

РАЗРАБОТКА СИСТЕМЫ КОНТРОЛЯ ТОВАРОВ ПЕКАРЕН НА PHP С ИСПОЛЬЗОВАНИЕМ БАЗЫ ДАННЫХ MYSQL С ПОМОЩЬЮ HTML И CSS

DEVELOPMENT OF A BAKERY GOODS CONTROL SYSTEM IN PHP USING A MYSQL DATABASE USING HTML AND CSS

**V. Panferova
I. Murashev**

Summary: The article discusses the process of developing a bakery goods control system through the interaction of PHP with a MySQL database using a local OpenServer server. We also use HTML and CSS to define the structure and styles of documents. The main functions of interaction with the database are considered, such as connection to the database itself, output, input, modification and deletion of information.

Keywords: PHP, MySQL, OpenServer, HTML, CSS, local server.

Панферова Валерия Сергеевна

Российский биотехнологический университет
(РОСБИОТЕХ)

panferval@mail.ru

Мурашев Игорь Дмитриевич

к.т.н., доцент, профессор,

Российский биотехнологический университет
(РОСБИОТЕХ)

murashov_45@mail.ru

Аннотация: В статье рассматривается процесс разработки системы контроля товаров пекарен за счёт взаимодействия PHP с базой данных MySQL с использованием локального сервера OpenServer. Также с помощью HTML и CSS мы задаём структуру и стили документов. Рассмотрены основные функции взаимодействия с базой данных, такие как подключение к самой базе данных, вывод, ввод, изменение и удаление информации.

Ключевые слова: PHP, MySQL, OpenServer, HTML, CSS, локальный сервер.

В данной работе были рассмотрены процессы взаимодействия с базой данных MySQL посредством отправления информации из созданных форм для контроля и учёта товаров различных пекарен. Система контроля товаров пекарен необходима, поскольку хлебобулочные изделия имеют небольшой срок годности, поэтому во избежание просроченной продукции, а также жалоб со стороны покупателей необходимо было создать систему, в которой будет храниться вся информация о товарах. В данной статье мы рассмотрим создание

системы контроля с основными функциями взаимодействия с базой данных, такими как добавление информации с помощью созданной формы, вывод информации на главный экран, обновление информации с помощью специальной формы и удаление информации [1].

В первую очередь нам необходимо создать базу данных сети наших пекарен (рис. 1).

В нашей работе действия с информацией будут происходить только с таблицей «Продукция». Но для более

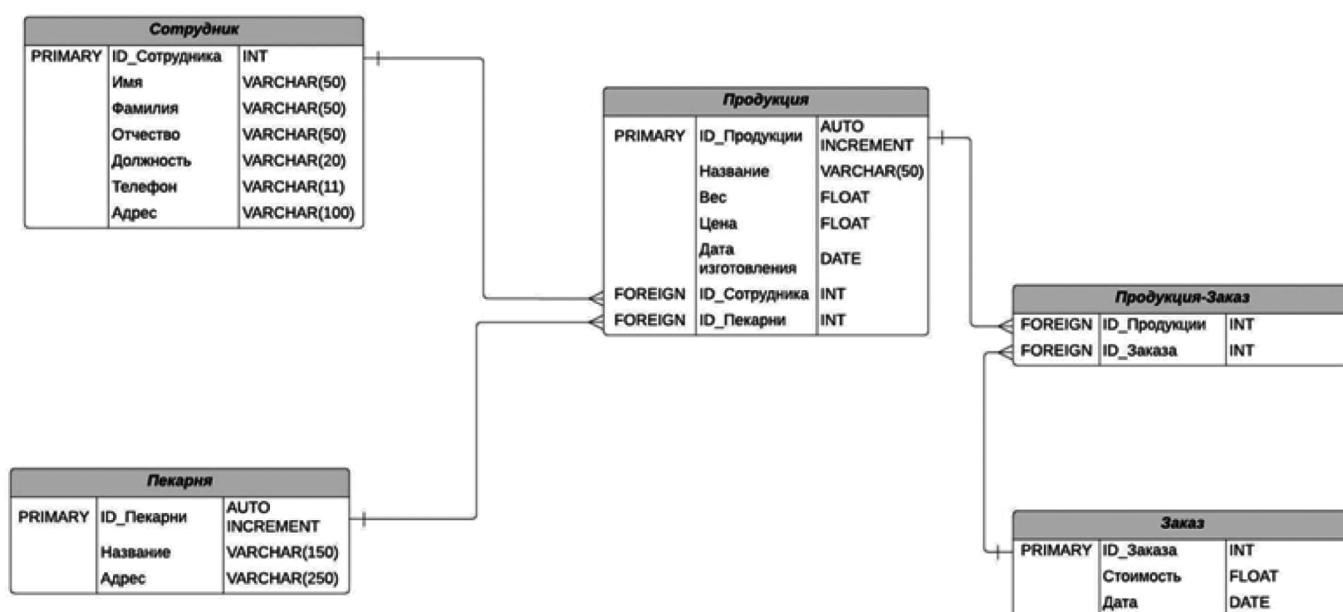


Рис. 1. База данных контроля товаров сети пекарен MySQL

наглядного примера, мы заранее заполним таблицы «Сотрудник» и «Пекарня» какой-либо информацией.

Чтобы реализовать наш проект, мы будем использовать такие инструменты, как: OpenServer [2] (на нём будет храниться созданная нами база данных), PHP (с помощью данного языка программирования мы будем отправлять запросы к базе данных), HTML (данный инструмент мы будем использовать для конструкции страницы), CSS (мы будем использовать для создания стилей страницы).

В первую очередь нам необходимо создать проект в папке проектов OpenServer (рис. 2). OpenServer является программным обеспечением, которое находится в открытом доступе в интернете, а также является бесплатным и удобным в использовании.

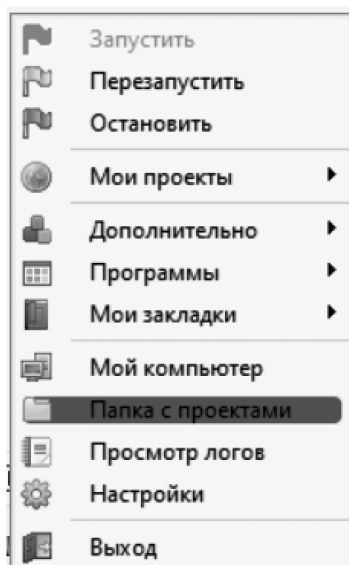


Рис. 2. Меню OpenServer при его запуске

Назовём папку проекта «Bakery». В папке проекта создадим ещё одну папку с названием config. В ней создадим файл connect.php (рис. 3). В данном файле будет храниться подключение к нашей базе данных [3].

```
<?php
$connect=mysqli_connect('localhost','root','','Bakery');
if(!$connect) {
    die('Ошибка подключения');
}
?>
```

Рис. 3. Подключение к базе данных в файле connect.php

В дальнейшем, если в каком-либо другом файле будет присутствовать взаимодействие с базой данных, будет достаточно прописать функцию include(«config/connect.php»); Подключение к базе данных лежащей на локальном сервере происходит за счёт функции mysqli_connect(). В данную функцию необходимо передать 4 параметра в качестве строк, такие как: localhost (является классическим названием уточнения сервера), имя пользователя

(при использовании OpenServer при любой доступной базе данных, имя пользователя и пароль для доступа к определённой базе данных описаны в документации в OpenServer, поэтому новых пользователей создавать не нужно), пароль пользователя, а также наименование самой базы данных, лежащей на сервере. В случае ошибки подключения выводится соответствующее сообщение.

Создаём файл index.php. Он будет являться главной страницей, на которой мы будем выводить всю информацию о товарах (из таблицы «Продукция» в базе данных), а также на ней будет форма для добавления новой информации.

В первую очередь напишем саму структуру файла, а также вывод информации с помощью таблицы. Структура файла стандартная (для упрощения начального написания страницы можно использовать такой плагин, как Emmet [4]. Такой плагин либо установлен по умолчанию, либо вы можете установить его самостоятельно в большинстве текстовых редакторов). В стандартной структуре изменим некоторые строки. <html lang=»ru»> с помощью атрибута lang мы задаём основной язык страницы. <title>Сеть пекарен</title> задаём название вкладки того или иного браузера. Далее создадим в общей папке файл style.css и сразу же подключим его к нашей главной странице путём написания <link rel=»stylesheet» href=»style.css»> в тэге head.

Создадим таблицу для вывода информации из базы данных (рис. 4).

На рисунке 4 мы обращаемся к некоторой переменной \$products. Её мы зададим в самом начале страницы.

На рисунке 5 мы обращаемся к базе данных к таблице «Продукция» и берём из неё всё содержимое, а на рисунке 4 мы выводим содержимое с помощью foreach.

Для наглядного вывода сотрудников и пекарен (в таблице «Продукция» хранятся только ID) мы осуществляем запросы, где ID в таблице «Продукция» совпадают с ID в таблицах «Сотрудник» и «Пекарня» и выводим всё необходимое. В последних двух столбцах у нас будут храниться ссылки на 2 другие функции: изменение информации и удаление информации. Результат представлен на рис. 6.

Далее нам необходимо создать форму для добавления информации в нашу базу данных (рис. 7).

В форме мы отправляем запросы к бд для того, чтобы можно было выбрать уже существующую пекарню (к которой относится товар) и сотрудника, ответственно за контроль товара. Создадим дополнительный файл create.php (рис. 8), код которого будет выполняться при отправке формы (по нажатию кнопки с типом submit).

```

<table>
<thead>
  <tr>
<th>ID</th>
<th>Название</th>
<th>Вес (г)</th>
<th>Цена (руб)</th>
<th>Дата изготовления</th>
<th>Сотрудник</th>
<th>Пекарня</th>
<th>&#9998;</th>
  <th>&#10006;</th>
</tr>
</thead>
<tbody>
<?php
foreach($products as $item) { ??
  <tr>
<td><?=$item[0]> ??</td>
<td><?=$item[1]> ??</td>
<td><?=$item[2]> ??</td>
<td><?=$item[3]> ??</td>
<td><?=$item[4]> ??</td>
<td><?php $employee=mysqli_query($connect,"SELECT `name`,`surname` FROM `Employees` WHERE `idEmployee`='{ $item[5]}'");
$employee=mysqli_fetch_assoc($employee);
echo $employee['name']." ".$employee['surname']; ??</td>
<td><?php $bakery=mysqli_query($connect,"SELECT `name` FROM `Bakeris` WHERE `idBakery`='{ $item[6]}'");
$bakery=mysqli_fetch_assoc($bakery);
echo $bakery['name']; ??</td>
<td><a href="update.php?id=<?=$item[0]> ??>Обновить</a></td>
<td><a href="delete.php?id=<?=$item[0]> ??>Удалить</a></td>
</tr>
<?php
} ??
</tbody>
</table>

```

Рис. 4. Код таблицы для вывода информации из таблицы «Продукция»

```

$products=mysqli_query($connect,"SELECT * FROM `Products`");
$products=mysqli_fetch_all($products);

```

Рис. 5. Осуществление запроса прочтения из таблицы базы данных

На рисунке 8 представлен запрос на добавление информации в таблицу «Продукция». Мы берём информацию из формы с помощью метода POST и заносим в переменные, далее сами переменные мы записываем в запрос и отправляем запрос на сервер. Результат представлен на рисунке 9.

Далее рассмотрим изменение информации в бд с помощью формы. Для удобства, мы создадим новый файл update.php, в котором будет представлена форма для изменения информации. Для удобства обновления информации, нам необходимо перенести в эту форму значения, которые были в записи до изменения. Для этого

мы задаём ссылку для перехода на страницу с формой для обновления на главном странице и передаём id товара через метод GET <a href="update.php?id=<?=\$item[0]> ??>>Обновить. После нажатия на ссылку, мы переходим на страницу update.php и \$_GET['id'] будет равен id того товара, который мы собираемся обновить [5].

На рисунке 10 мы присваиваем переменной значение id из GET и выполняем запрос. Создаём ассоциативный массив, где будут представлены значения строки, к которой мы обращаемся [6]. Далее нам остаётся просто вывести весь ассоциативный массив путём присваивания всем input и option значения value. Часть кода для примера представлена в рисунке 11

При отправке формы (по нажатию кнопки с типом submit) выполняется код в файле, который мы создадим с названием updt.php. Аналогично коду из файла create.

ID	Название	Вес (г)	Цена (руб)	Дата изготовления	Сотрудник	Пекарня	↖	✕
1	Булочки с сыром	60	40	2023-04-07	Елизавета Пушкина	ПаПан	Обновить	Удалить
2	Слойки с ветчиной и сыром	80	60	2023-04-08	Антон Писоченко	Булочные Фокина	Обновить	Удалить
3	Булочка с корицей	1000	30	2023-04-06	Иван Колечкин	Маковка	Обновить	Удалить
4	Чесночный багет	200	80	2023-04-06	Людмила Рязанкина	Волковский	Обновить	Удалить

Рис. 6. Вывод информации из таблицы «Продукция»

```

<h2>Добавить новый товар</h2>
<form action="create.php" method="post">
  <label for="title">Название товара</label>
  <input type="text" name="title" value="" placeholder="Введите название товара">
  <label for="weight">Вес товара (г)</label>
  <input type="number" name="weight" value="" placeholder="Введите вес товара">
  <label for="price">Цена товара (руб)</label>
  <input type="number" name="price" value="" placeholder="Введите цену товара">
  <label for="date">Дата изготовления</label>
  <input type="date" name="date" value="">
  <label for="employee">Сотрудник, ответственный за контроль</label>
  <select class="" name="employee">
    <?php
    $sql="SELECT * FROM `Employees`";
    $result=mysqli_query($connect,$sql);
    while($row=mysqli_fetch_assoc($result)) {
      $id=$row['idEmployee'];
      $name=$row['name'];
      $surname=$row['surname'];
      $patronymic=$row['patronymic'];
      $post=$row['post'];
      echo "<option value='{ $id }'>{ $name } { $surname } { $patronymic } / { $post }</option>";
    }?>
  </select>
  <label for="bakery">Пекарня, в которую доставлен товар</label>
  <select class="" name="bakery">
    <?php
    $sql="SELECT * FROM `Bakeris`";
    $result=mysqli_query($connect,$sql);
    while($row=mysqli_fetch_assoc($result)) {
      $id=$row['idBakery'];
      $name=$row['name'];
      echo "<option value='{ $id }'>{ $name }</option>";
    }?>
  </select>
  <button type="submit" name="submit">Добавить</button>
</form>

```

Рис. 7. Создание формы для добавления информации в бд

```

<?php
include("config/connect.php");
$title=$_POST['title'];
$weight=$_POST['weight'];
$price=$_POST['price'];
$date=$_POST['date'];
$employee=$_POST['employee'];
$bakery=$_POST['bakery'];
mysqli_query($connect,"INSERT INTO `Products` (`name`,`weight`,`price`,`date`,`idEmployee`,`idBakery`)
VALUES ('$title','$weight','$price','$date','$employee','$bakery')");
header('Location:index.php') ?>

```

Рис. 8. Отправка запроса в базу данных для добавления новой записи в таблице

php мы присваиваем переменным значения из метода POST [7]. Только теперь мы выполняем запрос на UPDATE (рис. 12). Необходимо указывать условие в WHERE, так как в противном случае информация обновится во всей таблице [8].

С помощью функции header мы можем перейти на указанную нами страницу после выполнения запроса [9].

Для оформления формы обновления товаров мы просто подключаем тот же файл style.css в тэге head. Ре-

зультат создания формы для обновления товаров в таблице представлен на рисунке 13.

Добавить новый товар

Рис. 9. Вид формы на добавление информации в базу данных

Обновить товар

Рис. 13. Вид формы на обновление информации в базе данных

Также нам необходимо реализовать удаление товаров из таблицы [10]. Для этого нам не надо создавать

```
$product_id=$_GET['id'];
$product=mysqli_query($connect,"SELECT * FROM `Products` WHERE `idProducts`='$product_id'");
$product=mysqli_fetch_assoc($product);
```

Рис. 10. Отправка запроса на чтение записи из таблицы

```
<form action="updt.php" method="post">
<input type="hidden" name="id" value="<?=$product_id ?>">
<label for="title">Название товара</label>
<input type="text" name="title" value="<?=$product['name'] ?>" placeholder="Введите название товара">
<label for="weight">Вес товара (г)</label>
<input type="number" name="weight" value="<?=$product['weight'] ?>" placeholder="Введите вес товара">
<label for="price">Цена товара (руб)</label>
<input type="number" name="price" value="<?=$product['price'] ?>" placeholder="Введите цену товара">
<label for="date">Дата изготовления</label>
<input type="date" name="date" value="<?=$product['date'] ?>">
```

Рис. 11. Форма для изменения информации о продукции в бд

```
<?php
include("config/connect.php");
$title=$_POST['title'];
$weight=$_POST['weight'];
$price=$_POST['price'];
$date=$_POST['date'];
$employee=$_POST['employee'];
$bakery=$_POST['bakery'];
$id=$_POST['id'];
mysqli_query($connect,"UPDATE `Products` SET `name`='$title',`weight`='$weight',`price`='$price',
`date`='$date',`idEmployee`='$employee',`idBakery`='$bakery' WHERE `idProducts`='$id'");
header('Location:index.php');
?>
```

Рис. 12. Отправка запроса на сервер для изменения информации в бд

```
<?php
include("config/connect.php");
$product_id=$_GET['id'];
mysqli_query($connect,"DELETE FROM `Products` WHERE `idProducts`='$product_id'");
header('Location:index.php'); ?>
```

Рис. 14. Отправление запроса на сервер для удаление записи из таблицы

дополнительную форму, так как нам отображается вся информация, которую мы собираемся удалить. Удаление будет происходить по нажатию ссылки на главной странице и передачи id товара в метод GET `>Удалить`. Код удаления товара будет содержаться в новом файле, который мы создадим под названием delete.php.

На рисунке 14 представлена программа для удаления товара из таблицы. В переменную передаём значение id товара, который собираемся удалить, а далее выполняем удаление с помощью запроса на DELETE. С помощью функции header возвращаемся на главную страницу.

В данном проекте мы реализовали все 4 основные вида взаимодействия с базой данных, такие как вывод информации с помощью SELECT, добавление новой информации с помощью INSERT INTO, обновление через UPDATE и удаление DELETE. Для осуществления любых действий к бд мы выполнили подключение к локальному серверу. Данный проект можно развивать и дальше, добавляя взаимодействия с другими таблицами бд, ставя проверку на заполнение формы и выводе ошибок пользователю, добавление новых таблиц в базу данных, преобразование страниц с помощью добавления стилей и т.д.

ЛИТЕРАТУРА

1. Дюбуа П. MySQL. Сборник рецептов. — СПб: Символ-Плюс, 2006. — 1056 с.
2. Документация // OpenServer URL: <https://ospanel.io/docs/> (дата обращения: 10.04.2023).
3. mysqli_connect // PHP URL: <https://www.php.net/manual/en/function.mysqli-connect.php> (дата обращения: 11.04.2023).
4. Как плагин Emmet помогает ускорить работу с программным кодом // Хабр URL: <https://habr.com/ru/articles/668476/> (дата обращения: 15.04.2023).
5. Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов. — СПб: ООО «Диалектика», 2020. — 464 с.
6. Arrays // php URL: <https://www.php.net/manual/en/language.types.array.php> (дата обращения: 12.04.2023).
7. Работа с формами // php URL: <https://www.php.net/manual/ru/tutorial.forms.php> (дата обращения: 13.04.2023).
8. Фильтрация данных. Оператор WHERE // Metanit.com URL: <https://metanit.com/sql/mysql/3.3.php> (дата обращения: 13.04.2023).
9. header // php URL: <https://www.php.net/manual/ru/function.header.php> (дата обращения: 14.04.2023).
10. Удаление данных в MySQLi // Metanit.com URL: <https://metanit.com/php/mysql/3.6.php> (дата обращения: 16.04.2023).

© Панферова Валерия Сергеевна (panferval@mail.ru); Мурашев Игорь Дмитриевич (murashov_45@mail.ru).
Журнал «Современная наука: актуальные проблемы теории и практики»