

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ
«Университет «Дубна»

ИНСТИТУТ СИСТЕМНОГО АНАЛИЗА И УПРАВЛЕНИЯ
КАФЕДРА СИСТЕМНОГО АНАЛИЗА И УПРАВЛЕНИЯ

Отчёт

по дисциплине

«Разработка приложений баз данных»

ТЕМА: «Разработка веб-приложения «Каталог товаров из интернет магазина»
(наименование темы)

Выполнил: студент группы 3012

Киселёв Артём Сергеевич
(ФИО)

Руководитель:

Иванцова Ольга Владимировна
(ФИО)

Дубна, 2022

Постановка задачи

Реализовать приложение «Каталог товаров из интернет магазина» для взаимодействия с базой данных (далее БД), которое включает в себя подключение к БД, удобное переключение между таблицами, просмотр и обработку данных. Готовая схема БД представлена на рис. 1.

Функционал приложения:

- Отображение новых поставок и продаж интернет-магазина;
- Взаимодействие с товарами (изменение цены, удаление из каталога, добавление новых данных);
- Отображение персонала интернет-магазина;
- Взаимодействие с персоналом (удаление, изменение и добавление новых сотрудников).

Минимальные требования для приложения:

- Приложение должно иметь минимум 6 таблиц;
- Приложение должно включать в себя создание, чтение, изменение и удаление данных;
- Приложение должно иметь минимум 1 триггер и 1 функцию;
- Приложение должно иметь простой и понятный для пользователя интерфейс;
- Приложение должно содержать в себе сортировку данных;
- Приложение должно содержать функционал с различными правами доступа.

Этапы работы:

- Сделать модель БД;
- Создать БД на локальном сервере и заполнить все таблицы;
- Реализовать *CRUD* приложение для взаимодействия с БД.

Создание модели базы данных

За основу модели для приложения «Каталог товаров из интернет магазина» было решено взять схему БД торговое предприятие с корректировкой некоторых таблиц. Готовая схема БД представлена на рисунке ниже.

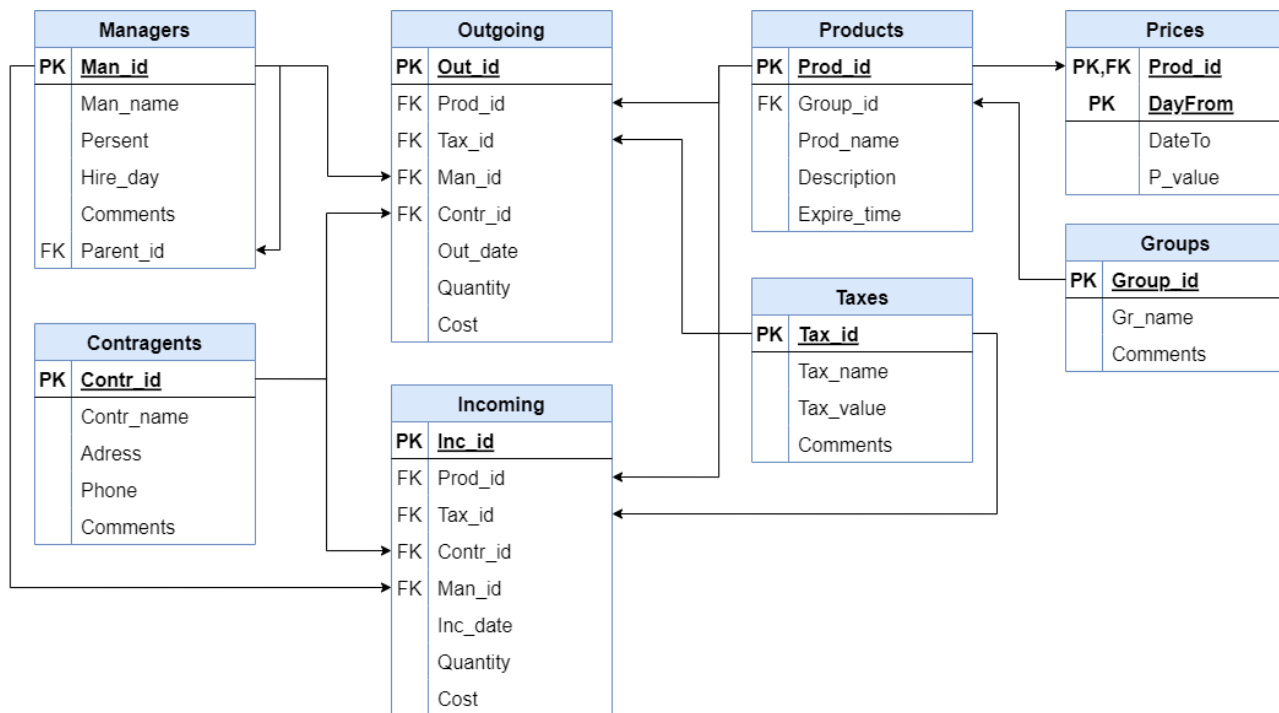


Рис. 1. Схема БД для приложения «Каталог товаров из интернет магазина»

Реализация серверной части

В качестве основного инструмента для создания приложения было решено взять свободную реляционную систему управления базами данных *MySQL*. В качестве кроссплатформенной сборки локального веб-сервера использовался *XAMPP*, для создания клиентской части приложения – *HTML* + *CSS*, для создания серверной части – *PHP*.

В первую очередь необходимо создать базу данных по модели, приведённой выше. Сделать это было решено через синтаксис языка *PHP*.

```
//create BD
$bd_connect->query("
    CREATE DATABASE IF NOT EXISTS applicationbd
");
$bd_connect = new mysqli($host, $user_bd, $pass_bd, $name_bd);
//table Users
$bd_connect->query("
    CREATE TABLE IF NOT EXISTS users (
        user_id INT(5) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        email VARCHAR(100) not null,
        login VARCHAR(30) not null,
        password VARCHAR(50) not null,
        grade VARCHAR(30) default 'viewer'
    )
");
```

Рис. 2. Создание БД и таблицы *Users*.

На рисунке выше представлено создание базы данных «*Applicationbd*» и одной из таблиц с названием «*Users*». Приложение содержит следующие таблицы:

- Users;
- Contragents;
- Groups;
- Taxes;
- Managers;
- Products;
- Prices;
- Incoming;
- Outgoing.

Взаимодействие с сервером происходит при помощи подключения *mysqli*, которое содержит имя хоста, имя пользователя, пароль и имя базы данных как атрибуты. Само подключение к серверу выглядит следующим образом (см. рис. 3).

```
<?php
$host = "localhost";
$user_bd = "root";
$pass_bd = "";
$name_bd = "applicationbd";

$bd_connect = new mysqli($host, $user_bd, $pass_bd);
$sql = mysqli_query($bd_connect, "show databases like 'applicationbd'");
$result = mysqli_fetch_array($sql);

if(!isset($result[0])) include("bd_create.php");
else $bd_connect = new mysqli($host, $user_bd, $pass_bd, $name_bd);
```

Рис. 3. Создание подключения к серверу

Изначально подключение имеет лишь три атрибута без названия БД. Далее код предусматривает два варианта развития событий:

- БД с именем «*Applicationbd*» существует;
- БД с именем «*Applicationbd*» не существует.

Если программа нашла БД на сервере, то к подключению добавляется параметр её имени. Если же программе не удалось найти нужную БД, то автоматически происходит её создание, которое включает в себя создание всех таблиц, вторичных ключей, функций, триггеров и первоначального заполнения полей созданных таблиц. Примеры создания представлены на рисунках ниже.

```
//primary keys
$bd_connect->query("
    alter table Managers add constraint managers_k2 foreign key (Parent_id) references Managers(Man_id)
");
```

Рис. 4. Создание вторичного ключа для таблицы “*Managers*”

```

//functions
//возвращает кол-во parentId у менеджеров
$bd_connect->query("
    create function ParentIdCount_func(P_id int)
    returns int
    begin
        declare ParentIdCount int default 0;
        select count(Parent_id) into ParentIdCount from Managers
        where Parent_id = P_id;
        return ParentIdCount;
    end;
");

```

Рис. 5. Создание функции, возвращающей количество подчинений у менеджеров

Приложение содержит следующие функции:

- Возвращает кол-во подчинённых у менеджера;
- Возвращает количество товара.

```

//triggers
//Запретить удаление менеджера с parentId != null
$bd_connect->query("
    create trigger `manDelete`
    before delete on `Managers`
    for each row
    begin
        if (OLD.Parent_id is not NULL) then
            signal sqlstate '45000' set message_text = 'manDeleteTrigger Error: Trying to delete manager with parent id';
        end if;
    end;
");

```

Рис. 6. Создание триггера, запрещающего удалять менеджера с подчинением другому менеджеру

Приложение содержит следующие триггеры:

- Запрет на удаление менеджера с подчинением;
- Запрет на добавление/изменение менеджера, если у *parentId* > 2 подчиненных;
- Запрет на продажу товаров, которых нет в наличии;
- Запрет на удаление товаров, которые есть в наличии;
- Запрет на установку *DayFrom* < *DateTo* в таблице *Prices*.

```

#region fillbd
//fill bd
$db = new PDO("mysql:host=$host;dbname=$name_bd", $user_bd, $pass_bd);
try{
    $dir = $url = __DIR__.'./fillbd.sql';
    $sql = file_get_contents($dir);
    $qr = $db->exec($sql);
} catch (Exception $e){
    echo "<script>console.log(\"{$e->getMessage()}\")</script>";
}
$db = null;
#endregion

```

Рис. 7. Заполнение таблиц

Подробнее остановимся на заполнении таблиц, представленном на рис. 7. Здесь создаётся ещё одно независимое подключение, только теперь *PDO*, а не объектно-ориентированного типа. После этого в блоке *try* происходит последовательное выполнение команд из файла *fillbd.sql* (структура файла представлена на рис. 8). Если при выполнении команд возникла ошибка, пользователю выводится сообщение. Происходит это в блоке обработки ошибок *catch*. В конце происходит «зануление» подключения для очистки памяти.

```

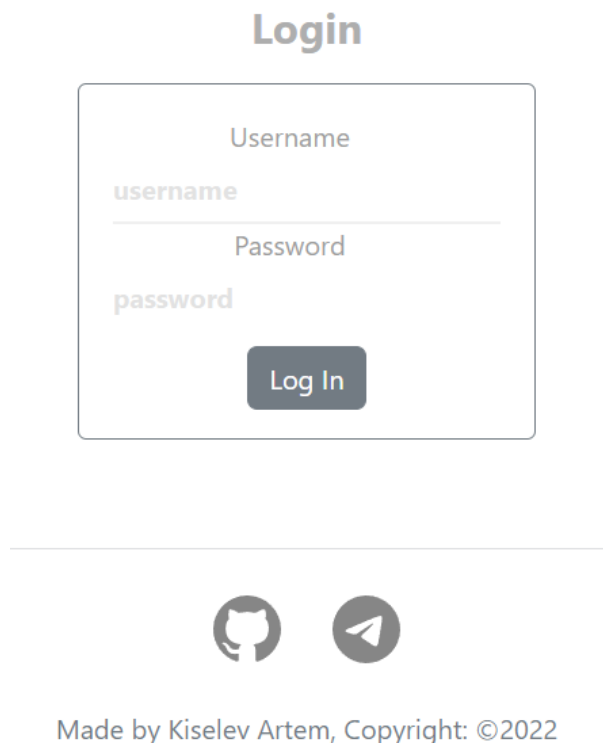
insert into applicationbd.Contragents (Contr_name, Adress, Phone, Comments) values
('Александра Ивановна', 'Москва, ул. Московская', '89998887777', 'сотрудник месяца'),
('Мария Семёновна', 'Москва, ул. Речная', '89981234567', null),
('Пётр Васильевич', 'Дмитров, ул. Советская', '89977654321', null),
('Валерия Ивановна', 'Дубна, пр-кт Мира', '89960000000', 'постоянные опоздания'),
('Алексей Сергеевич', 'Воронеж, ул. Профессиональная', '88005553535', null);

```

Рис. 8. Структура файла *fillbd.sql* на примере добавления данных в таблицу *Contragents*

Реализация клиентской части

После создания серверной части можно приступать к реализации клиентской.



The image shows a login form titled "Login". It contains two input fields: "Username" with the placeholder text "username" and "Password" with the placeholder text "password". Below the fields is a dark button labeled "Log In". At the bottom of the form, there are two circular icons: GitHub and Telegram. Below the icons, the text "Made by Kiselev Artem, Copyright: ©2022" is displayed.

Рис. 9. Форма входа в систему

При первом заходе в приложение пользователя встречает форма авторизации, представленная на рис. 9. При попытке зайти на другой домен (например, на главную страницу) программа автоматически перенаправит пользователя обратно к форме, что делает невозможным просмотр страниц приложения неавторизованными пользователями.

Форма регистрации здесь отсутствует, это сделано специально, чтобы пользователи «со стороны» не смогли регистрировать новые аккаунты и просматривать конфиденциальную информацию приложения. Добавление новых пользователей осуществляет администратор. Остановимся на этом подробнее.

Пользователи приложения имеют два разных грейда: «*Viewer*» и «*Admin*». В зависимости от грейда доступны те или иные функции приложения. Рассмотрим отличия на примере таблицы *Managers*:

Managers

Name	Percent	Hire day	Comment	Parent id
<input type="text"/>	<input type="text" value="0.0010"/>	<input type="text" value="YYYY-MM-DD"/>	<input type="text"/>	<input type="text" value="1"/>

Order by: Manager id ascending

Manager id	Name	Percent	Hire day	Comment	Parent id	Change	Delete
1	Светлана Артёмовна	0.1500	2001-01-01			<input type="button" value="Change"/>	<input type="button" value="Delete"/>
2	Юлия Сергеевна	0.1050	2002-10-10	Выписать премию		<input type="button" value="Change"/>	<input type="button" value="Delete"/>
3	Наталья Семёновна	0.1200	2005-04-05		1	<input type="button" value="Change"/>	<input type="button" value="Delete"/>
4	Варвара Петровна	0.0500	2012-12-10	Работник месяца	3	<input type="button" value="Change"/>	<input type="button" value="Delete"/>
5	Елизавета Макаровна	0.0300	2013-03-09		1	<input type="button" value="Change"/>	<input type="button" value="Delete"/>
6	Антон Сергеевич	0.0400	2014-04-06	Уволен		<input type="button" value="Change"/>	<input type="button" value="Delete"/>

Рис. 10. Страница таблицы Managers от лица пользователя с грейдом «Admin»

Order by: Manager id ascending

Manager id	Name	Percent	Hire day	Comment	Parent id
1	Светлана Артёмовна	0.1500	2001-01-01		
2	Юлия Сергеевна	0.1050	2002-10-10	Выписать премию	
3	Наталья Семёновна	0.1200	2005-04-05		1
4	Варвара Петровна	0.0500	2012-12-10	Работник месяца	3
5	Елизавета Макаровна	0.0300	2013-03-09		1
6	Антон Сергеевич	0.0400	2014-04-06	Уволен	

Рис. 11. Страница таблицы Managers от лица пользователя с грейдом «Viewer»

Можно заметить, что у пользователя с грейдом «Viewer» недоступна форма добавления, а также кнопки удаления и изменения данных. Доступны только просмотр данных таблицы и их сортировка (см. рис. 12).

Order by: Manager id ascending

Comment	Manager id	Name	Percent	Hire day	Parent id
---------	------------	------	---------	----------	-----------

Рис. 12. Форма сортировки данных

Сама форма сортировки представляет собой два всплывающих списка:

- Выбор столбца сортировки;

- Выбор вида сортировки (по возрастанию или по убыванию);

После нажатия на кнопку apply сортировка данных произойдёт автоматически.

Рассмотрим отличия для страницы *Database*. Сама страница содержит кнопки перехода к тем или иным таблицам базы, а также схему самой БД (см. рис. 13). Отличие здесь заключается лишь в одной кнопке: переход на таблицу *Users*. Для пользователя с грейдом «Viewer» эта кнопка недоступна, так как эта таблица является основной для хранения информации пользователей (логин, эл. почта, пароль и т. д.) и, как следствие, самой защищённой в приложении. При попытке перейти по домену таблицы *Users* пользователь будет мгновенно переадресован.

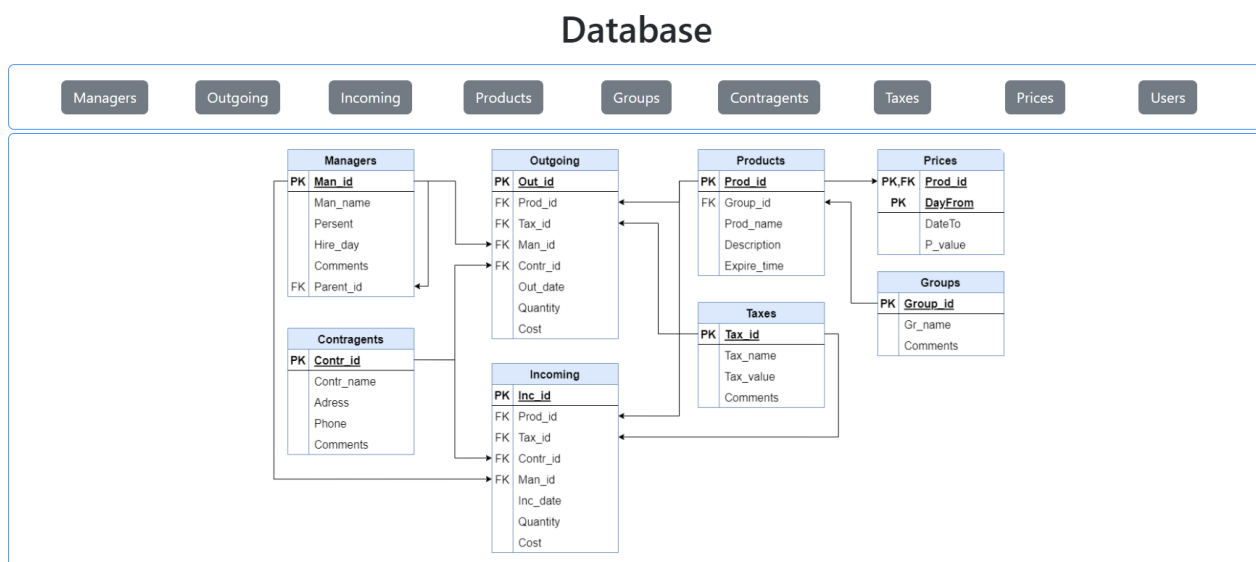


Рис. 13. Страница Database

Рассмотрим страницу редактирования данных на примере таблицы *Managers* (см. рис. 14). Здесь форма изменения имеет те же названия столбцов, что и сама таблица. Поле первичного ключа (в данном случае поле *Manager id*) редактировать невозможно. Оно присутствует для информирования пользователя. При попытке внесения некорректных данных, приложение выдаёт пользователю ошибку (см. рис. 15). В данном случае ошибку вызвал триггер *manUpdateTrigger*, который запрещает добавлять подчинение к менеджеру, у которого и так больше двух подчинений.

Update data

Manager id	Name	Percent	Hire day	Comment	Parent id
1	Светлана Артёмовна	0.1500	2001-01-01		

Рис. 14. Страница обновления данных таблицы *Managers*

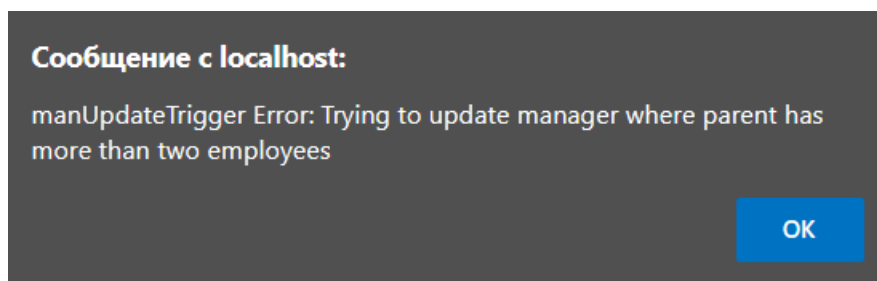


Рис. 15. Ошибка `manUpdateTriggerError`

Рассмотрим последнюю из основных страниц приложения, а именно страницу *Console* (см. рис. 16).

Console

Write a request in the text window

Рис. 16. Страница *Console*

Здесь всё также, в зависимости от грейда, доступны те или иные функции:

Для грейда «*Viewer*» доступны:

- Операторы *select*, кроме выбора данных из таблицы *Users*;

Для грейда «*Admin*» доступны:

- Операторы *select*;
- *DML* операторы (*insert*, *update*, *delete*);

Рассмотрим пример запроса к таблице *Managers*:

Write a request in the text window

```
select * from managers
```

Perform

Man_id	Man_name	Persent	Hire_day	Comments	Parent_id
1	Светлана Артёмовна	0.1500	2001-01-01		1
2	Юлия Сергеевна	0.1050	2002-10-10	Выписать премию	
3	Наталья Семёновна	0.1200	2005-04-05		1
4	Варвара Петровна	0.0500	2012-12-10	Работник месяца	3
5	Елизавета Макаровна	0.0300	2013-03-09		1
6	Антон Сергеевич	0.0400	2014-04-06	Уволен	

Рис. 17. Запрос к таблице Managers

После выполнения запроса ниже кнопки perform появляется таблица с результатами. Если же пользователь использует недоступный оператор, появляется следующая ошибка:

Write a request in the text window

```
delete from managers where man_id = 1
```

You have no permissions to use this request

Рис. 18. Ошибка при попытке удаления данных с таблицы *Managers* пользователем с грейдом «Viewer»

При попытке отправить неизвестную приложению команду, будет напечатана следующая ошибка:

Write a request in the text window

```
delete from managers where man_id = 1
```

You have no permissions to use this request

Рис. 19. Ошибка при попытке отправить неизвестный приложению запрос

Заключение

В результате данного проекта было разработано веб-приложение «Каталог товаров интернет магазина». Все требования функционала приложения, а также минимальные требования были выполнены. В ходе тестирования были выявлены незначительные ошибки, которые сразу же были исправлены.

В ходе разработки были получены новые знания по написанию клиентской части с использованием HTML+CSS, а также серверной части веб-приложения с использованием PHP + MySQL.

В качестве перспектив развития приложения, можно отметить добавление новых групп пользователей с различными правами доступа, добавление пользовательских профилей, а также улучшение консоли путём добавления новых доступных для использования операторов.