**Санкт- Петербургский политехнический университет Петра Великого**

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе

Курс: «Базы данных»

Тема: «Язык SQL-DDL»

**Выполнил:**

Бояркин Н.С. группа 43501/3

**Проверил:**

Мяснов А.В.

Санкт – Петербург

2016

# 1. Цель работы

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

# 2. 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). Продемонстрировать скрипт преподавателю.
3. Создайте скрипт, заполняющий все таблицы БД данными
4. Выполнение SQL-запросов, изменяющих схему созданной БД по заданию преподавателя. Продемонстрировать их работу преподавателю.
5. Изучите основные возможности IBExpert. Получите ER-диаграмму созданной БД с помощью Database Designer.
6. Автоматически сгенерируйте данные при помощи IBExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

# 3. Ход работы

В соответствии с диаграммой базы данных из предыдущей работы (Рис. 1), был написан SQL-скрипт, создающий базу данных.

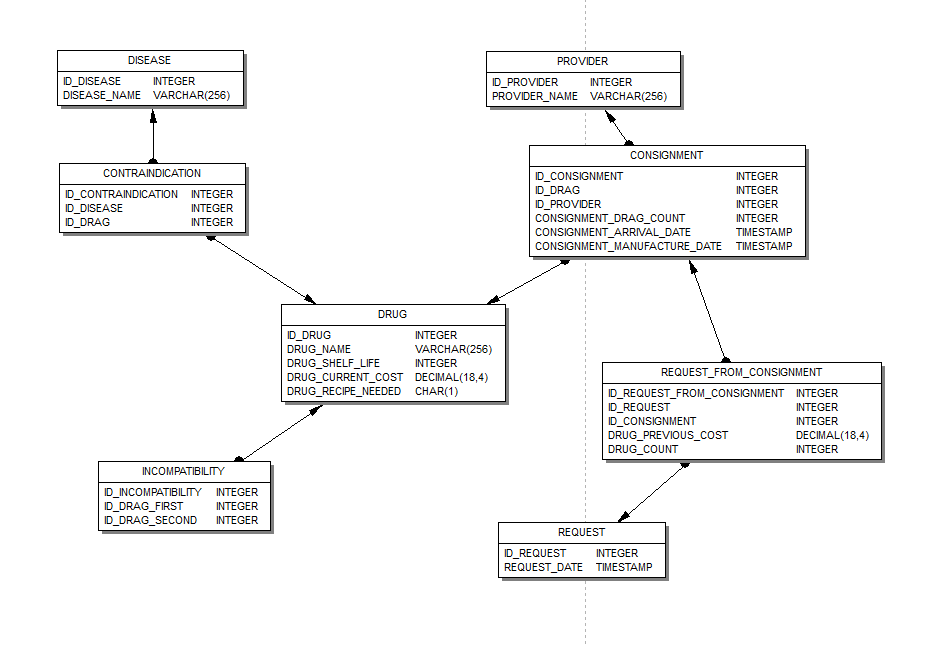


Рис. 1. SQL-диаграмма базы данных до изменения

create.sql:

/\* If database already exists drop it. \*/

CONNECT 'tiger.ftk.spbstu.ru:/var/lib/firebird/435013/pharmacy.fdb' USER 'SYSDBA' PASSWORD 'masterkey';

DROP DATABASE;

/\* Create database and connect. \*/

CREATE DATABASE 'tiger.ftk.spbstu.ru:/var/lib/firebird/435013/pharmacy.fdb' USER 'SYSDBA' PASSWORD 'masterkey';

CONNECT 'tiger.ftk.spbstu.ru:/var/lib/firebird/435013/pharmacy.fdb' USER 'SYSDBA' PASSWORD 'masterkey';

/\* Set charset. \*/

SET NAMES CYRL;

/\* Create tables. \*/

CREATE TABLE DISEASE

(

ID\_DISEASE INTEGER NOT NULL,

DISEASE\_NAME VARCHAR(256) NOT NULL COLLATE NONE,

CONSTRAINT PK\_DISEASE PRIMARY KEY (ID\_DISEASE)

);

COMMIT;

CREATE TABLE DRUG

(

ID\_DRUG INTEGER NOT NULL,

DRUG\_NAME VARCHAR(256) NOT NULL COLLATE NONE,

DRUG\_SHELF\_LIFE INTEGER NOT NULL,

DRUG\_CURRENT\_COST DECIMAL(18,4) NOT NULL,

DRUG\_RECIPE\_NEEDED CHAR(1) NOT NULL,

CONSTRAINT PK\_DRUG PRIMARY KEY (ID\_DRUG)

);

COMMIT;

CREATE TABLE CONTRAINDICATION

(

ID\_CONTRAINDICATION INTEGER NOT NULL,

ID\_DISEASE INTEGER NOT NULL REFERENCES DISEASE,

ID\_DRAG INTEGER NOT NULL REFERENCES DRUG,

CONSTRAINT PK\_CONTRAINDICATION PRIMARY KEY (ID\_CONTRAINDICATION)

);

COMMIT;

CREATE TABLE INCOMPATIBILITY

(

ID\_INCOMPATIBILITY INTEGER NOT NULL,

ID\_DRAG\_FIRST INTEGER NOT NULL REFERENCES DRUG,

ID\_DRAG\_SECOND INTEGER NOT NULL REFERENCES DRUG,

CONSTRAINT PK\_INCOMPATIBILITY PRIMARY KEY (ID\_INCOMPATIBILITY)

);

COMMIT;

CREATE TABLE PROVIDER

(

ID\_PROVIDER INTEGER NOT NULL,

PROVIDER\_NAME VARCHAR(256) NOT NULL,

CONSTRAINT PK\_PROVIDER PRIMARY KEY (ID\_PROVIDER)

);

COMMIT;

CREATE TABLE CONSIGNMENT

(

ID\_CONSIGNMENT INTEGER NOT NULL,

ID\_DRAG INTEGER NOT NULL REFERENCES DRUG,

ID\_PROVIDER INTEGER NOT NULL REFERENCES PROVIDER,

CONSIGNMENT\_DRAG\_COUNT INTEGER NOT NULL,

CONSIGNMENT\_ARRIVAL\_DATE TIMESTAMP NOT NULL,

CONSIGNMENT\_MANUFACTURE\_DATE TIMESTAMP NOT NULL,

CONSTRAINT PK\_CONSIGNMENT PRIMARY KEY (ID\_CONSIGNMENT)

);

COMMIT;

CREATE TABLE REQUEST

(

ID\_REQUEST INTEGER NOT NULL,

REQUEST\_DATE TIMESTAMP NOT NULL,

CONSTRAINT PK\_REQUEST PRIMARY KEY (ID\_REQUEST)

);

COMMIT;

CREATE TABLE REQUEST\_FROM\_CONSIGNMENT

(

ID\_REQUEST\_FROM\_CONSIGNMENT INTEGER NOT NULL,

ID\_REQUEST INTEGER NOT NULL REFERENCES REQUEST,

ID\_CONSIGNMENT INTEGER NOT NULL REFERENCES CONSIGNMENT,

DRUG\_PREVIOUS\_COST DECIMAL(18,4) NOT NULL,

DRUG\_COUNT INTEGER NOT NULL,

CONSTRAINT PK\_REQUEST\_FROM\_CONSIGNMENT PRIMARY KEY (ID\_REQUEST\_FROM\_CONSIGNMENT)

);

COMMIT;

Скрипт создает базу данных (если база данных уже создана, то удаляет старую), подключается к ней, устанавливает кодировку и создает таблицы со связями.

Теперь создадим скрипт, модифицирующий базу данных, в соответствии с индивидуальным заданием.

alter.sql:

CREATE TABLE DRUGSTORE

(

ID\_DRUGSTORE INTEGER NOT NULL,

DRUGSTORE\_ADDRESS VARCHAR(256) NOT NULL,

CONSTRAINT PK\_DRUGSTORE PRIMARY KEY (ID\_DRUGSTORE)

);

COMMIT;

DROP TABLE REQUEST\_FROM\_CONSIGNMENT;

CREATE TABLE REQUEST\_FROM\_DRUGSTORE

(

ID\_REQUEST\_FROM\_DRUGSTORE INTEGER NOT NULL,

ID\_REQUEST INTEGER NOT NULL REFERENCES REQUEST,

ID\_DRUGSTORE INTEGER NOT NULL REFERENCES DRUGSTORE,

DRUG\_PREVIOUS\_COST DECIMAL(18,4) NOT NULL,

DRUG\_COUNT INTEGER NOT NULL,

CONSTRAINT PK\_REQUEST\_FROM\_DRUGSTORE PRIMARY KEY (ID\_REQUEST\_FROM\_DRUGSTORE)

);

COMMIT;

ALTER TABLE CONSIGNMENT ADD ID\_DRUGSTORE INTEGER NOT NULL;

ALTER TABLE CONSIGNMENT ADD

CONSTRAINT FK\_CONSIGNMENT FOREIGN KEY (ID\_DRUGSTORE) REFERENCES DRUGSTORE;

COMMIT;

Скрипт создает новую таблицу DRUGSTORE, заменяет таблицу REQUEST\_FROM\_CONSIGNMENT, связанную с CONSIGNMENT на новую таблицу REQUEST\_FROM\_DRUGSTORE, связанную с DRUGSTORE. Также скрипт добавляет связь между таблицами CONSIGNMENT и DRUGSTORE по вторичному ключу ID\_DRUGSTORE, с помощью команды ALTER.

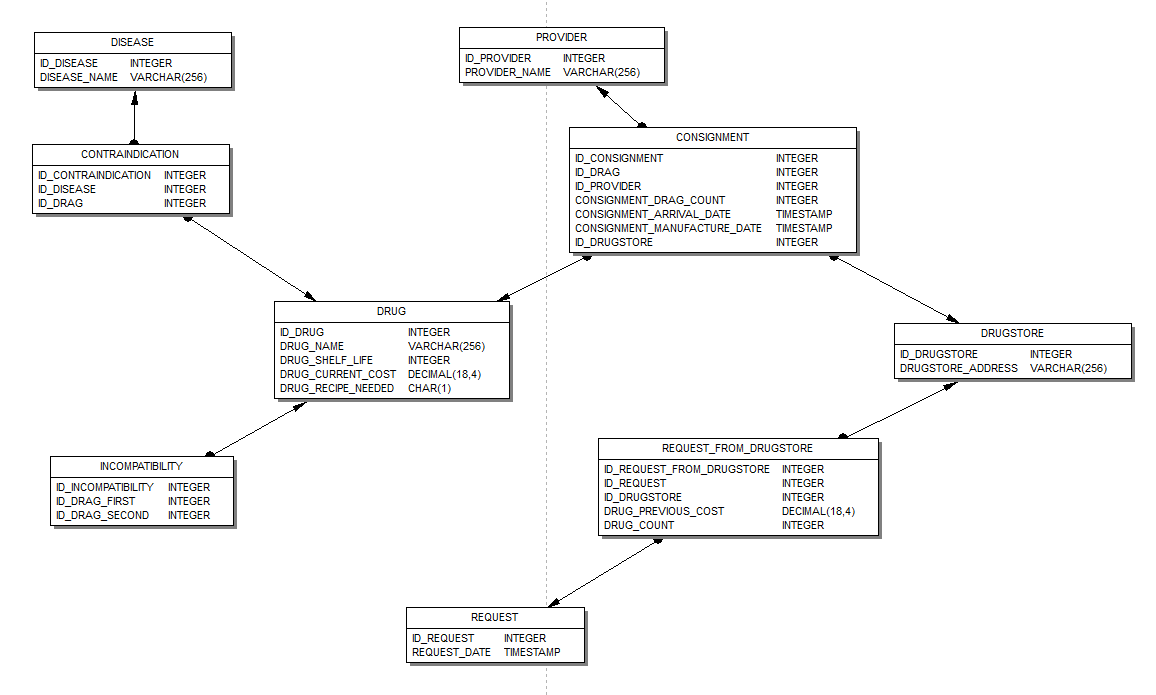


Рис. 2. SQL-диаграмма базы данных после изменения

Создадим скрипт, наполняющий базу данных небольшим объемом осмысленных данных:

insert.sql:

INSERT INTO DISEASE (ID\_DISEASE, DISEASE\_NAME) VALUES (1, 'Ветрянка');

INSERT INTO DISEASE (ID\_DISEASE, DISEASE\_NAME) VALUES (2, 'Корь');

INSERT INTO DISEASE (ID\_DISEASE, DISEASE\_NAME) VALUES (3, 'ОРЗ');

INSERT INTO DISEASE (ID\_DISEASE, DISEASE\_NAME) VALUES (4, 'Язва');

INSERT INTO DISEASE (ID\_DISEASE, DISEASE\_NAME) VALUES (5, 'Импотенция');

COMMIT;

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (1, 'Аспирин', 30, 32.22, 0);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (2, 'Нурафен', 40, 55.10, 0);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (3, 'Аспоркам', 20, 69.80, 1);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (4, 'Ношпа', 35, 20.90, 0);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (5, 'Диазолин', 50, 11.0, 0);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (6, 'Мезим', 30, 41.20, 0);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (7, 'Регидрон', 70, 5.40, 0);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (8, 'Флуоксетин', 50, 15.40, 1);

INSERT INTO DRUG (ID\_DRUG, DRUG\_NAME, DRUG\_SHELF\_LIFE, DRUG\_CURRENT\_COST, DRUG\_RECIPE\_NEEDED)

VALUES (9, 'Супрадин', 30, 25.0, 0);

COMMIT;

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (1, 1, 8);

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (2, 2, 3);

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (3, 1, 2);

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (4, 4, 3);

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (5, 6, 8);

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (6, 3, 7);

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (7, 9, 1);

INSERT INTO INCOMPATIBILITY (ID\_INCOMPATIBILITY, ID\_DRAG\_FIRST, ID\_DRAG\_SECOND) VALUES (8, 3, 4);

COMMIT;

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (1, 4, 8);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (2, 4, 3);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (3, 1, 6);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (4, 1, 2);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (5, 3, 3);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (6, 3, 7);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (7, 3, 9);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (8, 4, 2);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (9, 5, 6);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (10, 5, 1);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (11, 2, 4);

INSERT INTO CONTRAINDICATION (ID\_CONTRAINDICATION, ID\_DISEASE, ID\_DRAG) VALUES (12, 2, 5);

COMMIT;

INSERT INTO PROVIDER (ID\_PROVIDER, PROVIDER\_NAME) VALUES (1, 'Добрый доктор');

INSERT INTO PROVIDER (ID\_PROVIDER, PROVIDER\_NAME) VALUES (2, 'Доставщики9000');

INSERT INTO PROVIDER (ID\_PROVIDER, PROVIDER\_NAME) VALUES (3, 'Фармаштекер');

INSERT INTO PROVIDER (ID\_PROVIDER, PROVIDER\_NAME) VALUES (4, 'Компания Бориса');

INSERT INTO PROVIDER (ID\_PROVIDER, PROVIDER\_NAME) VALUES (5, 'Доктор силач');

COMMIT;

INSERT INTO DRUGSTORE (ID\_DRUGSTORE, DRUGSTORE\_ADDRESS) VALUES (1, 'Москва, ул. Ленина 21к1');

INSERT INTO DRUGSTORE (ID\_DRUGSTORE, DRUGSTORE\_ADDRESS) VALUES (2, 'Санкт-Петербург, пр. Энгельса 10к2');

INSERT INTO DRUGSTORE (ID\_DRUGSTORE, DRUGSTORE\_ADDRESS) VALUES (3, 'Санкт-Петербург, пр. Невский 5к1');

INSERT INTO DRUGSTORE (ID\_DRUGSTORE, DRUGSTORE\_ADDRESS) VALUES (4, 'Санкт-Петербург, ул. Парашютная 12к1');

COMMIT;

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (1, 5, 3, 120, '01-10-2011', '09-25-2010', 1);

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (2, 2, 5, 1000, '02-13-2011', '02-11-2011', 1);

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (3, 7, 4, 300, '01-14-2011', '01-10-2011', 4);

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (4, 1, 2, 300, '02-15-2011', '02-01-2011', 3);

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (5, 3, 1, 600, '01-10-2011', '11-11-2010', 3);

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (6, 4, 1, 200, '01-20-2011', '12-20-2010', 2);

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (7, 5, 5, 100, '01-24-2011', '01-21-2011', 2);

INSERT INTO CONSIGNMENT (ID\_CONSIGNMENT, ID\_DRAG, ID\_PROVIDER, CONSIGNMENT\_DRAG\_COUNT, CONSIGNMENT\_ARRIVAL\_DATE, CONSIGNMENT\_MANUFACTURE\_DATE, ID\_DRUGSTORE)

VALUES (8, 8, 3, 1000, '02-10-2011', '02-09-2011', 4);

COMMIT;

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (1, '03-17-2011');

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (2, '04-02-2011');

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (3, '03-11-2011');

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (4, '04-22-2011');

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (5, '03-10-2011');

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (6, '04-28-2011');

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (7, '04-10-2011');

INSERT INTO REQUEST (ID\_REQUEST, REQUEST\_DATE) VALUES (8, '03-02-2011');

COMMIT;

INSERT INTO REQUEST\_FROM\_DRUGSTORE (ID\_REQUEST\_FROM\_DRUGSTORE, ID\_REQUEST, ID\_DRUGSTORE, DRUG\_PREVIOUS\_COST, DRUG\_COUNT)

VALUES (1, 2, 2, 15.0, 10);

INSERT INTO REQUEST\_FROM\_DRUGSTORE (ID\_REQUEST\_FROM\_DRUGSTORE, ID\_REQUEST, ID\_DRUGSTORE, DRUG\_PREVIOUS\_COST, DRUG\_COUNT)

VALUES (2, 3, 2, 25.0, 5);

INSERT INTO REQUEST\_FROM\_DRUGSTORE (ID\_REQUEST\_FROM\_DRUGSTORE, ID\_REQUEST, ID\_DRUGSTORE, DRUG\_PREVIOUS\_COST, DRUG\_COUNT)

VALUES (3, 4, 3, 70.0, 2);

INSERT INTO REQUEST\_FROM\_DRUGSTORE (ID\_REQUEST\_FROM\_DRUGSTORE, ID\_REQUEST, ID\_DRUGSTORE, DRUG\_PREVIOUS\_COST, DRUG\_COUNT)

VALUES (4, 1, 1, 15.0, 20);

INSERT INTO REQUEST\_FROM\_DRUGSTORE (ID\_REQUEST\_FROM\_DRUGSTORE, ID\_REQUEST, ID\_DRUGSTORE, DRUG\_PREVIOUS\_COST, DRUG\_COUNT)

VALUES (5, 7, 3, 35.0, 10);

INSERT INTO REQUEST\_FROM\_DRUGSTORE (ID\_REQUEST\_FROM\_DRUGSTORE, ID\_REQUEST, ID\_DRUGSTORE, DRUG\_PREVIOUS\_COST, DRUG\_COUNT)

VALUES (6, 8, 4, 10.0, 40);

COMMIT;

Теперь заполним таблицы данными с помощью генератора данных среды разработки IBExpert.

Рассмотрим некоторые способы генерации данных в IBExpert: генерация случайных данных в некотором диапазоне, генерация случайных данных из списка, генерация случайных данных из другой таблицы. Для целочисленных типов существует генерация автоинкрементом, где можно также настроить величину шага. Эта опция обычно используется для генерирования первичных ключей.

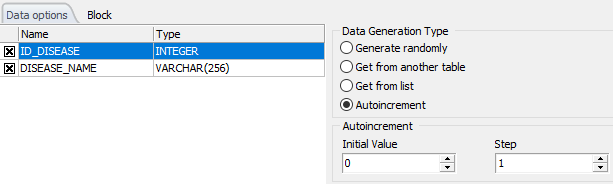


Рис. 3. Генерация автоинкрементом

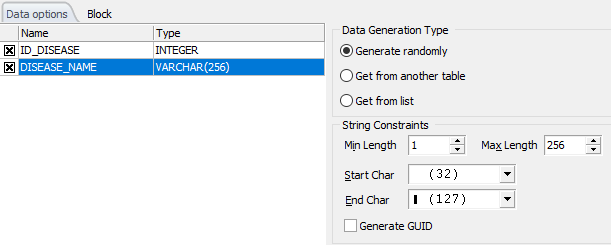


Рис. 4. Генерация случайных чисел в диапазоне

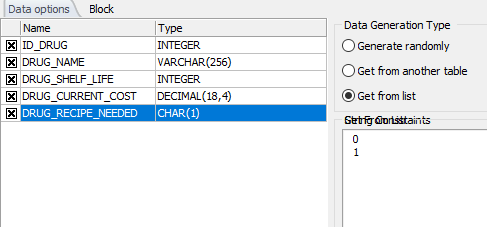


Рис. 5. Генерация случайных чисел из списка

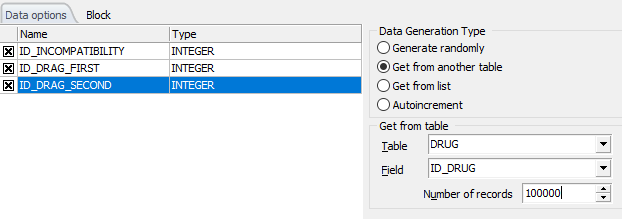


Рис. 6. Генерация случайных чисел из другой таблицы

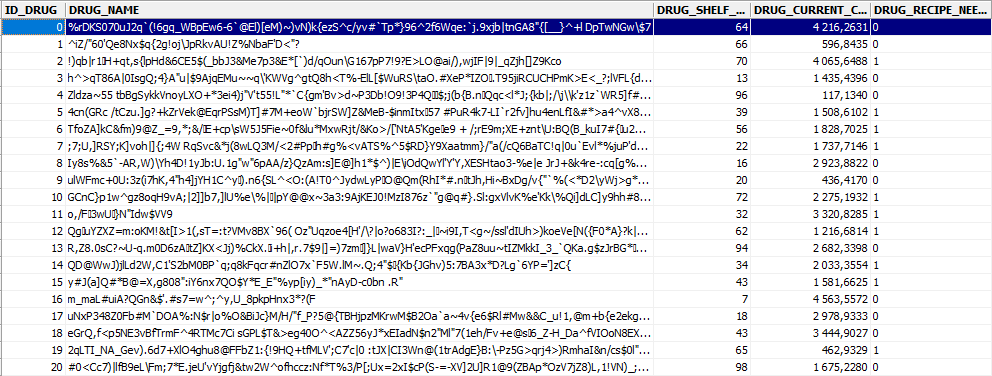


Рис 7. Пример сгенерированных данных

# 5. Вывод

В ходе работы я ознакомился с языком определения данных SQL-DDL. С его помощью можно определять, изменять и удалять структуры данных. При создании таблиц и заполнением их данными важно делать это в правильном порядке. Это связано со связями между таблицами, которые накладывают некоторые ограничения.

Работа была выполнена с помощью СУБД IBExpert, графический интерфейс которой значительно ускоряет процесс проектирования БД. Также данная среда разработки поддерживает множество полезных операций, таких как: создание ER-диаграмм, заполнение таблиц случайными данными, выполнение SQL скриптов и др.