МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики

и вычислительной техники

Кафедра информационной безопасности

Отчёт к курсовому проекту

по дисциплине “Безопасность систем баз данных”

**Разработка базы данных для службы доставки**

Выполнили: студенты группы БИ-31

Храбрых А.П.,  
Лоскутова О.И.,  
Загайнова М.В.

Проверил: доцент кафедры

ИБ Сучков Д.С.

Йошкар-Ола

2019 г.

**СОДЕРЖАНИЕ**

**Введение**3

**1. Техническое задание4**

1.1 Требования к курсовой работе4

1.2 Требования к базе данных4

1.3 Требования к API (минимальное количество реализованных методов)2

**2. Порядок выполнения работы5**

2.1 Этапы разработки базы данных5-7

2.2 Этапы разработки API8-11

**3. Приложения12**

3.1 ER-диаграмма12

3.2 Ссылка на github.com12

**4. Вывод12**

**Введение**

В курсовой работе рассматривается создание базы данных, предназначенной для автоматизации работы службы доставки цветов. База данных позволяет клиентам просматривать ассортимент, состав букетов. Также реализована автоматизация оформления заказа, которая позволяет клиентам формировать заказ.

## **Техническое задание**

## ***1.1 Требования к курсовой работе:***

* Получить структуру данных из файла, согласно варианта. Привести к 3й нормальной форме. Добавить недостающие таблицы.
* Составить ER-диаграмму
* Разработать API для базы данных на любом языке, выполняющемся на стороне сервера (php, ASP.NET, Java, python, node.js, etc)
* Взаимодействие должно осуществляться по клиент-серверной архитектуре, подключение с клиентской программы недопустимо
* Провести настройку пользователей базы данных для разграничения прав доступа, привести пример конфигурации
* Все документы и исходные коды для курсовой работы должны храниться под контролем системы контроля версий — git или mercurial (https://github.com/, https://bitbucket.org/)
* Во время сдачи курсового проекта необходимо предоставить отчет о проделанной работе в печатном виде (отчет)

## ***1.2 Требования к базе данных***

* Наличие не менее 7 таблиц, в том числе таблицы сессий и пользователей
* Структура таблицы должна содержать не менее 3-х полей, одно из которых ключевое
* Правомерное использование типов данных
* Обязательно использование триггеров и/или хранимых процедур
* Форма нормализации не менее 3NF
* Индексирование по полям поиска

## **1.3 Требование к API (минимальное количество реализованных методов)**

* аутентификация пользователя (создание сессии);
* добавление/удаление/изменение данных в таблицах;
* выборка данных их ключевых таблиц по запросам;
* выборка данных из таблиц с объединением результатов.

1. **Порядок выполнения работы**
   1. Этапы разработки базы данных

Разработана база данных, содержащая 8 таблиц. В том числе таблицы сессий (*flower\_session*) и пользователей (*flower\_users*). Для авторизации пользователя используется таблица - *flower\_accounts,* содержащая информацию об аккаунте пользователя. Для хранения информации об ассортименте используются следующие таблицы:

*flower\_view* – название и цена букета;

*flower\_structure –* состав букета (наполнение букета – вид цветов, упаковка);

Для автоматизации оформления заказа, были созданы следующие таблицы:

*flower\_availability –* баланс и платежные данные покупателя;

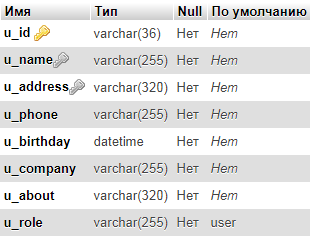
*flower\_cart* – информация о пиццах, добавленных в корзину;

*flower\_orders* – оформленные заказы на доставку пиццы;

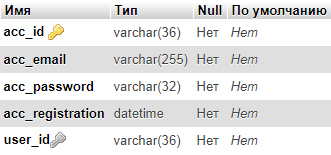
Структуры, реализованных таблиц:



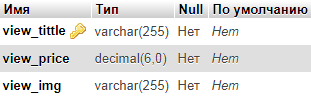
*Рис 1. flower\_session*

**

*Рис 2. flower\_users*

**

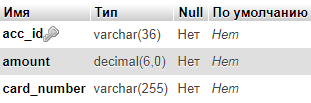
*Рис 3. flower\_accounts*

**

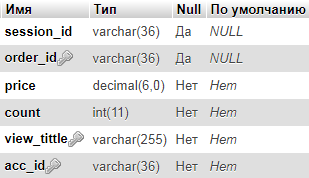
*Рис 4. flower\_view*

**

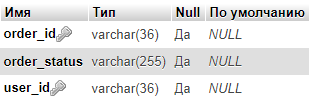
*Рис 5. flower\_structures*

**

*Рис 6. flower\_availability*

**

*Рис 7. flower\_cart*

**

*Рис 8. flower\_orders*

Используемые триггеры:

* *drop\_order –* триггер отвечающий за удаление товаров из корзины;

*DROP TRIGGER IF EXISTS `drop\_order`;*

*DELIMITER $$*

*CREATE TRIGGER `drop\_order` BEFORE DELETE ON `flower\_orders` FOR EACH ROW BEGIN*

*DELETE FROM ` flower\_cart`*

*WHERE ` flower\_cart`.order\_id = OLD.order\_id;*

*END*

*$$*

*DELIMITER;*

* *generate\_id –* триггер отвечающий за генерацию id аккаунта;

DROP TRIGGER IF EXISTS `*generate\_id*`;

DELIMITER $$

CREATE TRIGGER `*generate\_id*` BEFORE INSERT ON ` *flower\_accounts*` FOR EACH ROW BEGIN

IF ISNULL(NEW.acc\_id) THEN

SET NEW.acc\_id = UUID();

END IF;

END

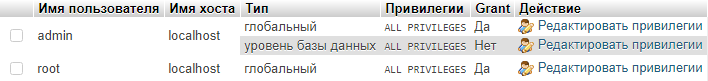
$$

DELIMITER;

Проведена настройка пользователей базы данных для разграничения прав доступа:



*Рис 9. пользователи базы данных*

**

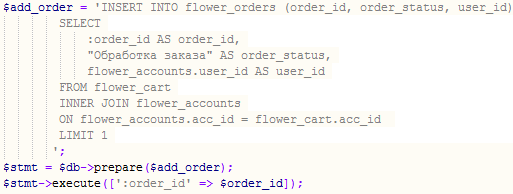
*Рис 10. привилегии пользователя admin*

2.2 Этапы разработки API

Было разработано API для аутентификации пользователя, состоящий из 3 функций, написанных на языке PHP.

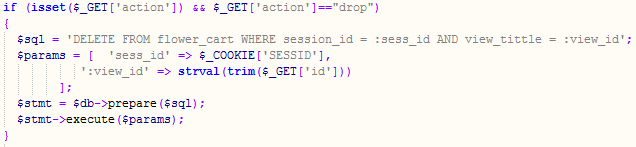


Разработана функция добавления, удаления и изменения данных.



Добавлена функция вывода и объединения данных из ключевых таблиц.

Рис.14 Добавление и изменение данных



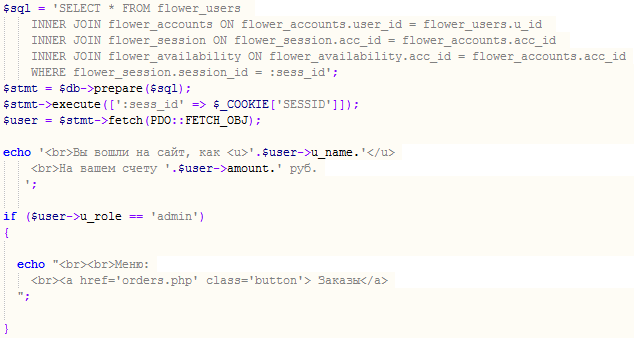


Рис.16 Вывод данных по заказам

Рис.15 Удаление данных

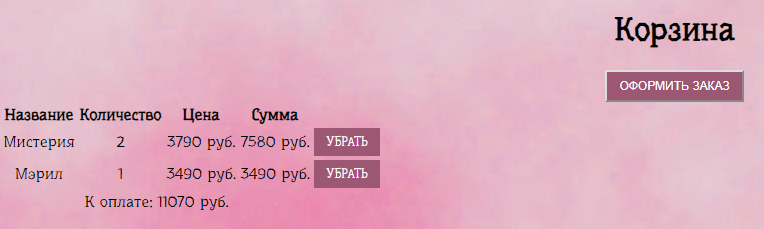


Рис.17 Корзина покупателя

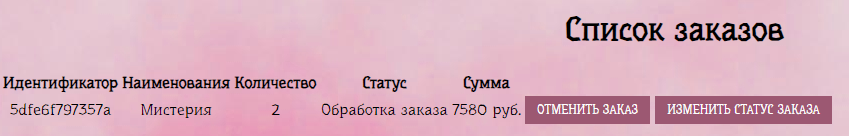


Рис.18 Список заказов (доступен только администратору)

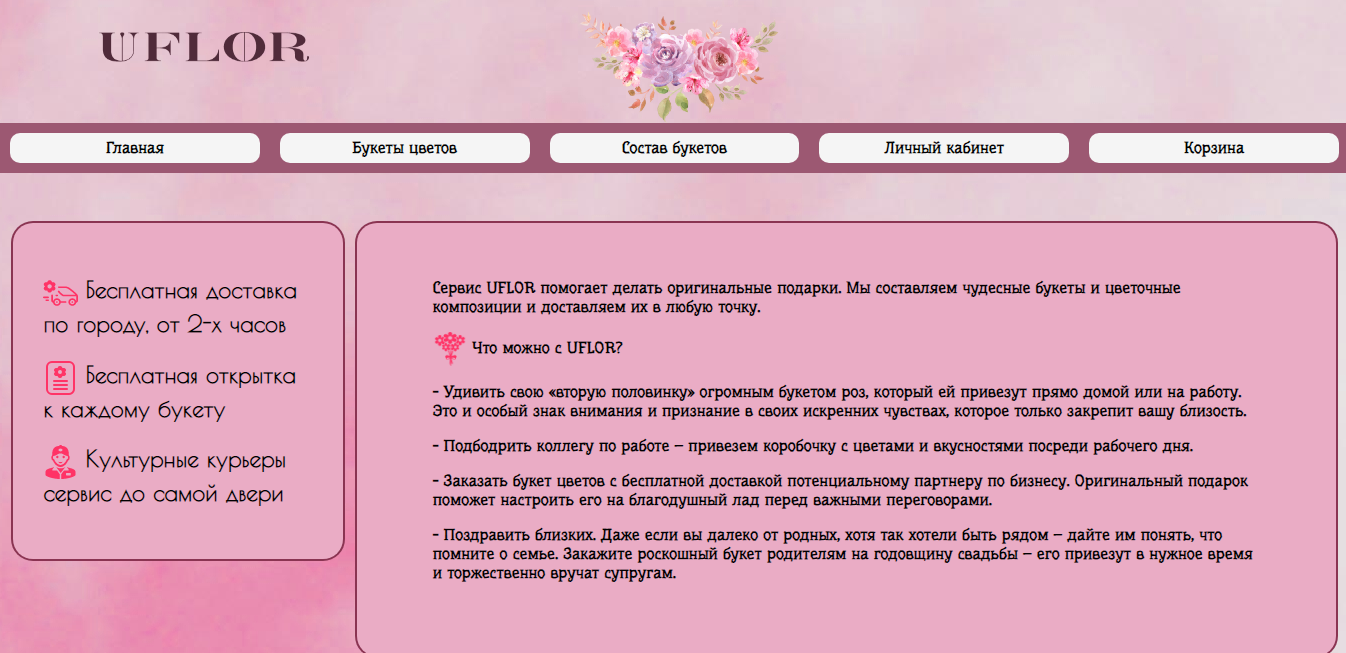


Рис.19 Главная страница

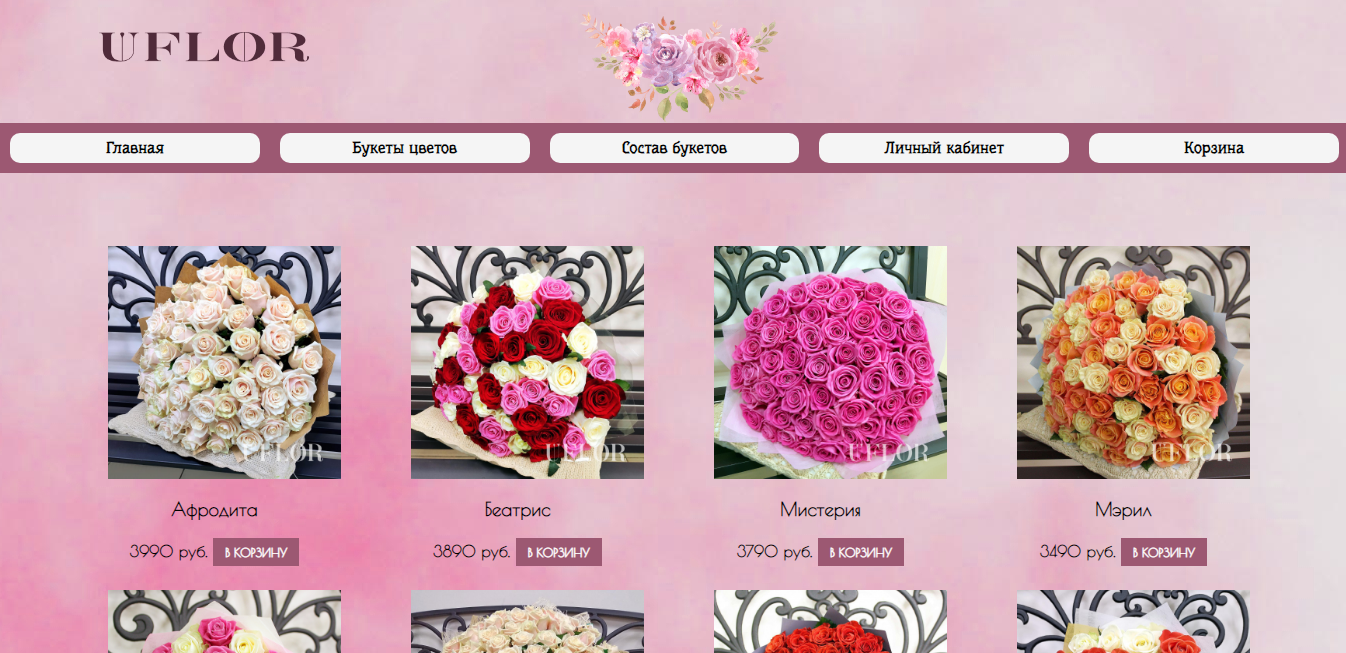


Рис.20 Ассортимент букетов

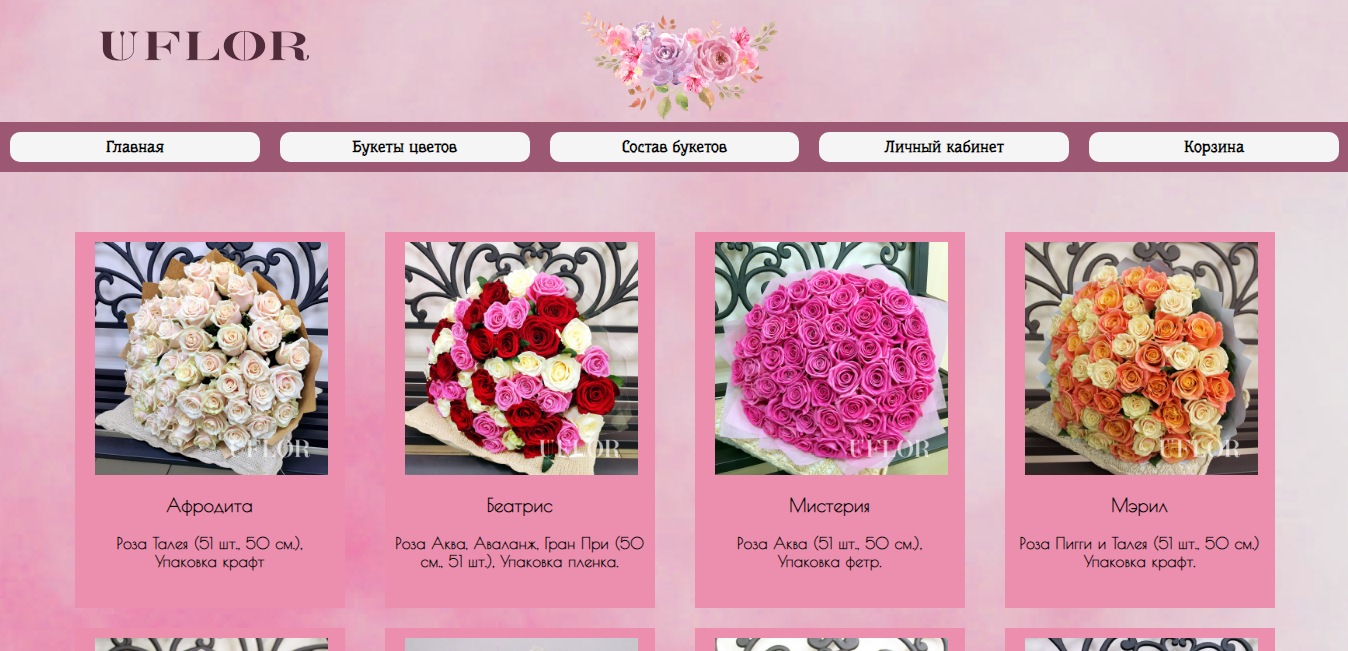
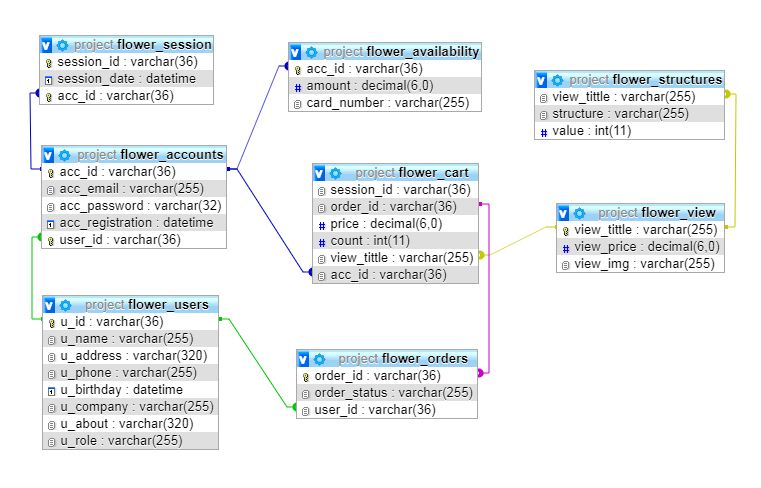


Рис.21 Состав букетов

**Приложения**

1. ER-диаграмма



1. Исходные коды и документы: <https://github.com/Murrao/bsbd_project_2019>

**Вывод**

Во время выполнения курсового проекта были изучены методы работы с базами данных, способы управления. Таким образом, в ходе работы была разработана база данных, а также API, для автоматизации работы службы доставки.