МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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 агентства недвижимости**

Выполнили: студенты группы БИ-31

Тарасов И.С., Вохминцев М.С.,

Белков А.И.

Проверил: доцент кафедры

ИБ Сучков Д.С.

Йошкар-Ола

2020 г.

**СОДЕРЖАНИЕ**

**Введение3**

**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 данных5-7

2.2 Этапы разработки API8-18

**3. Приложения19**

3.1 ER-диаграмма19

3.2 Ссылка на github.com19

**4. Вывод20**

**Введение**

В курсовой работе рассматривается создание базы данных, предназначенной для автоматизации работы книжного магазина. База данных позволяет продавцу заключать сделки, создавать покупателей, узнавать об ассортименте книг. Также реализована автоматизация продажи книги, где можно выбрать книгу, узнать ее стоимость, дату покупки и данные о продавце.

## **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, применяя mySQL Workbench или Dbearer.
* 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***

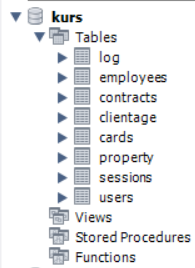
* Наличие не менее 7 таблиц, в том числе таблицы сессий и пользователей.
* Структура таблицы должна содержать не менее 3-х полей, одно из которых ключевое.
* Правомерное использование типов данных.
* Обязательно использование триггеров и/или хранимых процедур.
* Форма нормализации не менее 3NF.
* Индексирование 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 каждой таблице есть ключевое поле. Владельцем всех таблиц является root.



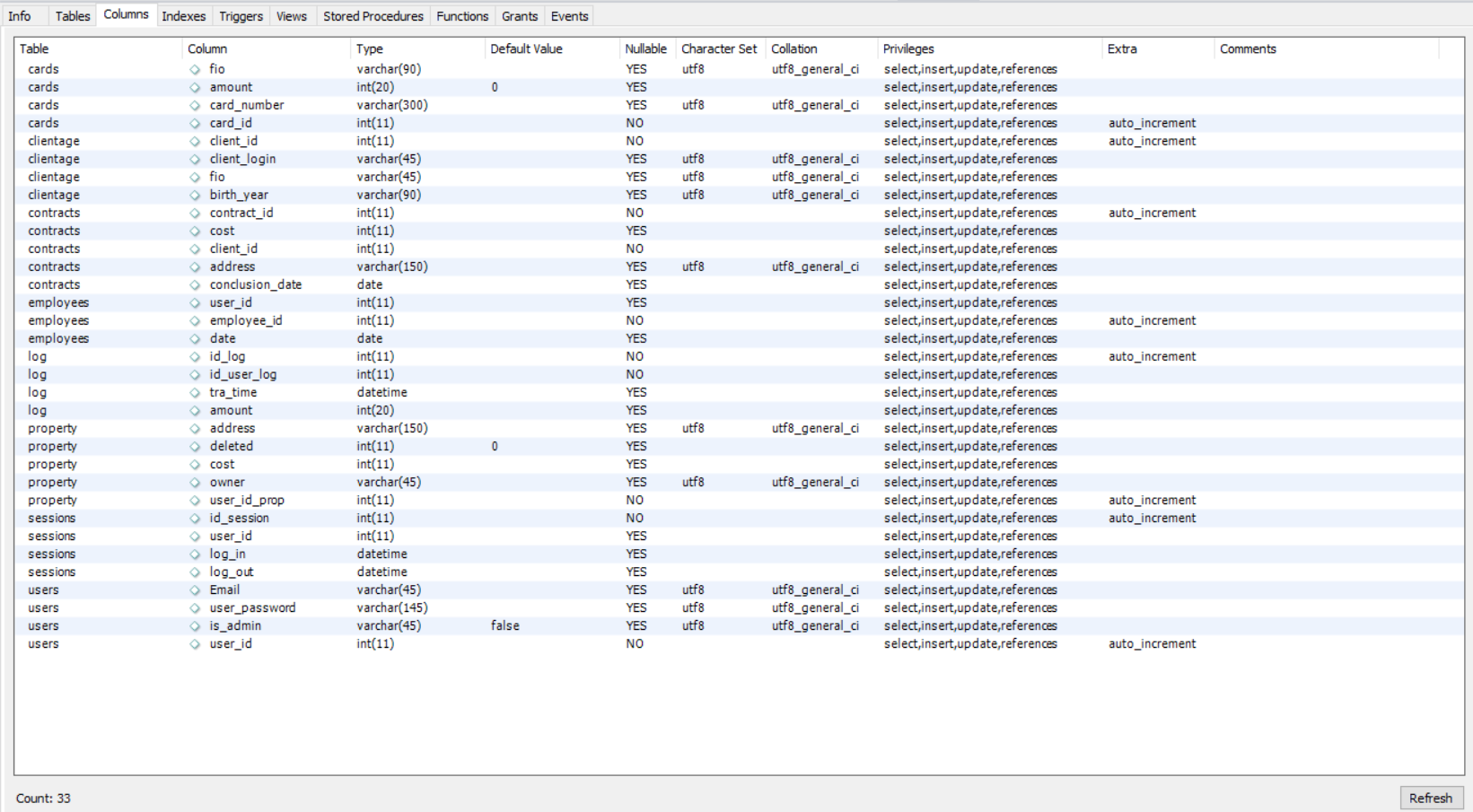
В том числе созданы таблицы сессий (*sessions)* и пользователей (*users*), также таблица данных клиентов (clientage). Также есть таблица, содержащие информацию о кредитной карте - *cards.* И данных о недвижимости - *property.*

Информацию о сделках можно посмотреть в таблице contracts.

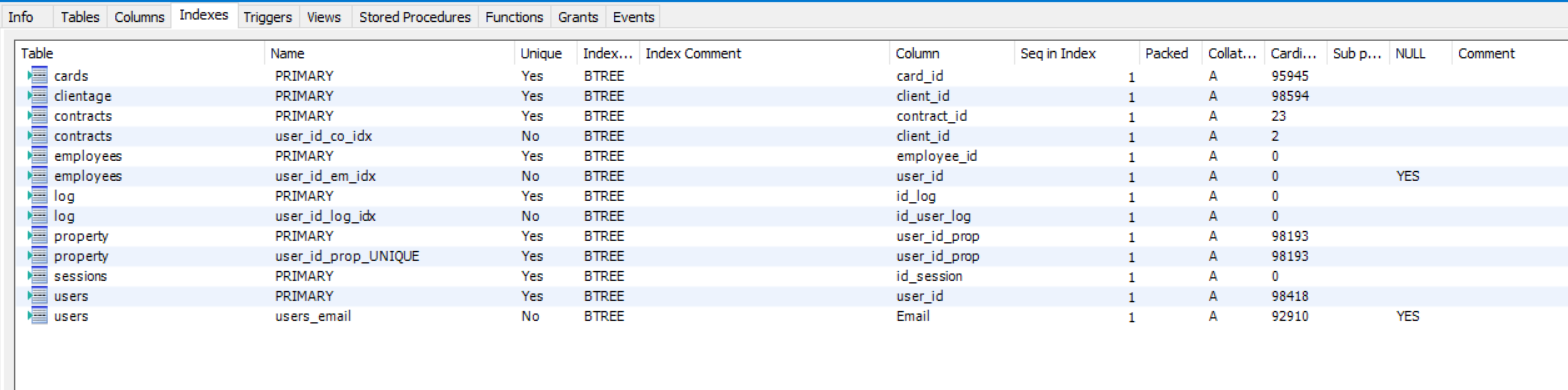
Отдельно можно посмотреть информацию о сотрудниках в таблице employees*.*

В таблице *log хранятся логи движения средств на счету*.

Структура реализованных таблиц:



Индексы:



Используемый триггер и функция для него:

*  триггер, отвечающий за шифрование номеров карт:

DELIMITER $$

CREATE TRIGGER crypt\_after\_insert before insert ON cards

FOR EACH ROW BEGIN

Set new.card\_number = (hex(AES\_ENCRYPT( new.card\_number, 'aN$%$3?Anv?}LBOQkW7@' ))) ;

END$$ DELIMITER ;

*  триггер для аудита движения средств на счету

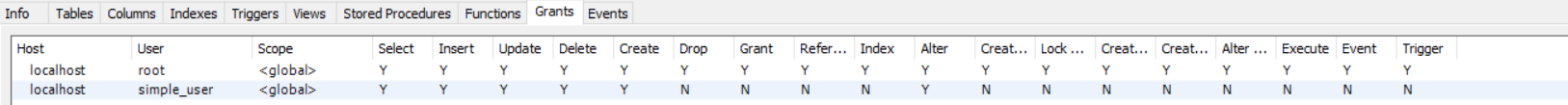
DELIMITER $$

CREATE TRIGGER log\_after\_update AFTER UPDATE ON cards

FOR EACH ROW BEGIN

INSERT INTO log (id\_user\_log,tra\_time,amount) VALUES (NEW.card\_id,NOW(),NEW.amount);   
END$$ DELIMITER ;

Проведена настройка пользователей базы данных для разграничения прав доступа и прав на редактирование структуры базы данных:

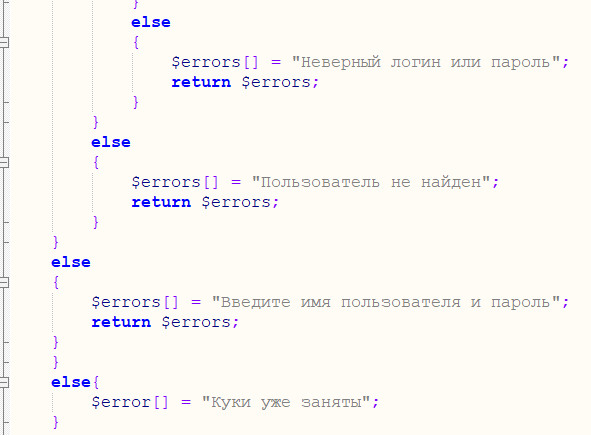


2.2 Этапы разработки API

Было разработано API для аутентификации и регистрации пользователей, просмотра и покупки недвижимости, просмотр заключенных контрактов, принятие на работу и увольнение сотрудников, а также просмотр и удаление пользователей написанное на языке Php.

Аутентификация:



Регистрация:



**Функции вывода из ключевых таблиц:**

Вывод пользователей:

****

Вывод данных по сделке:



Вывод списка недвижимости, здесь осуществлена возможность пагинации:

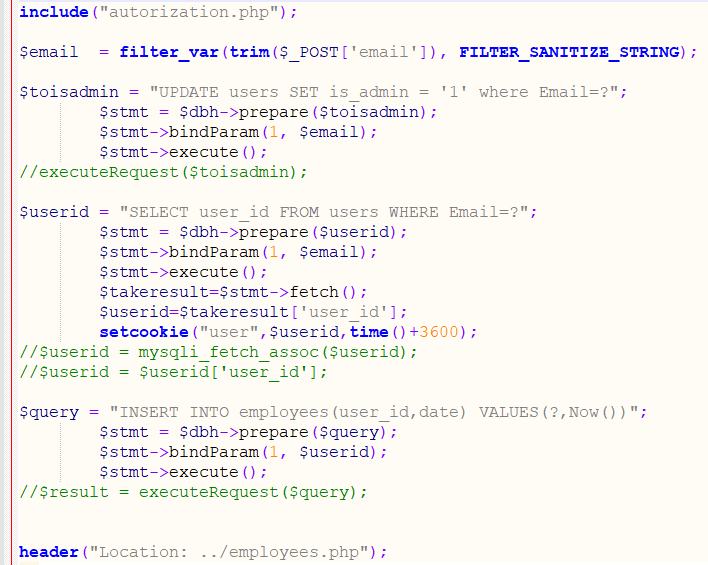


**Добавление данных в БД:**

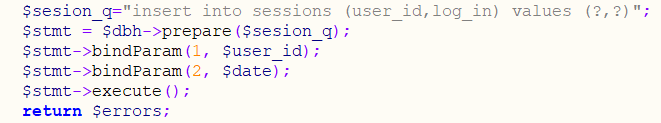
Покупка недвижимости:



Добавление нового сотрудника:

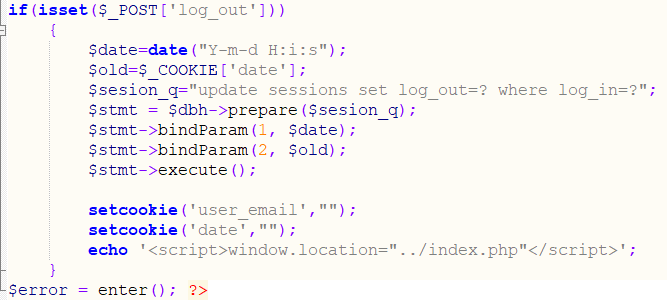


Добавление сессии происходит при авторизации:

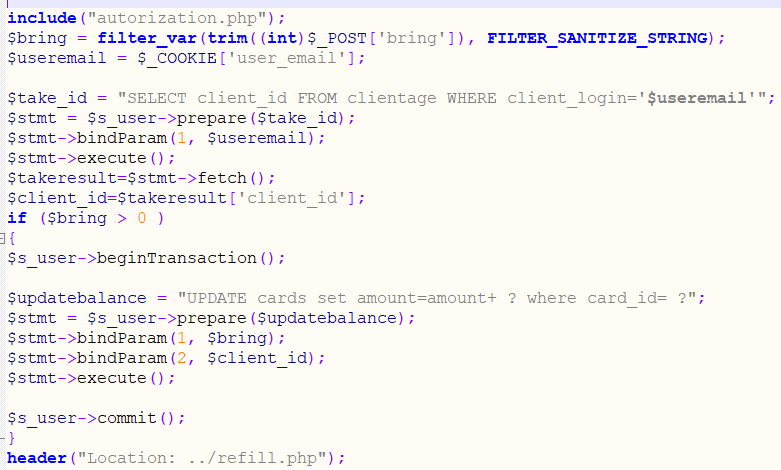


**Изменение данных в БД:**

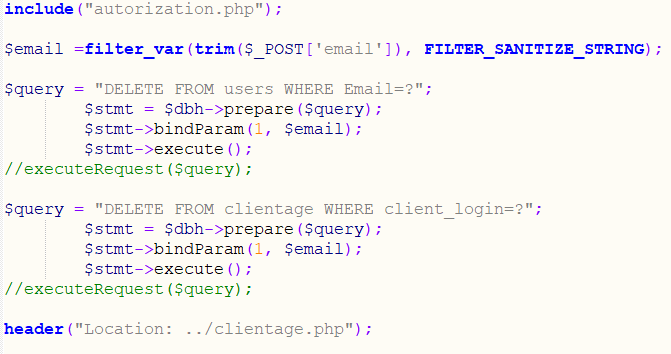
Обновление данных сессии:



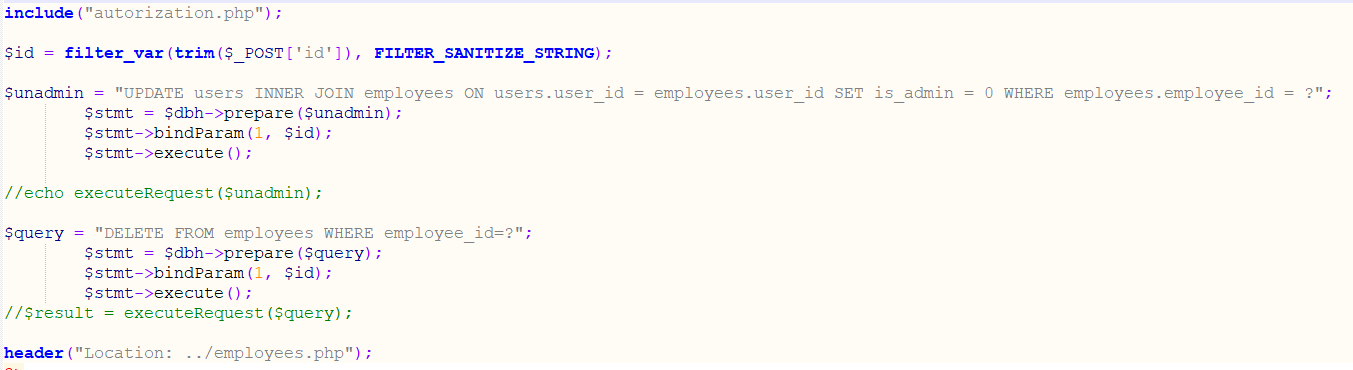
Пополнение баланса:



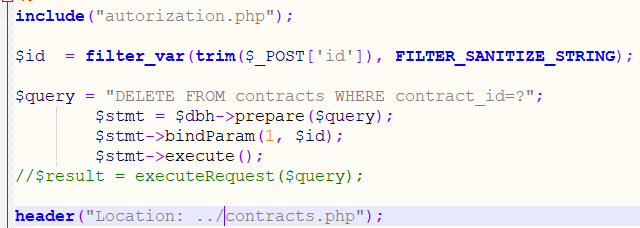
Удаление пользователя:



Удаление сотрудника:



Удаление контракта:



При помощи алгоритма AES происходит шифрация номеров карт. Advanced Encryption Standard (AES) - один из наиболее часто используемых и наиболее безопасных алгоритмов шифрования, доступных сегодня.  ЭТО [симметричный алгоритм](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B5_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) [блочного шифрования](https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80) (размер блока 128 бит, ключ 128/192/256 бит). История его успеха началась в 1997 году, когда NIST (Национальный институт стандартов и технологий) официально начал искать замену стареющему стандарту DES. Алгоритм под названием «Rijndael», разработанный бельгийскими криптографами Daemen и Rijmen, отличался повышенной безопасностью, а также производительностью и гибкостью.

Хэширование паролей происходит с помощью bcrypt. Это адаптивная криптографическая [хеш-функция](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) [формирования ключа](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA%D0%BB%D1%8E%D1%87%D0%B0), используемая для защищенного хранения [паролей](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%BE%D0%BB%D1%8C). Разработчики: [Нильс Провос](https://en.wikipedia.org/wiki/Niels_Provos) и David Mazières. Функция основана на шифре [Blowfish](https://ru.wikipedia.org/wiki/Blowfish" \o "Blowfish), впервые представлена на [USENIX](https://ru.wikipedia.org/wiki/USENIX) в 1999 году. Для защиты от атак с помощью [радужных таблиц](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D1%83%D0%B6%D0%BD%D0%B0%D1%8F_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0) bcrypt использует соль (salt); кроме того, функция является адаптивной, время её работы легко настраивается и её можно замедлить, чтобы усложнить атаку перебором.

Алгоритм bcrypt использует алгоритм настройки ключей из «Eksblowfish»:

EksBlowfishSetup(*cost*, *salt*, *key*)

*state*

{\displaystyle \gets }

InitState()

*state*

{\displaystyle \gets }

ExpandKey(*state*, *salt*, *key*)

**repeat** (2*cost*)

*state*

{\displaystyle \gets }

ExpandKey(state, 0, key)

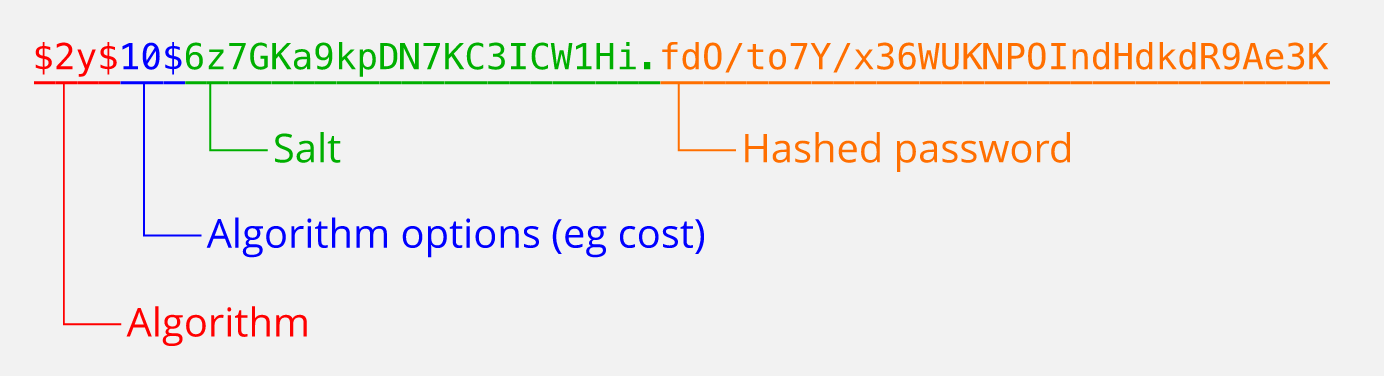
*state*

{\displaystyle \gets }

ExpandKey(state, 0, salt)

**return** *state*

После хэширования пароли хранятся в виде:



Ниже представлены скриншоты страниц от лица пользователя:

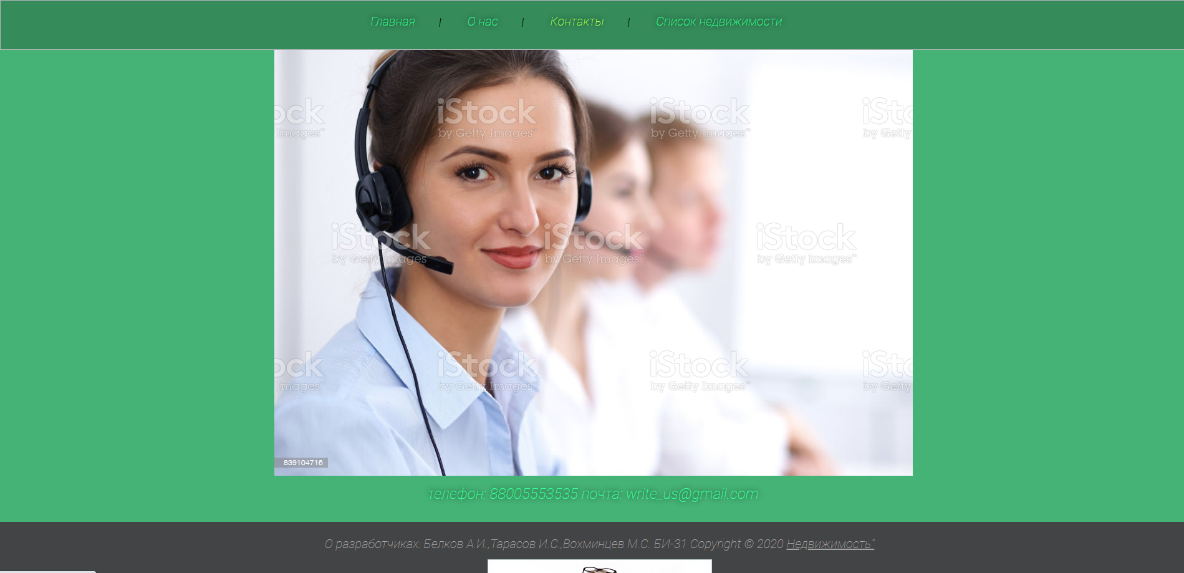
Главная страница:



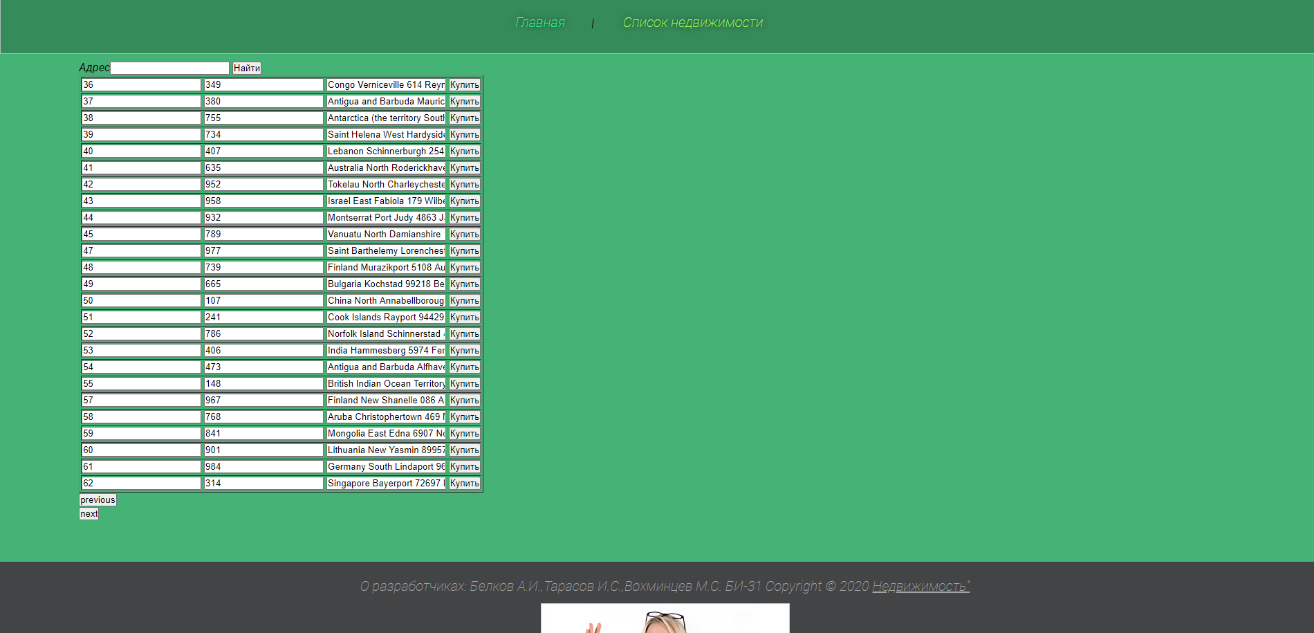
О нас:



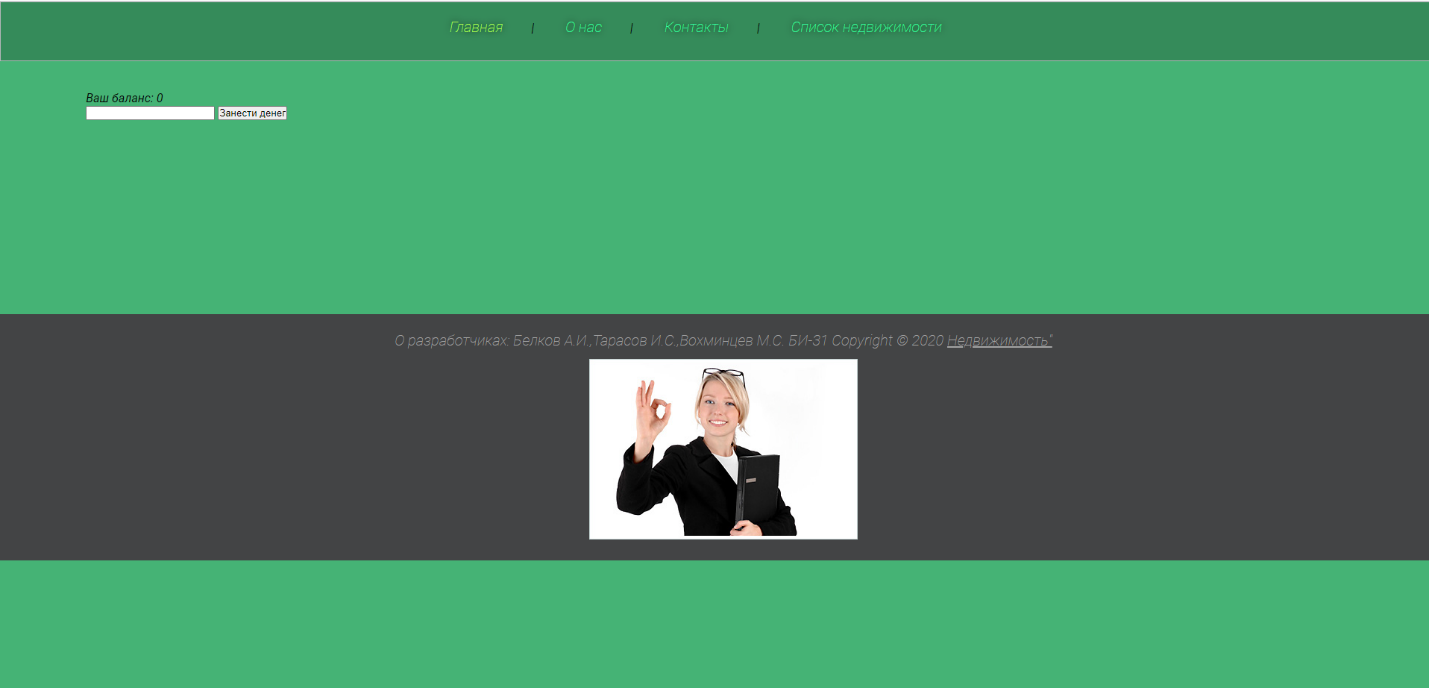
Контакты:



Просмотр недвижимости(реализована пагинация):



Пополнение баланса:



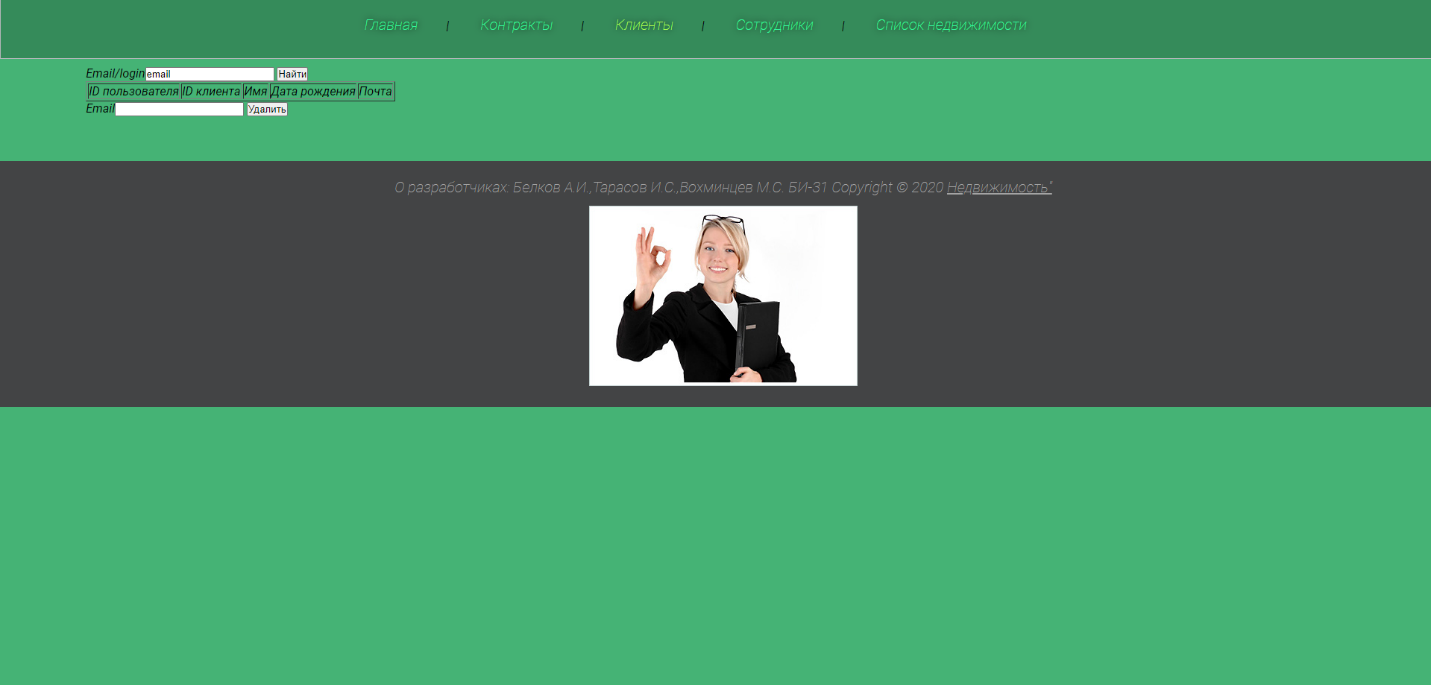
Ниже представлены скриншоты страниц от лица админа или сотрудника:



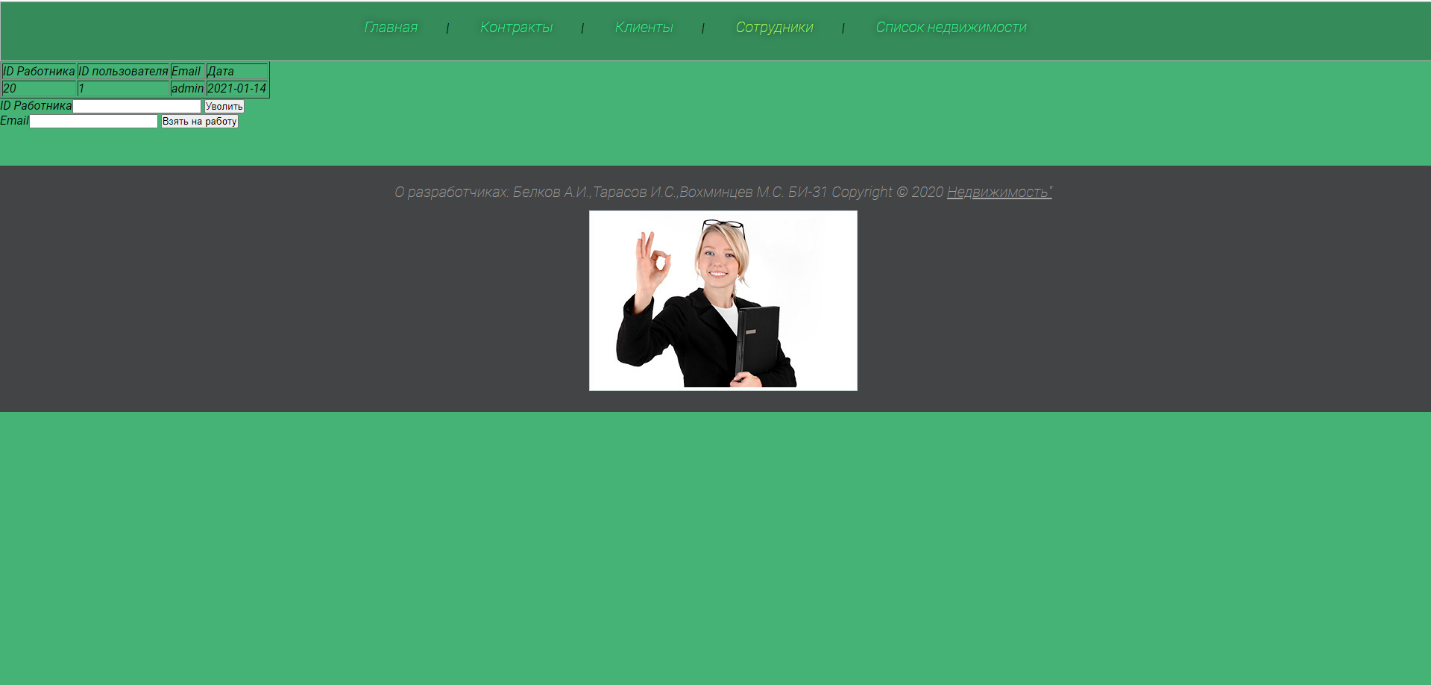
Просмотр контрактов агенства:



Клиенты:

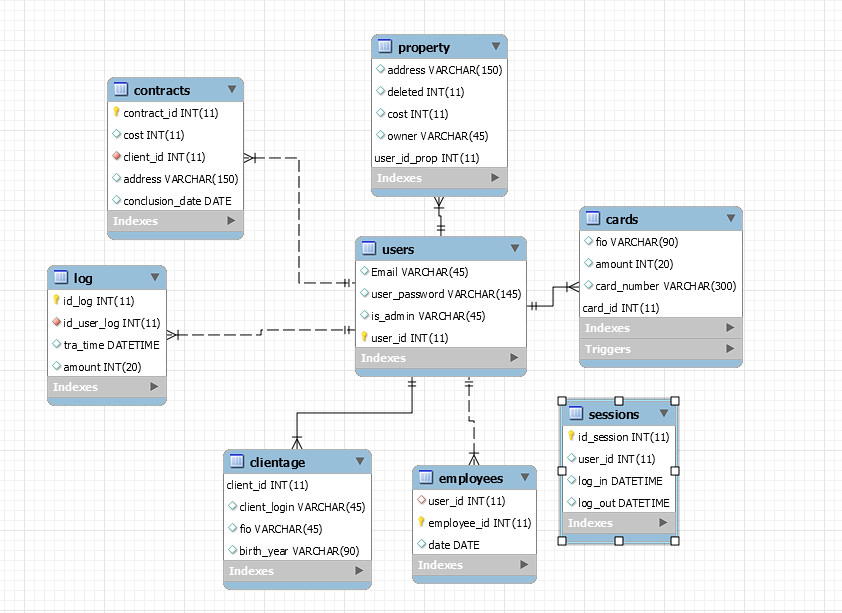


Сотрудники:



**Приложения**

1. ER-диаграмма



1. Исходные коды и документы:

Ссылка на гитхаб: https://github.com/rodcowartt/KURSA4

**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 агентства недвижимости.