_fns.php** и **cart\_fns.php**, содержащие конфигурации подключения к базе данных MySQL и обработки данных корзины. При подключение клиента скрипт создаем элемент «сессия», позволяющего отслеживать все его действия на веб-портале, заносить во временную таблицу выбранные блюда и подсчитывать общую стоимость. Если пользователь в первый раз заходит на данный портал, то скрипт автоматически создает три виртуальные переменные для хранения заказа: **cart** – список выбранных блюд; **total\_items** – количество выбранных блюд; **total\_price** – общая стоимость блюд. Далее в зависимости от параметров запуска скрипта **index.php** происходит загрузка данных в переменную **$product**, либо данных в сеансовые переменные **$\_SESSION['total\_items'],** **$\_SESSION['total\_price']**.

<?

include('db\_fns.php'); include('cart\_fns.php');

session\_start(); //запуск сессии

if (!isset($\_SESSION['cart'])) { //проверка заполнения корзины

$\_SESSION['cart'] = array();

$\_SESSION['total\_items']=0;

$\_SESSION['total\_price']= 0.00;}

$view = empty($\_GET['view']) ? 'index' : $\_GET['view'];

switch($view) { //выбор параметров обработки

case('index'):

$product=get\_products();

break;

case('cat'):

$cat=$\_GET['id'];

$product=get\_cat\_products($cat);

break;

case('product'):

$id=$\_GET['id'];

$product=get\_product($id);

break;

case('cart'): break;

case('add\_to\_cart'):

$id=$\_GET['id'];

$add\_item=add\_to\_cart($id);

$\_SESSION['total\_items'] = total\_items($\_SESSION['cart']);

$\_SESSION['total\_price'] = total\_price($\_SESSION['cart']);

header('Location: index.php?view=product&id='.$id);

break;

case('update\_cart'):

update\_cart();

$\_SESSION['total\_items'] = total\_items ($\_SESSION ['cart']);

$\_SESSION['total\_price'] = total\_price($\_SESSION['cart']);

header('Location: index.php?view=cart');

break;

case('order'): break;

}

$arr = array('index','cat','product','cart','add\_to\_cart','update\_cart','order');

if(!in\_array($view,$arr)) die("ERROR 404!");

include($\_SERVER['DOCUMENT\_ROOT'].'/shop.php');

?>

После выполнения скрипта **index.php** происходит загрузка скрипта **shop.php**, отвечающего за шаблон заказа блюд в ресторане. Шаблон заказа состоит из трех компонентов: область гиперссылок; корзина заказа; отображение блюд ресторана. Данный скрипт содержит возможность загружать скрипты в зависимости от параметров **$view** запуска **index.php**, и передачи этих параметров в скрипт **shop.php**: include($\_SERVER['DOCUMENT\_ROOT'].'/'.$view.'.php');

<html> <head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Магазин</title>

</head>

<body>

<table align="center" width="900" cellpadding="0" cellspacing="0" border="О" id="main-table">

<tr> <td>

<div id="header"></div>

<div id="menu">

<div><a href="index.php">Главная</a></div>

<div><a href="index.php?view=order">Заказ</a></div>

<?

$categories = get\_cat();

foreach($categories as $item) {

echo '<div><a href="index.php?view=cat&id='.$item['cat\_id'].'">'.$item['name'].'</a></div>'; }

?>

<div id="cart" href="index.php?view=cart">Ваша корзина (<? $\_SESSION['total\_items']; ?>)</a> - <?=number\_format($\_SESSION['total\_price'],2);?> $</div>

</div> </td> </tr>

<tr>

<td id="main-block" valign="top">

<?php include($\_SERVER['DOCUMENT\_ROOT'].'/'.$view.'.php'); ?>

<div style="clear"></div>

</td> </tr> <tr>

<td id="footer-td">

<div align="center">

<div class="footer">&copy; Ресторан.ру 2013</div>

</div> </td> </tr> </table>

</body> </html>

Для отображения списка блюд параметр **$view** принимает значение **cat**, в результате загружается скрипт **cat.php** с выбранной категорией блюд. В скрипте используется цикл **foreach** для перебора и отображения всех записей в переменной **$product**. Доступ к атрибутам записей осуществляется через массив **$item**, имеющего индексы аналогичны атрибутам таблицы «**product**»

<?php

echo '<table align="center" cellpadding="0" cellspacing="0" class="product" border="1">';

echo '<tr>';

?>

<? foreach($product as $item){

echo '<th valign="top">';

echo '<div> <a href="index.php?view=product&id='.$item['id'].'"><br>';

echo '<img src="'.$item['image'].'" alt="" height=100 width=100 /></a></div>';

echo '<div class="description">';

echo '<div class="product-name">'.$item['title'].'</div>';

echo '<div class="product-price">Цена:'.$item['price'].'$</div>';}

?>

<?php

echo '</div> </th> </tr> </table>';

?>

Внешний вид скрипта представлен на рис. 4.6.



Рис. 4.13 Внешний вид скрипта index.php c параметром «?view=cat»

Выбор блюда осуществляется нажатием левой кнопки мыши по его изображению, затем загружается веб-страница **index.php?view=product&id='.$item['id']**, передающая параметры запуска в скрипт **shop.php** и далее запуск скрипта отображения данных блюда **product.php**. Ниже показан программный код скрипта **product.php**.

<table align="center" cellpadding="0" cellspacing="0" class="product" border="1">

<td valign="top">

<div><a href="#">

<?php

echo '<img src="'.$product['image'].'" alt="" width=400 height=400 /></a></div>';

?>

<div class="description">

<?php

echo '<div class="product-name">'.$product['title'].'</div>';

echo '<div class="product-price">Ценa:'.$product['price'].'руб.</div>';

?>

</div>

</td>

<td valign="top" width="100">

<?php

echo '<div>'.$product['description'].'</div> <br> <a href="index.php?view=add\_to\_cart&id='.$product['id'].'">Добавить в корзину</a> </div>'

?>

</td>

</table>

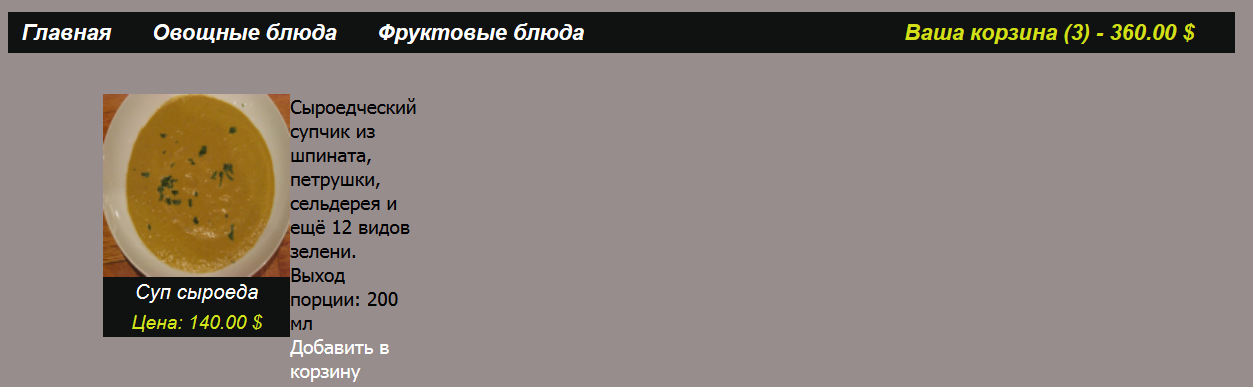


Рис. 4.13 Внешний вид скрипта index.php c параметров «?view=product»

В скрипте **product.php** присутствует гиперссылка **index.php?view=add\_to\_cart&id='.$product['id']** для добавления выбранного блюда в корзину, т.е. происходит вызов скрипта **index.ph**p и передача параметров в сессионные переменные. Оператор **header('Location: index.php?view=product&id='.$id)** предназначен для просмотра выбранного блюда после добавления его в корзину.

Для формирования предварительного списка блюд заказа в корзине вызывается скрипт для оформления заказа **orders.php**. Условие **if($\_SESSION['cart'] && !isset($\_POST['order']))** проверяет наличие данных в корзине, либо данных о клиенте, сформированных формой скрипта. В случае, если данных клиента заполнены и отправлены данному скрипту, то происходит сохранение заказа в таблицу «**orders**».

<h2 align="center"> Оформление заказа</h2>

<?php

if($\_SESSION['cart'] && !isset($\_POST['order'])) {

echo '<form action="index.php?view=order" method="post" id="cart-form">';

echo '<table id="mycart" align="center" cellspacing="0" cellpadding="0" border="1">';

echo '<tr> <th>Tовap</th> <th>Цeнa</th> <th>Koл-во</th> <th>Bceгo</th> </tr>';

foreach($\_SESSION['cart'] as $id => $quantity){

$product = get\_product($id);

echo '<tr>';

echo '<td align="center">'.$product['title'].'</td>';

echo '<td align="center">$'.number\_format($product['price'],2).'</td>';

echo '<td align="center">'.$quantity.'</td>';

echo '<td align="center">$'.number\_format($product ['price'] \* $quantity ,2).'</td>';

echo '</tr>';

}

echo '</table>';

echo '<p class="total" align-"center">Общая сумма заказа: <span class-"product-price">'.number\_format($\_SESSION['total\_price'], 2).'$</span></p>';

echo '<p align="center" style="">Ваше Имя: <br>';

echo '<input type="text" name="name"><br> Фамилия: <br>';

echo '<input type="text" name="s\_name"><br> Адрес: <br>';

echo '<input type="text" name="address"><br> Почтовый индекс: <br>';

echo '<input type="text" name="post\_index"><br> E-mail: <br>';

echo '<input type="text" name="email"><br>';

echo '</p>';

echo '<p align="center"><input type="submit" name="order" value="Заказать" /></p>';

echo '</form>';

}

if ($\_SESSION['cart'] && isset($\_POST['order']))

{

foreach($\_POST as $ArrKey => $ArrStr)

{

$ArrKey = $\_POST[$ArrKey];

}

$name= $\_POST['name']; $s\_name= $\_POST['s\_name']; $address= $\_POST['address'];$post\_index= $\_POST['post\_index'];$email= $\_POST['email'];

$date = date('y-m-d'); $time = date('h:i:s');

foreach($\_SESSION['cart'] as $id => $quantity){

$product = get\_product($id);

$query="INSERT INTO orders(name,s\_name,address,post\_index,email,date,time,product,prod\_id,price,qty) VALUES ('$name', '$s\_name', '$address', '$post\_index', ' $email', '$date', ' $time', '{$product['title']}', '{$product['id']}','{$product['price']}','$quantity')";

$query = mysql\_query($query);

//echo $query;

}

echo "<p align='center' style=''>Baш заказ успешно принят! Спасибо за покупку!</p>";

}

?>

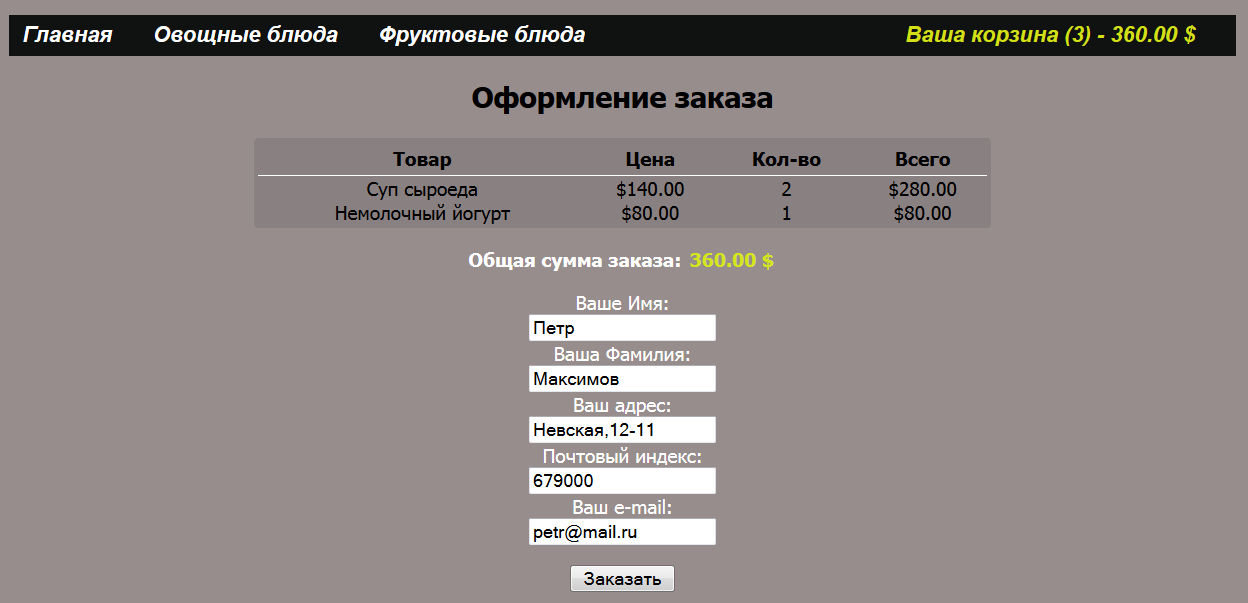


Рис. 4.13 Внешний вид скрипта order.php

Скрипт конфигурирования подключения к базе данных **db\_fns.php** содержит функции для работы с базой данных MySQL. Подключение и реализация запросов осуществляется стандартными функциями: mysql\_connect, mysql\_query, mysql\_fetch\_array.

<?

function db\_connect1(){

$host = 'localhost';

$user = 'shop';

$pswd = 'YmaswPRDqbqA2RLB';

$db = 'shop';

$connection = mysql\_connect($host, $user, $pswd);

mysql\_query("SET NAMES utf8");

if (!$connection || !mysql\_select\_db($db,$connection)) {

return false;

}

return $connection;}

function db\_result\_to\_array($result){

$res\_array = array();

$count =0;

while($row = mysql\_fetch\_array($result)){

$res\_array[$count] = $row; $count++;

}

return $res\_array;}

function get\_products(){

db\_connect1();

$query = "SELECT \* FROM products ORDER BY id DESC";

$result = mysql\_query($query);

$result = db\_result\_to\_array($result);

return $result;}

function get\_cat\_products($cat) {

db\_connect1();

$query = "SELECT \* FROM products WHERE cat='$cat' ORDER BY id DESC";

$result = mysql\_query($query);

$result = db\_result\_to\_array($result);

return $result;}

function get\_cat() {

db\_connect1();

$query = "SELECT \* FROM categories ORDER BY id DESC";

$result = mysql\_query($query);

$result = db\_result\_to\_array($result);

return $result;}

function get\_product($id) {

db\_connect1();

$query = ("SELECT \* FROM products WHERE id='$id'");

$result = mysql\_query($query);

$row = mysql\_fetch\_array($result);

return $row;}

?>

Скрипт **cart\_fns.php** предназначен для работы с данными корзины. Содержит функции добавления блюда в корзину **add\_to\_cart($id)**, функцию обновления данных корзины **update\_cart()**, функции **total\_items()** и **total\_price()** для подсчета общего количества и стоимости блюд в корзине.

<?

function add\_to\_cart($id) {

if(isset($\_SESSION['cart'][$id])) {

$\_SESSION['cart'][$id]++;

return true;

}

else {

$\_SESSION['cart'][$id] = 1;

return true;

}

return false;

}

function update\_cart() {

foreach($\_SESSION['cart'] as $id -> $qty)

if($\_POST['$id']== '0') {

unset($\_SESSION['cart'][$id]);

}

else {

$\_SESSION['cart'][$id]=$\_POST[$id];

}

}

function total\_items($cart) {

$num\_items = 0;

if(is\_array($cart)) {

foreach($cart as $id => $qty) {

$num\_items += $qty;

}

}

return $num\_items;

}

function total\_price($cart) {

$total\_price = 0.0;

db\_connect1();

if(is\_array($cart)) {

foreach($cart as $id => $qty) {

$query = "SELECT price FROM products WHERE id='$id'";

$result = mysql\_query($query);

if($result) {

$item\_price = mysql\_result ($result, 0,'price');

$total\_price += $item\_price \* $qty;

}

}

}

return $total\_price;

}

?>

**Контрольные вопросы**

1. Каким образом вставить конструкции PHP в HTML-документ?

2. Каким образом обеспечить, чтобы встречающиеся в строке переменные были заменены их значениями?

3. Каковы правила определения функций в языке PHP?

4. Каковы особенности передачи значений переменных из HTML-формы в переменные PHP?

5. Каким образом осуществляется взаимодействие с базами данных в языке PHP?

6. В чем заключается технология "cookies"?

**Задания**

Создать по вариантно веб-портал с использованием следующих баз данных:

1. Кинотеатр;
2. Супермаркет;
3. Автомагазин;
4. Аптека;
5. Страхование;
6. Книжный магазин;
7. Автосервис;
8. Мебельный центр;
9. Салон красоты;
10. Фотосалон.