

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.

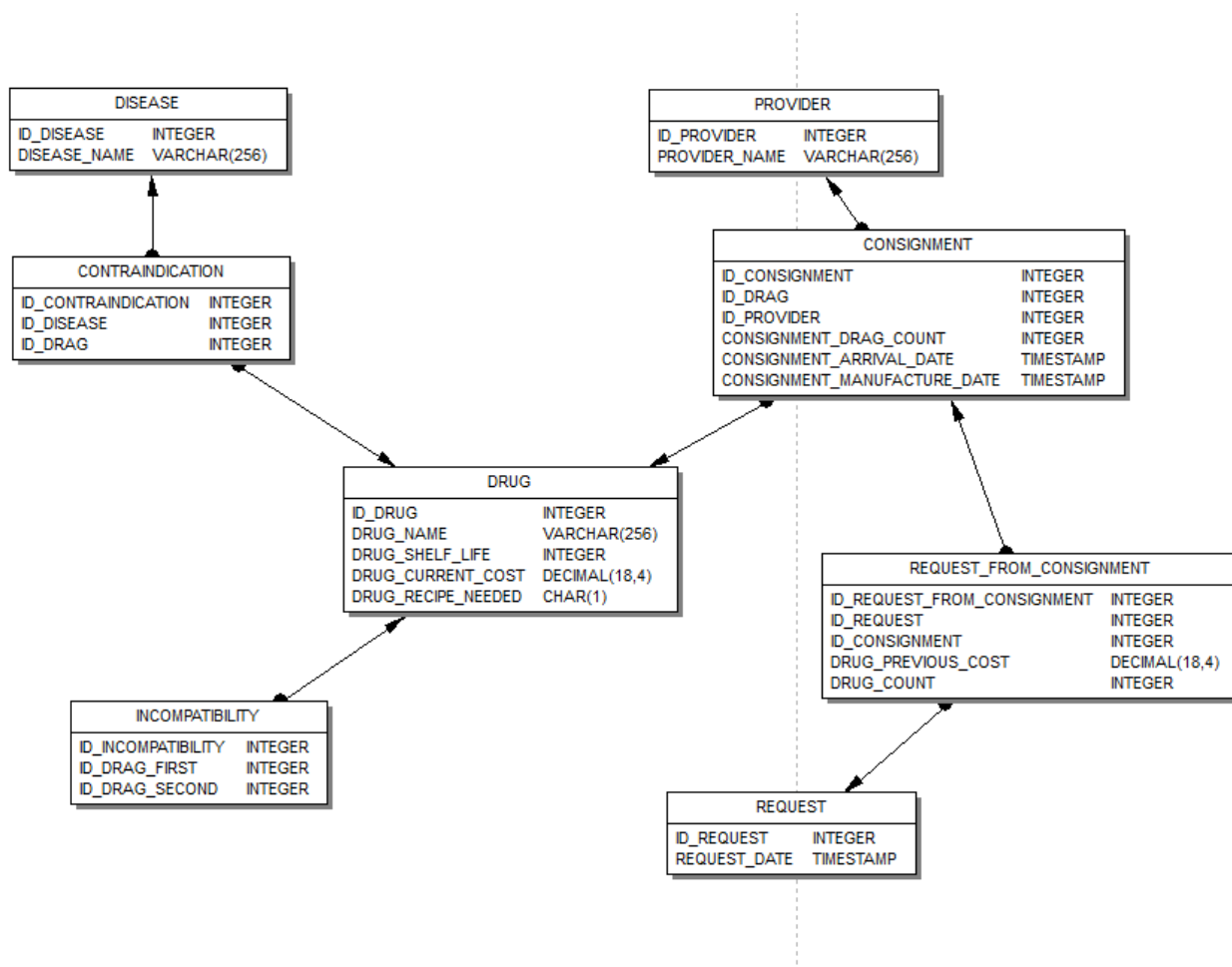


Рис. 1. SQL-диаграмма базы данных до изменения

create.sql:

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

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

DROPDATABASE;

/* Create database and connect. */

CREATEDATABASE '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_DRUG                   INTEGER            NOT NULL REFERENCES DRUG,
    CONSTRAINT PK_CONTRAINDICATION PRIMARY KEY (ID_CONTRAINDICATION)
);

COMMIT;

CREATE TABLE INCOMPATIBILITY
(
    ID_INCOMPATIBILITY        INTEGER            NOT NULL,
    ID_DRUG_FIRST             INTEGER            NOT NULL REFERENCES DRUG,
    ID_DRUG_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.

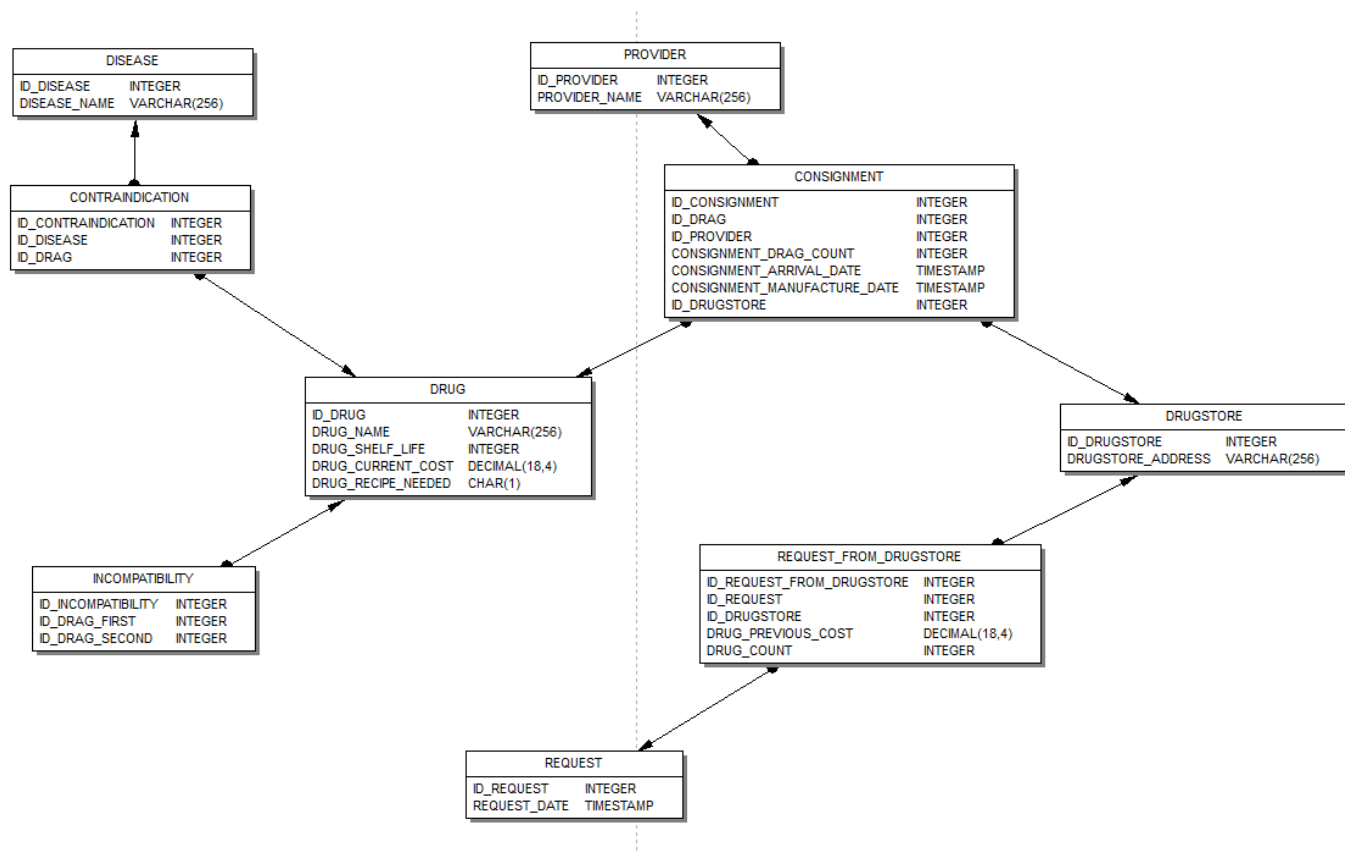


Рис. 2. SQL-диаграмма базы данных после изменения

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, 'Добрыйдоктор');
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, 'КомпанияБориса');
INSERT INTO PROVIDER (ID_PROVIDER, PROVIDER_NAME) VALUES (5, 'Докторсилач');

```

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.

Data options Block

Name	Type
<input checked="" type="checkbox"/> ID_DISEASE	INTEGER
<input checked="" type="checkbox"/> DISEASE_NAME	VARCHAR(256)

Data Generation Type

☐ Generate randomly
☐ Get from another table
☐ Get from list
☒ Autoincrement

Autoincrement

Initial Value: 0 Step: 1

Рис. 3. Генерация автоинкрементом

Data options Block

Name	Type
<input checked="" type="checkbox"/> ID_DISEASE	INTEGER
<input checked="" type="checkbox"/> DISEASE_NAME	VARCHAR(256)

Data Generation Type

☒ Generate randomly
☐ Get from another table
☐ Get from list

String Constraints

Min Length: 1 Max Length: 256

Start Char: (32)

End Char: | (127)

☐ Generate GUID

Рис. 4. Генерация случайных чисел в диапазоне

Data options Block

Name	Type
<input checked="" type="checkbox"/> ID_DRUG	INTEGER
<input checked="" type="checkbox"/> DRUG_NAME	VARCHAR(256)
<input checked="" type="checkbox"/> DRUG_SHELF_LIFE	INTEGER
<input checked="" type="checkbox"/> DRUG_CURRENT_COST	DECIMAL(18,4)
<input checked="" type="checkbox"/> DRUG_RECIPE_NEEDED	CHAR(1)

Data Generation Type

☐ Generate randomly
☐ Get from another table
☒ Get from list

Get from list

0
1

Рис. 5. Генерация случайных чисел из списка

Data options Block

Name	Type
<input checked="" type="checkbox"/> ID_INCOMPATIBILITY	INTEGER
<input checked="" type="checkbox"/> ID_DRAG_FIRST	INTEGER
<input checked="" type="checkbox"/> ID_DRAG_SECOND	INTEGER

Data Generation Type

☐ Generate randomly
☒ Get from another table
☐ Get from list
☐ Autoincrement

Get from table

Table: DRUG

Field: ID_DRUG

Number of records: 100000

Рис. 6. Генерация случайных чисел из другой таблицы

ID_DRUG	DRUG_NAME	DRUG_SHELF_...	DRUG_CURRENT_C...	DRUG_RECIPE_NEE...
0	%rDKS070uJ2g'(l6gq_WBpEw6-6'@Ej)(eM)~)vN)k(ezS^c/vv#`Tp*)96^2f6Wqe:'.9xjb tnGA8'{'f_')^+H DpTwNGw 57	64	4 216,2631	0
1	^iZ/'60'Qe8Nx\$q(2gloj)prkvAUlZ%aNbaFD<?`	66	596,8435	0
2	!)qb lR+qt,s(lpHd&6CE5\$(_bbJ3&Me7p3&E*["`d/qOun\G167pP7I9E>LO@al/),wJf9 _qzjh[]Z9Kco	70	4 065,6488	1
3	h^>qT86A)0IsgQ;4)A`u \$9AjqEMu~~~q\KVVg^gtQ8h<T%-ELL[\$WURS]taO.#XeP*IZO:l.T95j)RCUCHPMK>E<_?; VFL(d...	13	1 435,4396	0
4	Zldza~55 tbBgSykVnoyLXO+*3ei4j)"t55IL"*`C(gmBv>d~P3Db!O9!3P4Q3\$;j)(b(n3Qqc<l*J;{kb ; j\k'z1z`WR5)f#...	96	117,1340	0
5	4cn(GRC/tCzu.)g?+kZrVek@EqP5sM)T]#7M+eoW`bjrSW]Z&MeB-\$inmIbx57 #PuR4k7-L' r2fv hu4enlf&#*>a4^vX8...	39	1 508,6102	1
6	TfoZa)kC&fm)9@Z_~9,*;+E+cp sW5J5Fie~0f&lu*MxwRjt/&Ko>/[NtA5KgeJe9 + /rE9m;XE+znt\U:BQ(B_kuI7#(Ju2...	56	1 828,7025	1
7	;7;U,]RSY;Kjvoh[];4W RqSvc&*j(8wLQ3M/<2#PpJh#g%<vATS%^5\$RD}Y9Xaatmm)/a/(cQ6BaTC!q 0u`Evl*%juP'd...	22	1 737,7146	1
8	Iy8s%&5`-AR,W)\Yh4D!1yJb:U.1g"w"6pAA/z}QzAm:s]E@]h1*\$^}JE\OdQwYfY'Y,XESHTao3-%e e.JrJ+&k4re-:cq g%...	16	2 923,8822	0
9	uWfmc+0U:3z(i7hK,4h4j)YH1C^yJ).n6(SL^<O:(AIT0^JydwLyPJO@Qm(RhI*#.nitJh,Hi~BxDg/v{"`%(<*D2 yWj>g*...	20	436,4170	0
10	GCnC)p1w^gz8oqH9vA;[2]]b7, U%e\% l pY@x~3a3:9AJKEJ0IMzI876z`"g@q#}.Sl:gxVlvK%eYk\%Qj dLC y9hh#8...	72	2 275,1932	1
11	o, /F13wUJ\$Ntdw\$V9	32	3 320,8285	1
12	QgJuYZKZ~m:okMf&t[I]>1,(sT=:t?VMv8BX`96(Oz"Uqzoe4[H'/? o?o683I?:_l]~I9I,T<g~/ssf'dIUh>)koeVe[N({F0*A}?)k ...	62	1 216,6814	1
13	R,Z8.0sC?~U-q.mOD6zAItZ)KX<J))%CKX.I+H ,r.7\$9[]=)7zm]]L waV)H'edPFxqg(PaZ8uu~tIZMkkI_3`QKa.g\$zJrBG*Il...	94	2 682,3398	0
14	QD@WwJ)j LdZW,C1'S2bMOBP`q;q8kFqr#nZIO7x`F5W,IM~Q;4*\$3{Kb{JGhv}5:7BA3x*D?Lg`6YP~]zC{	34	2 033,3554	1
15	y#J[a]Q#*B@=X,g808":iY6nx7QO\$Y"E_E"%yp[iy]_""hAyD-c0bn.R"	43	1 581,6625	1
16	m_maL#uiA?QGn&\$'.#s7=w~:~y,U_8pkpHnx3*?(F	7	4 563,5572	0
17	uNxP34820Fb#M`DOA%:N\$R o%O&BIc)M/H/f_P?5@(TBHjzpMkrWm\$B20a`a~4v(e6\$gJ#Mw&C_uI_1,@m+b(e2ekg...	18	2 978,9333	0
18	eGrQ,f<p5NE3vBfTmF^4RTMc7Ci sGPLST&>eg40O^<AZZ56yJ*xEladN\$N2"MI"7(1eh/Fv+e@sI6_Z-H_Da^fVIOoN8EX...	43	3 444,9027	0
19	2qLTI_NA_Gev).6d7xXIO4ghu8@FFbZ1:{9HQ+tfMLV';C7c O :tJX CI3Wn@{(1trAdgE)B:~Pz5G>qqr4>)RmhaI&n/cs\$0l'...	65	462,9329	1
20	#0<Cc7)l f89eL\Fm;7*E.jeU'vYjgJ&tw2W^ofhccz:Nf*%3/P;Ux=2xI\$<P(S-=XV)2U]R1@9(ZBAp*Ozv7Z8)L,1 VN)_j...	98	1 675,2280	0

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 IVExpert, 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.