НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Дисциплина: «Базы данных»

Домашнее задание на тему:

«Проект – база данных»

(Риэлторская система)

Пояснительная записка

Выполнил: Осипов Лев,

студент группы БПИ121 (1).

Преподаватель:

Брейман Александр Давидович,

к. т. н., доцент.

Москва, 2015 г.

СОДЕРЖАНИЕ

[Предназначение системы 3](#_Toc421789746)

[Функциональные требования 4](#_Toc421789747)

[Ограничения на данные 5](#_Toc421789748)

[Схема 6](#_Toc421789749)

[Корректировка 7](#_Toc421789750)

[SQL DDL 8](#_Toc421789751)

[Примеры запросов на SQL DML 11](#_Toc421789752)

[Группировка запросов в транзакции (примеры) 12](#_Toc421789753)

[Реализация 13](#_Toc421789754)

[Прочие требования 14](#_Toc421789755)

[Выводы 15](#_Toc421789756)

[Список литературы 16](#_Toc421789757)

# ПРЕДНАЗНАЧЕНИЕ СИСТЕМЫ

Система предназначена для облегчения информационного обмена между клиентами и риэлторами.

Система позволяет риэлторам управлять своим профилем с контактными данными и выставлять свои объекты недвижимости.

Клиенты могут управлять своим профилем, просматривать объекты недвижимости, осуществлять поиск по их параметрам, а также просматривать связанных с ними риэлторов и добавлять их (объекты) в закладки.

# ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Система должна обеспечивать выполнение следующих функций:

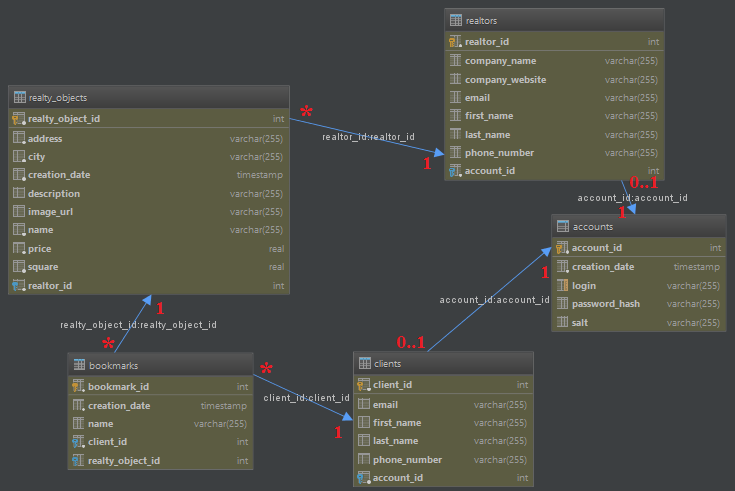
1. регистрация в системе (с уникальным логином) как за риэлтора, так и за клиента;
2. авторизация в системе (при условии ввода корректных данных);
3. изменение личного профиля (у клиента – имя, фамилия, адрес электронной почты, телефон, у риэлтора – имя, фамилия, адрес электронной почты, телефон, имя компании, сайт компании);
4. для риэлтора – управление объектами недвижимости (создание, изменение параметров (имя, город, адрес, площадь, цена, описание, фотография), просмотр, удаление);
5. для клиента – поиск объектов недвижимости по городу, площади, цене; у объекта недвижимости – просмотр информации о связанном с ним риэлторе и добавление в закладки; управление закладками (изменение имени, просмотр, удаление);
6. выход из аккаунта;
7. удаление аккаунта.

# ОГРАНИЧЕНИЯ НА ДАННЫЕ

Ввиду функциональных требований ставятся следующие ограничения на данные:

1. каждому клиенту ставится в соответствие один аккаунт;
2. каждому риэлтору ставится в соответствие один аккаунт;
3. аккаунт может принадлежать либо риэлтору, либо клиенту;
4. каждому объекту недвижимости ставится в соответствие один риэлтор;
5. у риэлтора может быть много объектов недвижимости;
6. каждой закладке ставится в соответствие один клиент и один объект недвижимости;
7. у клиента может быть много закладок;
8. на объект недвижимости может указывать много закладок.

# СХЕМА



# КОРРЕКТИРОВКА

После проектирования выяснилось, что с условием для вторичного ключа таблицы закладок на таблицу объектов недвижимости NOT NULL необходимо удалять закладку при удалении объекта. Это поведение аномально для клиента, поэтому было решено сделать возможным значение NULL этого атрибута. Таким образом, после удаления объекта недвижимости, на который у клиентов созданы закладки, у клиентов будет отображаться, что информации по этому объекту недвижимости нет.

# SQL DDL

Для создания базы применим следующий скрипт (DDL SQL):

CREATE TABLE accounts

(

account\_id integer NOT NULL DEFAULT nextval('accounts\_account\_id\_seq'::regclass),

creation\_date timestamp without time zone NOT NULL,

login character varying(255),

password\_hash character varying(255),

salt character varying(255),

CONSTRAINT accounts\_pkey PRIMARY KEY (),

CONSTRAINT uk\_cc2c9baeppipgy2rjeccwcqs0 UNIQUE ()

)

WITH (

OIDS=FALSE

);

ALTER TABLE accounts

OWNER TO postgres;

CREATE SEQUENCE accounts\_account\_id\_seq

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 6

CACHE 1;

ALTER TABLE accounts\_account\_id\_seq

OWNER TO postgres;

CREATE TABLE bookmarks

(

bookmark\_id serial NOT NULL,

creation\_date timestamp without time zone NOT NULL,

name character varying(255),

client\_id integer NOT NULL,

realty\_object\_id integer,

CONSTRAINT bookmarks\_pkey PRIMARY KEY (bookmark\_id),

CONSTRAINT fk\_8yn686cf2dwdckxckwpcqr9a4 FOREIGN KEY (client\_id)

REFERENCES clients (client\_id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT fk\_j057w9nryfxbo4gg26kry2lqp FOREIGN KEY (realty\_object\_id)

REFERENCES realty\_objects (realty\_object\_id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

ALTER TABLE bookmarks

OWNER TO postgres;

CREATE SEQUENCE bookmarks\_bookmark\_id\_seq

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 15

CACHE 1;

ALTER TABLE bookmarks\_bookmark\_id\_seq

OWNER TO postgres;

CREATE TABLE clients

(

client\_id serial NOT NULL,

email character varying(255),

first\_name character varying(255),

last\_name character varying(255),

phone\_number character varying(255),

account\_id integer NOT NULL,

CONSTRAINT clients\_pkey PRIMARY KEY (client\_id),

CONSTRAINT fk\_mowyd0cvw6erkqi5vreoc5rwy FOREIGN KEY (account\_id)

REFERENCES accounts (account\_id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

ALTER TABLE clients

OWNER TO postgres;

CREATE SEQUENCE clients\_client\_id\_seq

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 2

CACHE 1;

ALTER TABLE clients\_client\_id\_seq

OWNER TO postgres;

CREATE TABLE realtors

(

realtor\_id serial NOT NULL,

company\_name character varying(255),

company\_website character varying(255),

email character varying(255),

first\_name character varying(255),

last\_name character varying(255),

phone\_number character varying(255),

account\_id integer NOT NULL,

CONSTRAINT realtors\_pkey PRIMARY KEY (realtor\_id),

CONSTRAINT fk\_oxcqaclbgfbxr2j2aqcihguqi FOREIGN KEY (account\_id)

REFERENCES accounts (account\_id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

ALTER TABLE realtors

OWNER TO postgres;

CREATE SEQUENCE realtors\_realtor\_id\_seq

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 4

CACHE 1;

ALTER TABLE realtors\_realtor\_id\_seq

OWNER TO postgres;

CREATE TABLE realty\_objects

(

realty\_object\_id serial NOT NULL,

address character varying(255) NOT NULL,

city character varying(255) NOT NULL,

creation\_date timestamp without time zone NOT NULL,

description character varying(255),

image\_url character varying(255),

name character varying(255) NOT NULL,

price real NOT NULL,

square real NOT NULL,

realtor\_id integer NOT NULL,

CONSTRAINT realty\_objects\_pkey PRIMARY KEY (realty\_object\_id),

CONSTRAINT fk\_9y2gb8rfls4g7ncmbbumcqp6l FOREIGN KEY (realtor\_id)

REFERENCES realtors (realtor\_id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

ALTER TABLE realty\_objects

OWNER TO postgres;

CREATE SEQUENCE realty\_objects\_realty\_object\_id\_seq

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 15

CACHE 1;

ALTER TABLE realty\_objects\_realty\_object\_id\_seq

OWNER TO postgres;

# ПРИМЕРЫ ЗАПРОСОВ НА SQL DML

**Поиск объекта недвижимости:**

select \* from realty\_objects inner join realtors on realty\_objects.realtor\_id=realtors.realtor\_id where city='Moscow' and square>=100 and square<=200 and price>=20000000 and price<=100000000

**Изменение профиля риэлтора:**

update realtors set company\_name='Httyy', company\_website='httyyv.com', email='ishorov@gmail.com', first\_name='Ivan', last\_name='Shorov', phone\_number='89167835623' where realtor\_id=3

**Удаление закладки:**

delete from bookmarks where bookmark\_id=10

# ГРУППИРОВКА ЗАПРОСОВ В ТРАНЗАКЦИИ (ПРИМЕРЫ)

**Создание профиля клиента:**

insert into accounts (creation\_date, login, password\_hash, salt) values (‘2015-06-11 04:29:34.485000’, ‘login1’, ‘902fa5c570b068c7e7e4e806cfa25c70f1f9819416a3de5addb69520ebb13547’, ‘omwhofn’)

insert into clients (account\_id, email, first\_name, last\_name, phone\_number) values (1, null, ‘Vladimir’, null, null)

**Удаление профиля клиента:**

delete from bookmarks where bookmark\_id=11

delete from clients where client\_id=3

delete from accounts where account\_id=5

**Удаление профиля риэлтора:**

update bookmarks set realty\_object\_id=null where bookmark\_id=12

delete from realty\_objects where realty\_object\_id=6

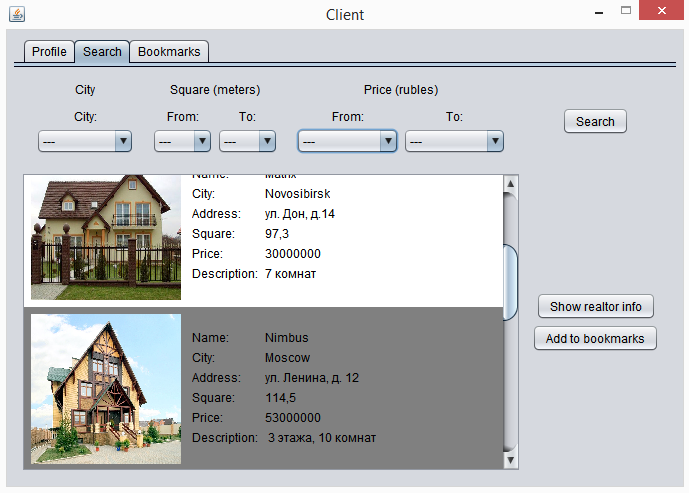
delete from realtors where realtor\_id=7

delete from accounts where account\_id=9

# РЕАЛИЗАЦИЯ

Для реализации была использована технология Hibernate и СУБД PostgreSQL 9.4. Для графического интерфейса были использованы библиотеки Swing.

Было реализовано 2 Java-приложения, для клиента и риэлтора соответственно. Приложения в полной мере выполняют поставленные функциональные требования. Пример скриншота приложения клиента (окно поиска):



# ПРОЧИЕ ТРЕБОВАНИЯ

Из реализации вытекают следующие нефункциональные требования:

1. установленная Java 8 или выше;
2. установленная PostgreSQL 9.4 или выше, существование базы согласно скрипту DDL, описаному в соответствующем разделе;
3. настроенная директория для хранения изображений.

# ВЫВОДЫ

В результате работы получилась работоспособная система, отвечающая поставленным функциональном требованиям.

Был получен опыт в использовании Hibernate и различных видах взаимодействия с базами данных.

Также был получено дополнительное понимание структуры СУБД и таблиц в ней.

# СПИСОК ЛИТЕРАТУРЫ

1. Материалы по курсу «Базы данных»;
2. Hibernate DAO – <http://examples.javacodegeeks.com/enterprise-java/hibernate/hibernate-jpa-dao-example/>;
3. PostgreSQL tutorial – <http://www.tutorialspoint.com/postgresql/>