**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 Матлаш И.Г.

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:

БД хранит информацию об машинах, их характеристиках, состоянии, а также информацию о владельце, дате размещения объявления и действительно ли сейчас оно.

## 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 'C:/db/autosale.fdb' user 'SYSDBA' password 'masterkey';

commit;

connect 'C:/db/autosale.fdb' user 'SYSDBA' password 'masterkey';

create table mark (

id int primary KEY,

name varchar(20) not null unique);

create table body (

id int primary KEY,

name varchar(20) not null unique);

create table engine (

id int primary KEY,

name varchar(20) not null unique);

create table transmission (

id int primary KEY,

name varchar(20) not null unique);

create table condition (

id int primary KEY,

name varchar(20) not null unique);

create table status (

id int primary KEY,

name varchar(20) not null unique);

create table person (

id int primary KEY,

name varchar(20) not null,

tel decimal(11,0) not null unique,

adress varchar(20));

CREATE TABLE model (

id int primary KEY,

name varchar(20) not null unique,

mark int references mark(id) not null);

CREATE TABLE car (

id INT PRIMARY KEY,

photo BLOB,

model INT NOT NULL REFERENCES model(id),

mileage INT check (mileage >= 0),

edition\_year decimal(4,0) check (edition\_year >= 1806),

body INT REFERENCES body(id),

engine INT REFERENCES engine(id),

transmission INT REFERENCES transmission(id),

color VARCHAR(20),

condition INT REFERENCES condition(id),

review VARCHAR(100),

price INT NOT NULL,

person INT REFERENCES person(id),

status INT REFERENCES status(id),

create\_date DATE,

delete\_date DATE);

Скрипт для заполнения таблиц данными:

### insert into mark values (1,'Mercedes-Benz');

### insert into mark values (2,'Porsche');

### insert into mark values (3,'Lada');

### insert into body values (1,'SUV');

### insert into body values (2,'Cabriolet');

### insert into body values (3,'Sedan');

### insert into engine values (1,'gasoline');

### insert into engine values (2,'gas');

### insert into transmission values (1,'Automatic');

### insert into transmission values (2,'Manual');

### insert into condition values (1,'new');

### insert into condition values (2,'good');

### insert into condition values (3,'average');

### insert into condition values (4,'bad');

### insert into condition values (5,'terrible');

### insert into status values (1,'on sale');

### insert into status values (2,'saled');

### insert into person values (1,'Ivan','89211234253','SPB,Nevski 11-32');

### insert into person values (2,'Max','89114326789','SPB,2 line 23-9');

### insert into person values (3,'Vitaliy','89219864365','SPB,Liteiniy 4-2');

### insert into model values (1,'911 Carrera',2);

### insert into model values (2,'GL450',1);

### insert into model values (3,'2104',3);

### insert into car values (1,NULL,1,200000,2002,2,1,1,'Pink',1,'best car ever',14999,1,1,'2014-01-28',NULL);



Рис 1. Запрос select

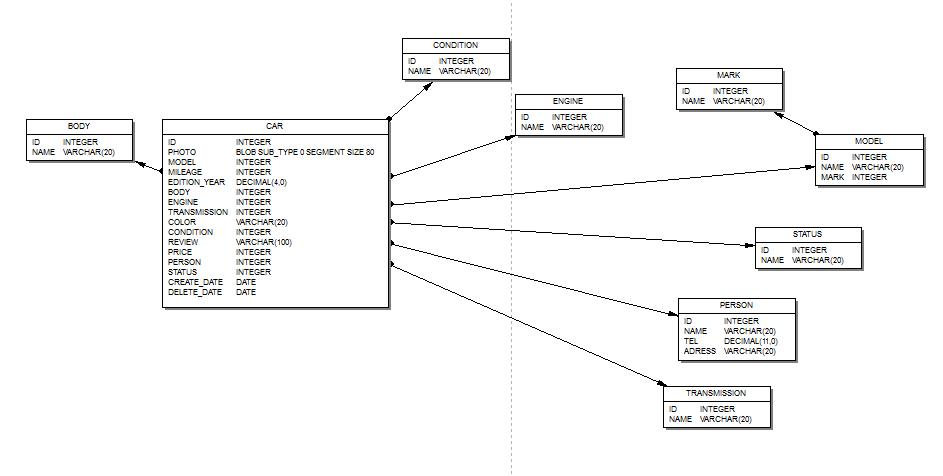


Рис 2. ER-диаграмма, построенная в IBExpert.

## Заполнение таблиц тестовыми данными

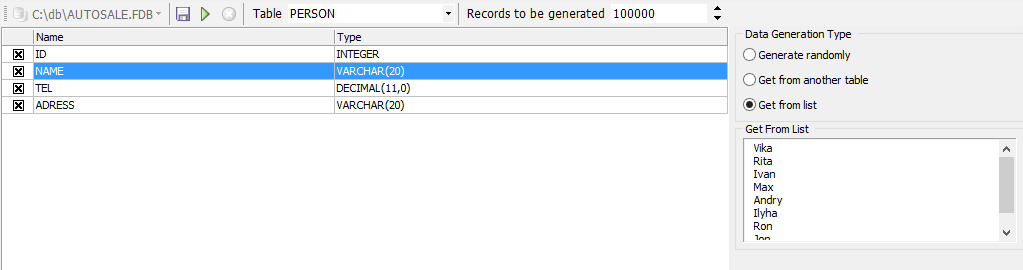


Рис 3. Окно IBExpert. Test data generator.

Для заполнения 3 таблиц тестовыми данными использовался IBExpert. Поскольку он не может генерировать целочисленные данные больше предела 32bit integer, использовался дополнительный запрос типа

### SQL> update person set person.tel = person.tel+89210000000;

Результат генерации:

SQL> select \* from person;

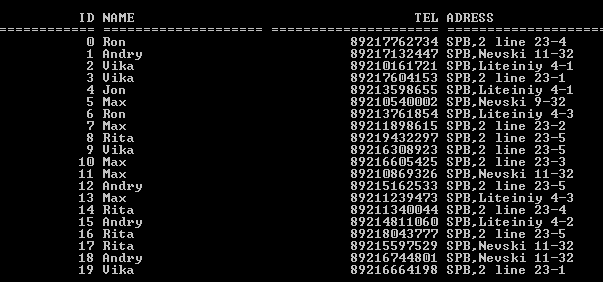


Рис 4. Запрос select

### SQL> select \* from person where person.id>55 and person.id<60;

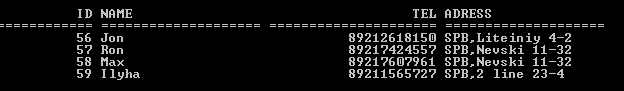


Рис 5. Запрос select

Реализовать учёт платных объявлений:

alter table car add payed\_end date;

alter table car add payed demical(10,2);

alter table person add money demical(10,2); счёт пользователя

# create table money(

# pers int references person(id),

# days date,

# pay demical(10,2),

# car\_id int); на какое объявление были положены средства

Добавить возможность учета статистики просмотров объявлений:

# create table views(

# car int references car(id),

# days date,

# views int);

# 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. Тем не менее при работе c языком SQL-DDL иногда возникали затруднения при отладке команд.