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

Отчет к курсовому проекту

по дисциплине «Безопасность систем баз данных»

**Разработка базы данных кондитерской фирмы**

Вариант № 17

Выполнили: студенты группы БИ-31

Иконников К.М., Тюлькин А.А., Мокеев И.А.

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)4

**2. Порядок выполнения работы5**

2.1 Этапы разработки базы данных5-8

2.2 Этапы разработки API8

2.3 Скриншоты программы8-9

**3. Приложения10**

3.1 ER-диаграмма10

3.2 Ссылка на github.com10

**4. Вывод10**

**Введение:** в курсовой работе рассматривается создание базы данных, предназначенной для автоматизации работы кондитерской фирмы. База данных позволяет клиентам просматривать ассортимент, состав товара. Также администратору взаимодействовать с базой пользователей.

1. **Техническое задание**

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

* Получить структуру данных из файла, согласно варианту. 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.

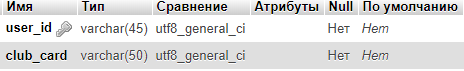
1. **Порядок выполнения работы**
   1. Этапы разработки базы данных

Разработана база данных, содержащая 12 таблиц. В том числе таблицы сессий (*sessions*) и пользователей (*people*). Для авторизации пользователя используется таблица – *people,* содержащая информацию об аккаунте пользователя. Для хранения информации об ассортименте используются следующие таблицы:

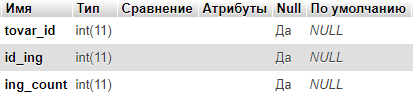
*tovar* – название и цена товара;

*spec\_ingredients –* состав товара (ингредиенты);

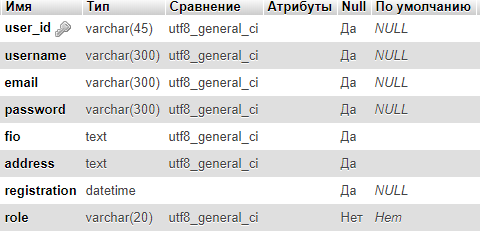
Структуры, реализованных таблиц:



*Рис 1. Структура таблицы buyer*



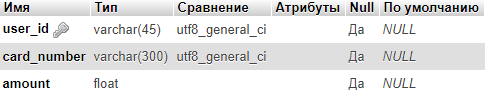
*Рис 2. Структура таблицы ingredients*



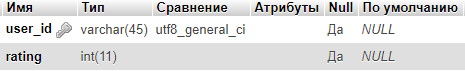
*Рис 3. Структура таблицы people*



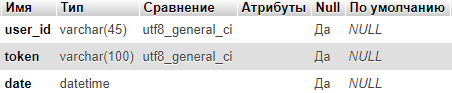
*Рис 4. Структура таблицы people\_info*



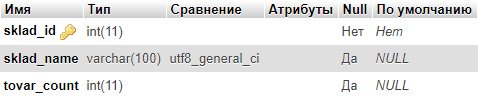
*Рис 5. Структура таблицы people\_info\_two*



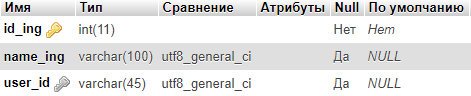
*Рис 6. Структура таблицы provider*



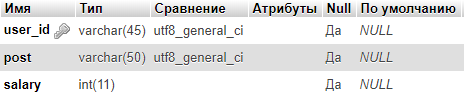
*Рис 7. Структура таблицы sessions*



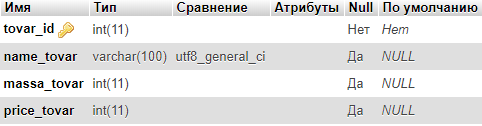
*Рис 8. Структура таблицы sklad*



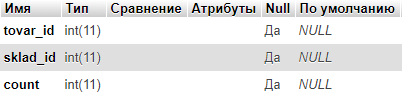
*Рис 9. Структура таблицы spec\_ingredients*



*Рис 10. Структура таблицы staff*



*Рис 11. Структура таблицы tovar*



*Рис 12. Структура таблицы tovar\_on\_sklad*

Используемые триггеры:

* *add\_tovar\_on\_sklad –* триггер отвечающий за добавление товара на склад;

1. BEGIN
2. UPDATE sklad set tovar\_count = tovar\_count - OLD.count;
3. END

* *drop\_tovar\_on\_sklad –* триггер отвечающий за изменения кол-ва товара на складе при удалении товара;

1. BEGIN
2. UPDATE sklad set tovar\_count = tovar\_count - OLD.count;
3. END

* *drop\_ing\_2 –* триггер отвечающий за удаление ингредиентов при удалении поставщика;

1. BEGIN
2. DELETE FROM spec\_ingredients WHERE id\_ing = OLD.id\_ing;
3. END

* *drop\_ing –* триггер отвечающий за удаление ингредиентов при удалении товара;

1. BEGIN
2. DELETE FROM ingredients WHERE tovar\_id = OLD.tovar\_id;
3. END

Проведена настройка пользователей базы данных для разграничения прав доступа:

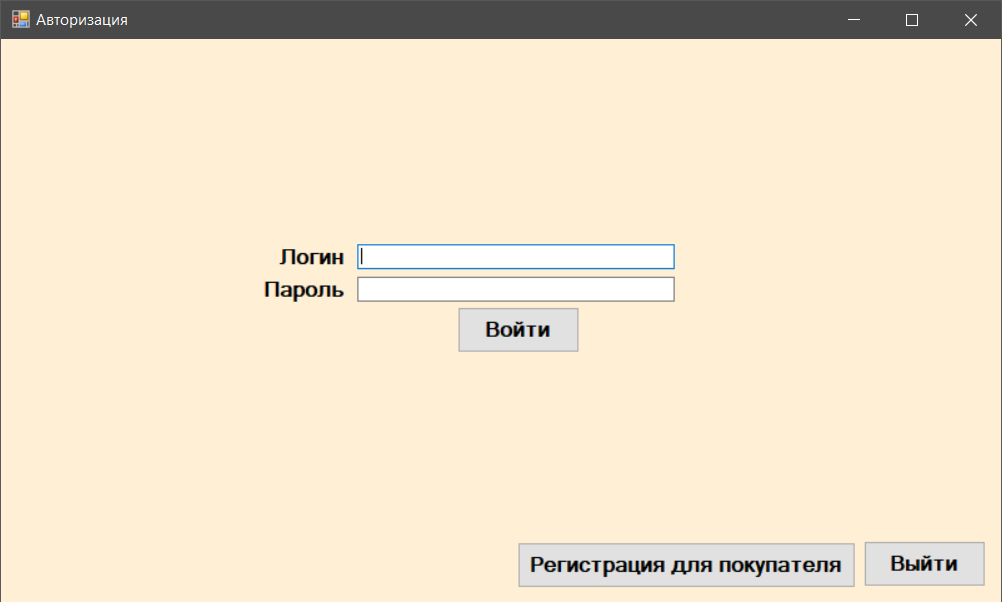


*Рис 13. Структура таблицы tovar*

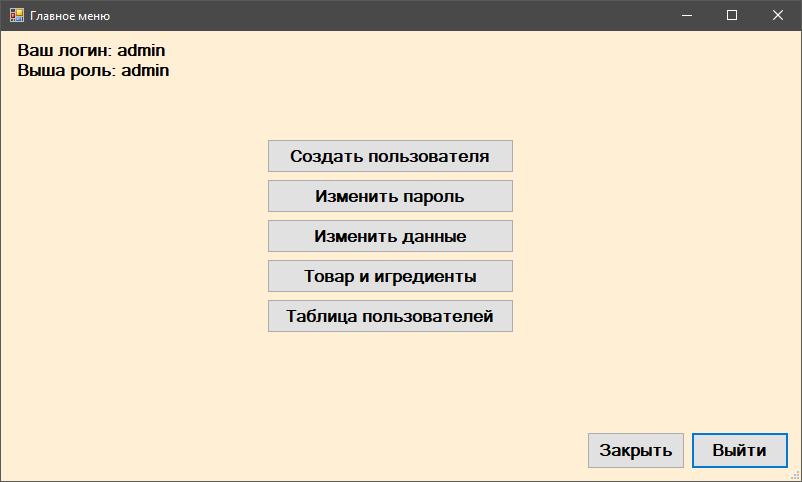
2.2 Этапы разработки API

Было разработано API для аутентификации пользователя и взаимодействия с базой. API было написано на языке С# .

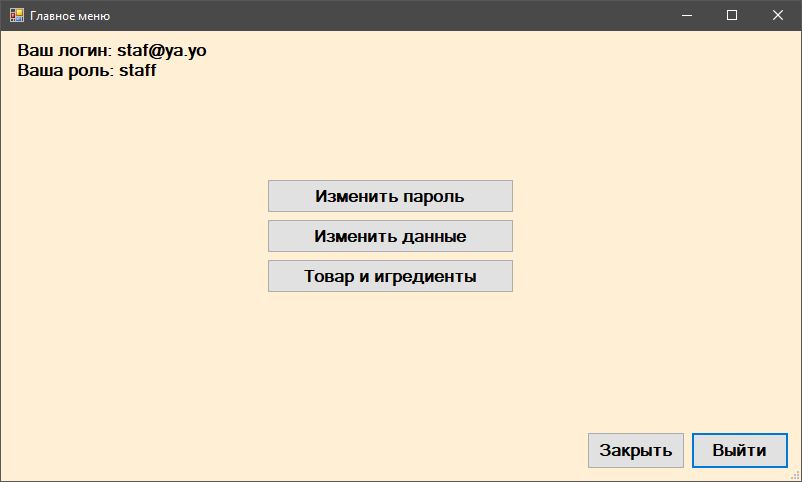
2.3 Скриншоты



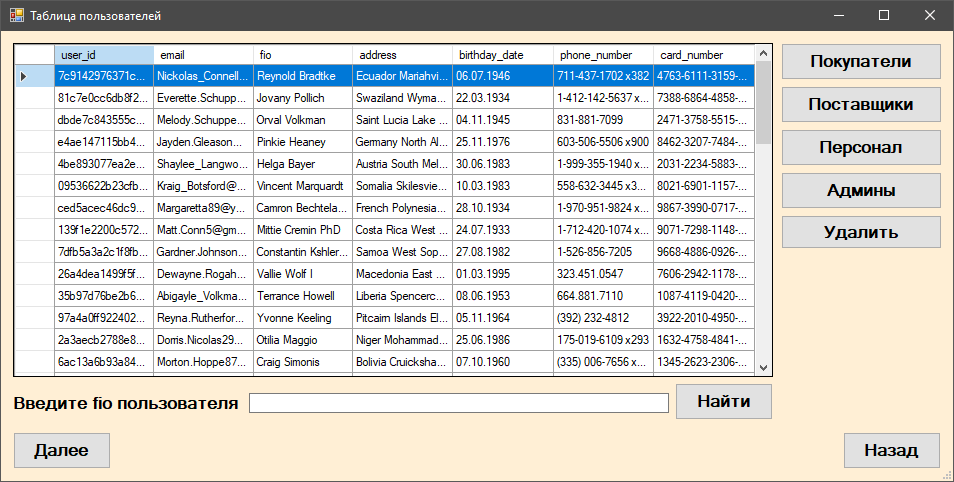
*Рис 14. Авторизаиця*



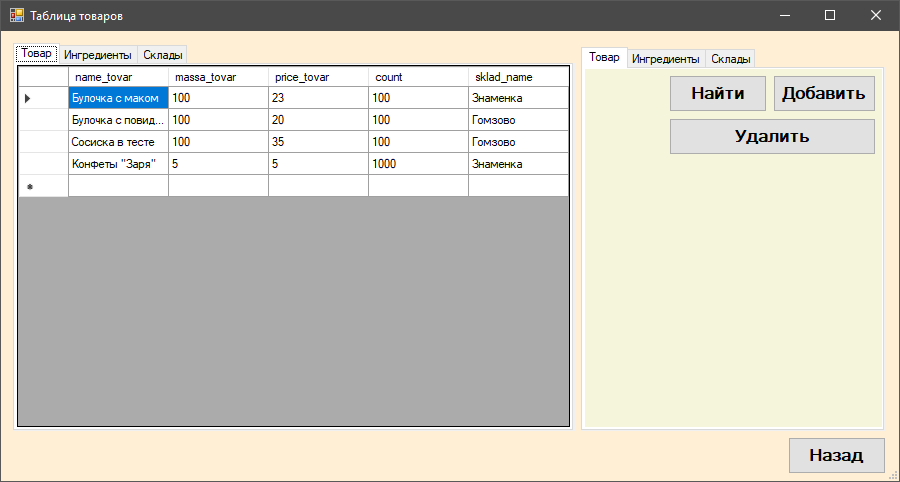
*Рис 15. Главное меню(администратор)*



*Рис 16. Главное меню(юзер)*



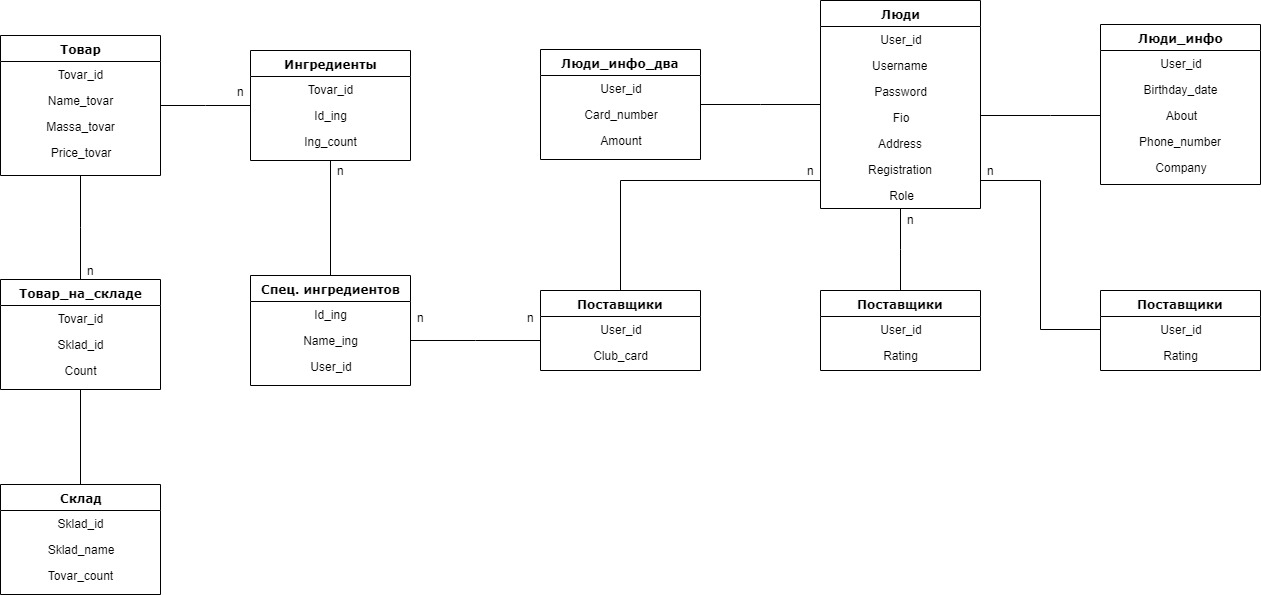
*Рис 17. Таблица пользователей доступная админу*



*Рис 18. Взаимодействие с товаром*

**Приложения**

1. ER-диаграмма



1. Исходные коды и документы:

<https://github.com/kiryosha/DS-course-work>

**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 кондитерской фирмы.