**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №3**

**«Язык SQL-DDL»**

**Информационное обеспечение систем управления**

Студент гр. 43501/1 Красильников В.В.

Преподаватель Мяснов А.В.

Санкт-Петербург

2015

# 1.Цель работы

Ознакомится  с основами проектирования схемы БД, языком описания сущностей и ограничений БД SQL-DDL.

## Индивидуальное задание:

Учет квартир/комнат с характеристиками: кол-во комнат, площадь общая, жилая, каждой комнаты, санузел совмещенный/раздельный, этаж, удаленность от метро, лифт, мусоропровод, текущая стоимость и т.д. Учет продавцов/покупателей. Выборки по характеристикам. Выборки по клиентам.

## Программа работы:

1. Самостоятельное изучение SQL-DDL

2. Создание скрипта БД в соответствии с согласованной схемой (должны присутствовать первичные и внешние ключи, ограничения на диапазоны значений).

3. Создать скрипт, заполняющий все таблицы БД данными

4. Получить ER-диаграмму созданной БД с помощью Database Designer.

5. Автоматически сгенерировать данные при помощи IBExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

# 2. Создание БД

Скрипт для создания БД:

create database 'localhost:D:/db/FLAT3.fdb' user 'SYSDBA' password 'masterkey'

default character set win1251;

CREATE TABLE PEOPLES

(

ID\_PPL int NOT NULL primary key,

SECONDNAME varchar(12),

FIRSTNAME varchar(12),

PATRONYMIC varchar(12),

PHONE varchar (11) unique,

PASSPORT\_NUMBER varchar (11) unique

);

CREATE TABLE DISTRICTS

(

ID\_DISTRICT int NOT NULL primary key,

DISTRICTNAME varchar(15) unique

);

CREATE TABLE TYPE\_H

(

ID\_TYPE int NOT NULL primary key,

TYPENAME varchar(15) unique

);

CREATE TABLE METROSTATE

(

ID\_STATE int NOT NULL primary key,

STATENAME varchar(25) unique

);

CREATE TABLE FLATS

(

ID\_KV int NOT NULL primary key,

Rooms int,

S DECIMAL(5,2),

S\_Y varchar(3),

etaj int,

ydal int,

lift varchar(3),

m\_p varchar (3),

cash int,

DISTRICT int, FOREIGN KEY (DISTRICT) REFERENCES DISTRICTS(ID\_DISTRICT),

METRO int, FOREIGN KEY (METRO) REFERENCES METROSTATE(ID\_STATE),

ID\_OWNER int, FOREIGN KEY(ID\_OWNER) REFERENCES PEOPLES(ID\_PPL),

TYPE int, FOREIGN KEY(TYPE) REFERENCES TYPE\_H (ID\_TYPE)

);

CREATE TABLE ROOMS

(

ID\_ROOM int NOT NULL primary key,

ID\_FLAT int, FOREIGN KEY (ID\_FLAT) REFERENCES FLATS(ID\_KV),

S int

);

CREATE TABLE DEAL

(

ID\_DEAL int NOT NULL primary key,

ID\_SALER int, FOREIGN KEY (ID\_SALER) REFERENCES PEOPLES (ID\_PPL),

ID\_POKUP int, FOREIGN KEY (ID\_POKUP) REFERENCES PEOPLES (ID\_PPL),

DEALCASH int

);

Скрипт для заполнения таблиц данными:

insert into PEOPLES values (1,'Ivanov','Anton','Olegovich',89114564435,4564543);

insert into PEOPLES values (2,'Petrov','Oleg','Ivanovich',89114532435,4543543);

insert into PEOPLES values (3,'Sidorov','Petr','Antonovich',89114364435,3564543);

insert into DISTRICTS values (1,'Petrodvorcovij');

insert into DISTRICTS values (2,'Admiralteiskij');

insert into DISTRICTS values (3,'Viborgskij');

insert into TYPE\_H values (1,'blok');

insert into TYPE\_H values (2,'kirpich');

insert into METROSTATE values (1,'Pl.Mujestva');

insert into METROSTATE values (2,'Lesnaya');

insert into METROSTATE values (3,'Politehnicheskaya');

insert into FLATS values (1,4,70,'yes',4,500,'yes','yes',200000,1,1,1,2);

insert into FLATS values (2,3,50,'yes',6,1500,'yes','yes',150000,1,2,2,2);

insert into FLATS values (3,2,40,'yes',14,2500,'yes','no',100000,2,2,3,2);

insert into FLATS values (4,1,30,'no',3,600,'yes','yes',80000,1,3,1,1);

insert into FLATS values (5,4,75,'yes',1,700,'no','yes',260000,1,3,2,2);

insert into ROOMS values (1,1,15);

insert into ROOMS values (2,2,12);

insert into ROOMS values (3,3,13);

insert into ROOMS values (4,1,13);

insert into ROOMS values (5,2,14);

insert into ROOMS values (6,3,17);

insert into ROOMS values (7,4,18);

insert into DEAL values (1,1,2,100000);

insert into DEAL values (2,2,3,160000);

insert into DEAL values (3,3,1,180000);

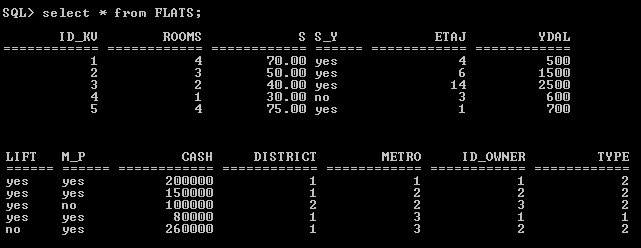


Рис 1. Запрос select

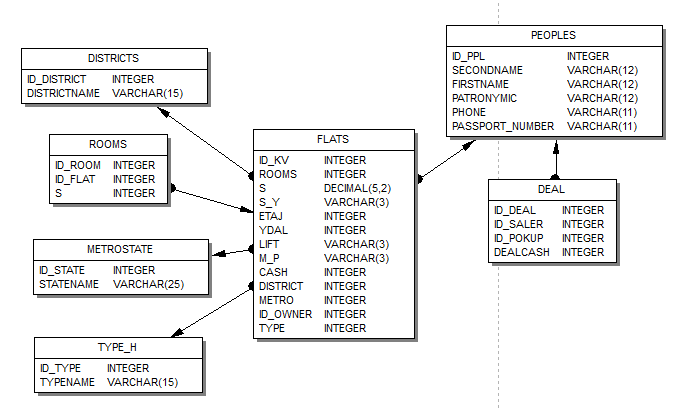


Рис 2. ER-диаграмма, построенная в IBExpert.

## Заполнение таблиц тестовыми данными

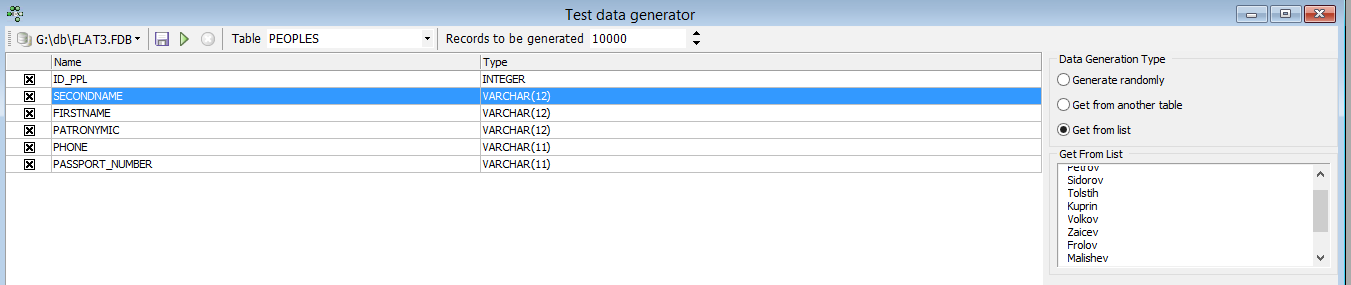


Рис 3. Окно IBExpert. Test data generator.

Для заполнения 3 таблиц тестовыми данными использовался IBExpert.

Результат генерации:

SQL> select \* from PEOPLES;

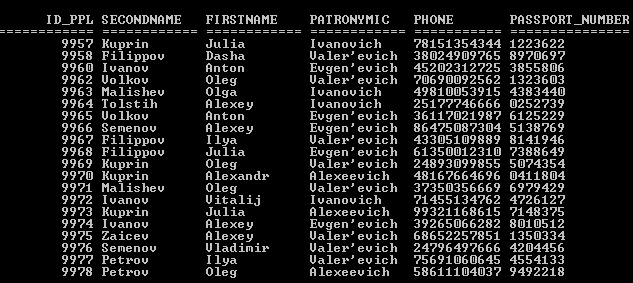


Рис 4. Запрос select

### SQL> select \* from PEOPLES where ID\_PPL>=6000 and ID\_PPL<=6013;

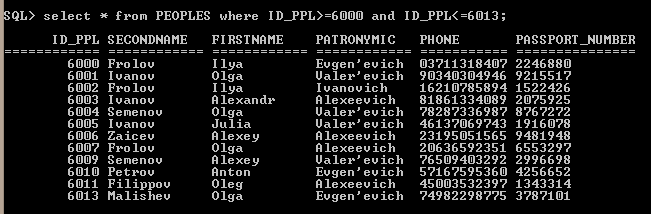


Рис 5. Запрос select

Ввести учет агентств недвижимости:

CREATE TABLE AGENCY

(

ID\_AGENCY int NOT NULL primary key,

AGENCYNAME varchar(30) unique,

PHONE\_NUMBER varchar(20) unique,

DISTRICT int, FOREIGN KEY (DISTRICT) REFERENCES DISTRICTS(ID\_DISTRICT),

METRO int, FOREIGN KEY (METRO) REFERENCES METROSTATE(ID\_STATE),

PERSENT DECIMAL(4,2)

);

Ввести учет агентов, которые получают комиссию по сделке:

CREATE TABLE AGENT

(

ID\_AGENT int NOT NULL primary key,

IDPPL\_AGENT int, FOREIGN KEY(IDPPL\_AGENT) REFERENCES PEOPLES(ID\_PPL),

DEALSUM DECIMAL(9,2),

AGENTAGENCY int, FOREIGN KEY (AGENTAGENCY) REFERENCES AGENCY(ID\_AGENCY)

);

# 3. Вывод

В ходе работы были изучены основы языка SQL-DDL – языка определения структур и ограничений целостности баз данных. Были изучены команды создания и удаления баз данных; создания, изменения и удаления таблиц. Для создания большого количества тестовых данных использовалась программа IBExpert, предоставляющая графический интерфейс работы с БД.

Была создана база данных в соответствии с заданием. Графический интерфейс безусловно упрощает работу с базой данных. SQL-DDL позволяет за счёт довольно простых в формулировке команд определить структуру целой базы данных, задавать обязательность уникальности данных, а также первичные ключи и вводить ограничения на данные.