

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: Язык SQL-DDL

Выполнил студент гр. 43501/1

(подпись) Г. А. Жемелев

Преподаватель

(подпись) А. В. Мяснов

“26” ноября 2016 г.

Санкт-Петербург

2016

Цель работы

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

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).

Ход работы

Согласно индивидуальному заданию «Интернет-магазин компьютерных комплектующих» была составлена SQL-схема БД, приведенная на Рис. 1. Для создания соответствующей базы данных через утилиту Firebird ISQL был написан следующий скрипт:

```
--<status> ::= CREATED(0) | CONFIRMED(1) | READY(2) | COMPLETED(3) | CANCELLED(4)

CREATE DOMAIN D_status AS smallint DEFAULT 0 CHECK (VALUE IS NULL OR VALUE IN (0, 1, 2, 3, 4));
CREATE DOMAIN D_rating AS smallint DEFAULT NULL CHECK (VALUE IN (1, 2, 3, 4, 5));
CREATE DOMAIN D_money AS numeric(9,2) DEFAULT NULL CHECK (VALUE >= 0);

CREATE GENERATOR Seq_Article;
CREATE GENERATOR Seq_OrderNo;
CREATE GENERATOR Seq_arrID;
CREATE GENERATOR Seq_manID;
CREATE GENERATOR Seq_revID;
CREATE GENERATOR Seq_stoID;
```

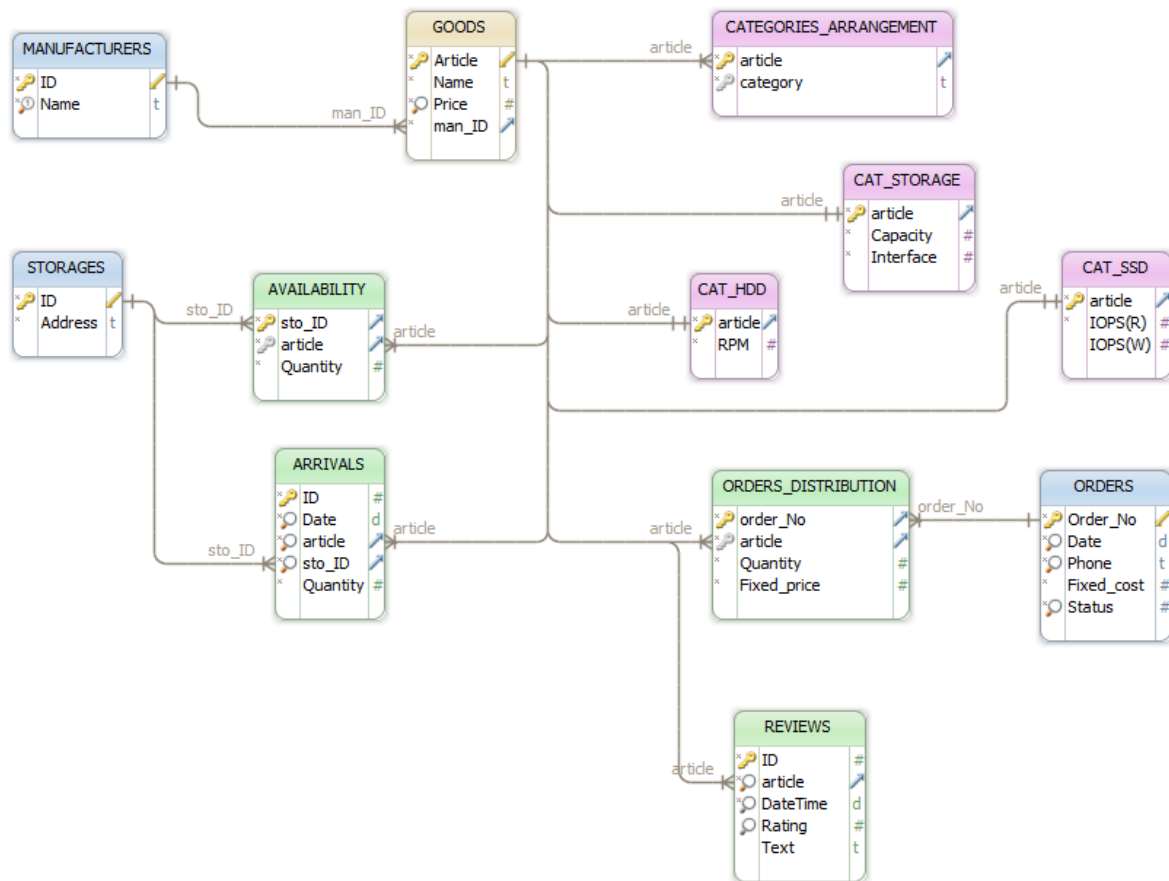


Рис. 1. Первоначальная схема разрабатываемой базы данных

```

CREATE TABLE MANUFACTURERS (
    ID                integer      NOT NULL PRIMARY KEY,
    Name              varchar(64)  NOT NULL UNIQUE
);

CREATE TABLE STORAGES (
    ID                integer      NOT NULL PRIMARY KEY,
    Address            varchar(256) NOT NULL
);

CREATE TABLE ORDERS (
    Order_No          integer      NOT NULL PRIMARY KEY,
    Order_Date        date         NOT NULL,
    Phone              varchar(32) NOT NULL,
    --Fixed_cost       D_money      NOT NULL,
    Status             D_status    NOT NULL
);

CREATE INDEX Idx_ORDERS_date ON ORDERS ( Order_Date );
CREATE INDEX Idx_ORDERS_phone ON ORDERS ( Phone );
CREATE INDEX Idx_ORDERS_status ON ORDERS ( Status );

CREATE TABLE GOODS (
    Article            integer      NOT NULL PRIMARY KEY,
    Name               varchar(256) NOT NULL,
    Price              D_money      ,
    man_ID             integer      NOT NULL REFERENCES MANUFACTURERS(ID)
);

CREATE INDEX Idx_GOODS_Price ON GOODS ( Price );
  
```

```

CREATE TABLE ARRIVALS (
    ID                integer        NOT NULL PRIMARY KEY,
    Arr_Date          date           NOT NULL,
    article            integer        NOT NULL REFERENCES GOODS(Article),
    sto_ID             integer        NOT NULL REFERENCES STORAGES(ID),
    Quantity           integer        NOT NULL CHECK(Quantity > 0)
);

CREATE INDEX Idx_ARRIVALS_date ON ARRIVALS ( Arr_Date );
CREATE INDEX Idx_ARRIVALS_article ON ARRIVALS ( article );
CREATE INDEX Idx_ARRIVALS_sto_ID ON ARRIVALS ( sto_ID );

CREATE TABLE AVAILABILITY (
    sto_ID             integer        NOT NULL REFERENCES STORAGES(ID),
    article            integer        NOT NULL REFERENCES GOODS(Article),
    Quantity           integer        NOT NULL CHECK(Quantity >= 0),
    CONSTRAINT pk_AVAILABILITY PRIMARY KEY ( sto_ID, article )
);

CREATE TABLE ORDERS_DISTRIBUTION (
    order_No           integer        NOT NULL REFERENCES ORDERS(Order_No),
    article            integer        NOT NULL REFERENCES GOODS(Article),
    Quantity           integer        NOT NULL CHECK(Quantity > 0),
    Fixed_price        D_money        ,
    CONSTRAINT pk_ORDERS_DISTRIBUTION PRIMARY KEY ( order_No, article )
);

CREATE TABLE REVIEWS (
    ID                 bigint         NOT NULL PRIMARY KEY,
    article            integer        NOT NULL REFERENCES GOODS(Article),
    DateTime           timestamp      NOT NULL,
    Rating             D_rating       ,
    Text               varchar(2000)
);

CREATE INDEX Idx_REVIEWS_articleWdatetime ON REVIEWS ( article, DateTime );
CREATE INDEX Idx_REVIEWS_articleWrating ON REVIEWS ( article, Rating );

CREATE TABLE CATEGORIES_ARRANGEMENT (
    article            integer        NOT NULL REFERENCES GOODS(Article),
    Category           varchar(64)    NOT NULL,
    CONSTRAINT pk_CATEGORIES_ARRANGEMENT PRIMARY KEY ( article, Category )
);

CREATE TABLE CAT_STORAGE (
    article            integer        NOT NULL PRIMARY KEY REFERENCES GOODS(Article),
    Capacity           integer        NOT NULL CHECK(Capacity > 0),
    Interface          varchar(64)    NOT NULL
);

CREATE TABLE CAT_HDD (
    article            integer        NOT NULL PRIMARY KEY REFERENCES GOODS(Article),
    RPM               integer        NOT NULL CHECK(RPM > 0)
);

CREATE TABLE CAT_SSD (
    article            integer        NOT NULL PRIMARY KEY REFERENCES GOODS(Article),
    "IOPS(R)"         integer        ,
    "IOPS(W)"         integer        ,
    --CONSTRAINT pk_CAT_SSD PRIMARY KEY ( article )
);
COMMIT;

```

Скрипт для заполнения созданной БД данными имеет следующий вид:

```
INSERT INTO manufacturers VALUES
(NEXT VALUE FOR Seq_manID, 'Asus');
INSERT INTO manufacturers VALUES
(NEXT VALUE FOR Seq_manID, 'MSI');
INSERT INTO manufacturers VALUES
(NEXT VALUE FOR Seq_manID, 'Intel');
INSERT INTO manufacturers VALUES
(NEXT VALUE FOR Seq_manID, 'AMD');
INSERT INTO manufacturers VALUES
(NEXT VALUE FOR Seq_manID, 'Samsung');
INSERT INTO manufacturers VALUES
(NEXT VALUE FOR Seq_manID, 'Seagate');
INSERT INTO manufacturers VALUES
(NEXT VALUE FOR Seq_manID, 'Logitech');

INSERT INTO storages VALUES
(NEXT VALUE FOR Seq_stoID, 'СПб, ул. Пушкина, 76');
INSERT INTO storages VALUES
(NEXT VALUE FOR Seq_stoID, 'СПб, Дунайский пр., 7к1');

INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'Nvidia GeForce GTX 960', 14500.00, (SELECT ID FROM
manufacturers WHERE Name = 'Asus'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'Nvidia GeForce GTX 960', 12760.00, (SELECT ID FROM
manufacturers WHERE Name = 'MSI'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'Nvidia GeForce GTX 970', 18000.00, (SELECT ID FROM
manufacturers WHERE Name = 'Asus'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'Nvidia GeForce GTX 980', 24400.00, (SELECT ID FROM
manufacturers WHERE Name = 'Asus'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'Core i5-6400', 14153.50, (SELECT ID FROM manufacturers
WHERE Name = 'Intel'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'Core i7-6700K', 22165.00, (SELECT ID FROM manufacturers
WHERE Name = 'Intel'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'HD Pro Webcam C920', 8221.00, (SELECT ID FROM
manufacturers WHERE Name = 'Logitech'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'Webcam C170', 1279.90, (SELECT ID FROM manufacturers
WHERE Name = 'Logitech'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, '850 EVO 120Gb MZ-75E120BW', 5000.00, (SELECT ID FROM
manufacturers WHERE Name = 'Samsung'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, '850 EVO 250Gb MZ-75E250BW', 7550.00, (SELECT ID FROM
manufacturers WHERE Name = 'Samsung'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, '850 EVO 250Gb MZ-N5E250BW', 7920.00, (SELECT ID FROM
manufacturers WHERE Name = 'Samsung'));
INSERT INTO goods VALUES
(NEXT VALUE FOR Seq_Article, 'STDR2000200', 5340.00, (SELECT ID FROM manufacturers
WHERE Name = 'Seagate'));

INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '25.09.2016', 1, 1, 20);
```

```

INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '25.09.2016', 3, 1, 10);
INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '25.09.2016', 4, 1, 5);
INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '26.09.2016', 2, 2, 30);
INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '26.09.2016', 5, 1, 25);
INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '26.09.2016', 5, 2, 20);
INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '29.09.2016', 8, 1, 50);
INSERT INTO arrivals VALUES
(NEXT VALUE FOR Seq_arrID, '30.09.2016', 8, 1, 50);

INSERT INTO orders VALUES
(NEXT VALUE FOR Seq_OrderNo, '30.09.2016', '+79119786196', 3);

INSERT INTO orders_distribution VALUES
(1, 1, 1, 12999.00); -- Купил GTX 960 по акции!

INSERT INTO reviews VALUES
(NEXT VALUE FOR Seq_revID, 4, '29.09.2016 04:01:31', 5, 'Хорошая видеокарта! Покупать я её, конечно же, не буду.');
```

```

INSERT INTO reviews VALUES
(NEXT VALUE FOR Seq_revID, 1, '03.10.2016 17:23:45', 4, 'Хорошая видеокарта по привлекательной цене, только дроссели свистят.');
```

```

INSERT INTO categories_arrangement VALUES (9, 'Накопители');
INSERT INTO categories_arrangement VALUES (10, 'Накопители');
INSERT INTO categories_arrangement VALUES (11, 'Накопители');
INSERT INTO categories_arrangement VALUES (12, 'Накопители');
INSERT INTO categories_arrangement VALUES (9, 'SSD');
INSERT INTO categories_arrangement VALUES (10, 'SSD');
INSERT INTO categories_arrangement VALUES (11, 'SSD');
INSERT INTO categories_arrangement VALUES (12, 'HDD');
```

```

INSERT INTO cat_storage VALUES (9, 120, 'SATA');
INSERT INTO cat_storage VALUES (10, 250, 'SATA');
INSERT INTO cat_storage VALUES (11, 250, 'M.2');
INSERT INTO cat_storage VALUES (12, 2000, 'USB 3.0');
```

```

INSERT INTO cat_hdd VALUES (12, 7200);

INSERT INTO cat_ssd VALUES (9, NULL, 88000);
INSERT INTO cat_ssd VALUES (10, NULL, 88000);
INSERT INTO cat_ssd VALUES (11, NULL, 89000);

COMMIT;
```

Пример вывода содержимого таблицы товаров GOODS приведен на рис. 2.

В соответствии с заданием преподавателя схема базы данных была изменена таким образом, чтобы добавление новой категории не требовало добавления новых таблиц (рис. 3).

```
SQL> select * from GOODS;
```

ARTICLE	NAME	PRICE	MAN_ID
=====	=====	=====	=====
1	NVidia GeForce GTX 960	14500.00	1
2	NVidia GeForce GTX 960	12760.00	2
3	NVidia GeForce GTX 970	18000.00	1
4	NVidia GeForce GTX 980	24400.00	1
5	Core i5-6400	14153.50	3
6	Core i7-6700K	22165.00	3
7	HD Pro Webcam C920	8221.00	7
8	Webcam C170	1279.90	7
9	850 EVO 120Gb MZ-75E120BW	5000.00	5
10	850 EVO 250Gb MZ-75E250BW	7550.00	5
11	850 EVO 250Gb MZ-N5E250BW	7920.00	5
12	STDR2000200	5340.00	6

Рис. 2. Содержимое таблицы товаров

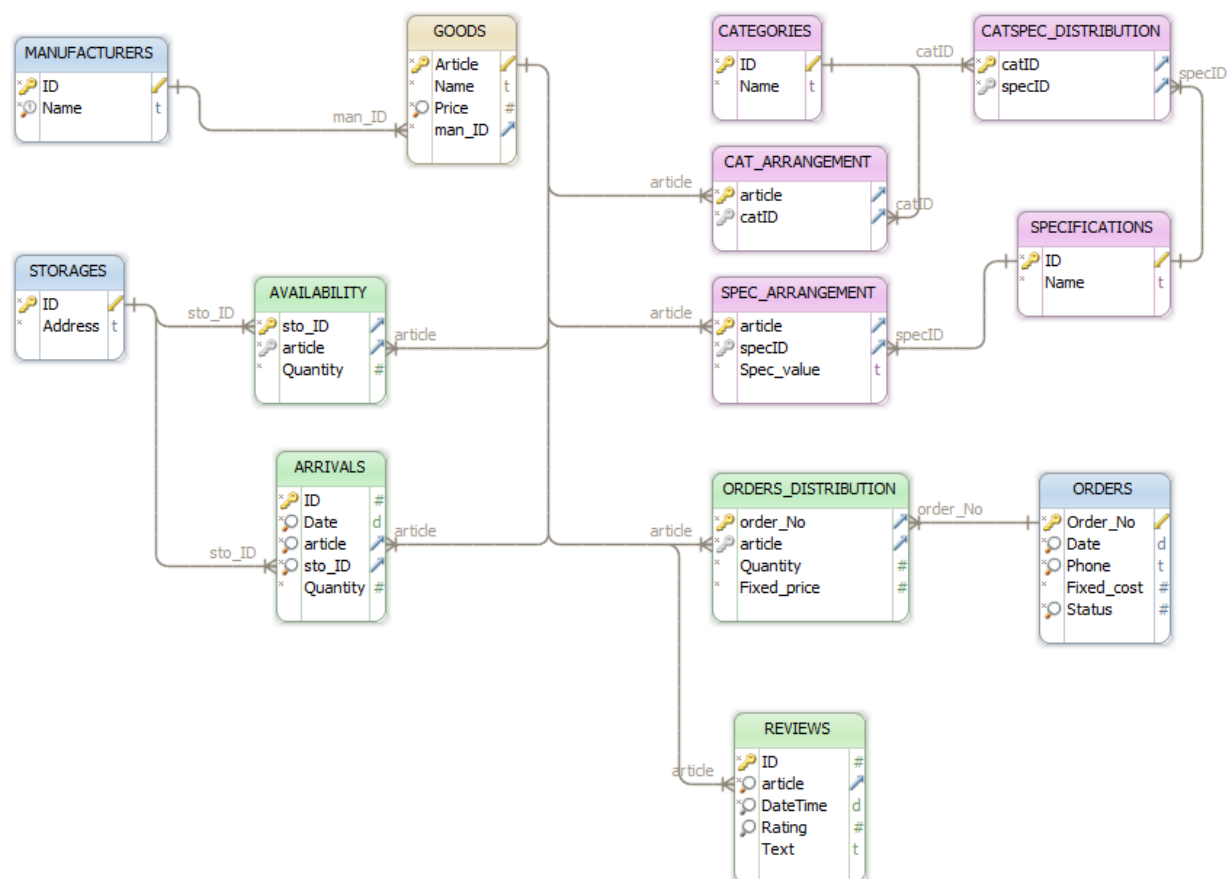


Рис. 3. Измененная схема разрабатываемой базы данных

Для внесения изменений был написан следующий скрипт:

```
CREATE GENERATOR Seq_catID;
CREATE GENERATOR Seq_specID;

CREATE TABLE CATEGORIES (
    ID                integer      NOT NULL PRIMARY KEY,
    Name              varchar(64)  NOT NULL
);

SET TERM ^;
CREATE TRIGGER set_catID BEFORE INSERT ON CATEGORIES
AS
BEGIN
    IF(NEW.ID IS NULL) THEN
        NEW.ID = GEN_ID(Seq_catID, 1);
END^
SET TERM ;^

INSERT INTO CATEGORIES (Name) SELECT DISTINCT Category FROM CATEGORIES_ARRANGEMENT
ORDER BY 1 ASC;

CREATE TABLE SPECIFICATIONS (
    ID                integer      NOT NULL PRIMARY KEY,
    Name              varchar(64)  NOT NULL
);

SET TERM ^;
CREATE TRIGGER set_specID BEFORE INSERT ON SPECIFICATIONS
AS
BEGIN
    IF(NEW.ID IS NULL) THEN
        NEW.ID = GEN_ID(Seq_specID, 1);
END^
SET TERM ;^

INSERT INTO SPECIFICATIONS (Name)
SELECT * FROM (select RDB$FIELD_NAME as Spec from RDB$RELATION_FIELDS where
RDB$RELATION_NAME = 'CAT_SSD') WHERE Spec <> 'ARTICLE'
UNION
SELECT * FROM (select RDB$FIELD_NAME as Spec from RDB$RELATION_FIELDS where
RDB$RELATION_NAME = 'CAT_HDD') WHERE Spec <> 'ARTICLE'
UNION
SELECT * FROM (select RDB$FIELD_NAME as Spec from RDB$RELATION_FIELDS where
RDB$RELATION_NAME = 'CAT_STORAGE') WHERE Spec <> 'ARTICLE'
ORDER BY 1 ASC;

COMMIT;

CREATE TABLE CATSPEC_DISTRIBUTION (
    catID             integer      NOT NULL REFERENCES CATEGORIES (ID),
    specID            integer      NOT NULL REFERENCES SPECIFICATIONS(ID),
    CONSTRAINT pk_CATSPEC_DISTRIBUTION PRIMARY KEY ( catID, specID )
);

INSERT INTO CATSPEC_DISTRIBUTION VALUES (1, 5);
INSERT INTO CATSPEC_DISTRIBUTION VALUES (2, 3);
INSERT INTO CATSPEC_DISTRIBUTION VALUES (2, 4);
INSERT INTO CATSPEC_DISTRIBUTION VALUES (3, 1);
INSERT INTO CATSPEC_DISTRIBUTION VALUES (3, 2);
```



```

COMMIT;

CREATE TABLE CAT_ARRANGEMENT (
    article          integer          NOT NULL REFERENCES GOODS (Article),
    catID            integer          NOT NULL REFERENCES CATEGORIES (ID),
    CONSTRAINT pk_CAT_ARRANGEMENT PRIMARY KEY ( article, catID )
);

INSERT INTO CAT_ARRANGEMENT VALUES (9, 3);
INSERT INTO CAT_ARRANGEMENT VALUES (9, 2);
INSERT INTO CAT_ARRANGEMENT VALUES (10, 3);
INSERT INTO CAT_ARRANGEMENT VALUES (10, 2);
INSERT INTO CAT_ARRANGEMENT VALUES (11, 3);
INSERT INTO CAT_ARRANGEMENT VALUES (11, 2);
INSERT INTO CAT_ARRANGEMENT VALUES (12, 3);
INSERT INTO CAT_ARRANGEMENT VALUES (12, 1);

COMMIT;

CREATE TABLE SPEC_ARRANGEMENT (
    article          integer          NOT NULL REFERENCES GOODS (Article),
    specID           integer          NOT NULL REFERENCES SPECIFICATIONS(ID),
    Spec_value       varchar(256)    ,
    CONSTRAINT pk_SPEC_ARRANGEMENT PRIMARY KEY ( article, specID )
);

INSERT INTO SPEC_ARRANGEMENT VALUES (9, 1, '120 Gb');
INSERT INTO SPEC_ARRANGEMENT VALUES (9, 2, 'SATA');
INSERT INTO SPEC_ARRANGEMENT VALUES (9, 3, NULL);
INSERT INTO SPEC_ARRANGEMENT VALUES (9, 4, '88000');
INSERT INTO SPEC_ARRANGEMENT VALUES (10, 1, '250 Gb');
INSERT INTO SPEC_ARRANGEMENT VALUES (10, 2, 'SATA');
INSERT INTO SPEC_ARRANGEMENT VALUES (10, 3, NULL);
INSERT INTO SPEC_ARRANGEMENT VALUES (10, 4, '88000');
INSERT INTO SPEC_ARRANGEMENT VALUES (11, 1, '250 Gb');
INSERT INTO SPEC_ARRANGEMENT VALUES (11, 2, 'M.2');
INSERT INTO SPEC_ARRANGEMENT VALUES (11, 3, NULL);
INSERT INTO SPEC_ARRANGEMENT VALUES (11, 4, '89000');
INSERT INTO SPEC_ARRANGEMENT VALUES (12, 1, '2000 Gb');
INSERT INTO SPEC_ARRANGEMENT VALUES (12, 2, 'USB 3.0');
INSERT INTO SPEC_ARRANGEMENT VALUES (12, 5, '7200');

COMMIT;

DROP TABLE CATEGORIES_ARRANGEMENT;
DROP TABLE CAT_STORAGE;
DROP TABLE CAT_SSD;
DROP TABLE CAT_HDD;

COMMIT;

```

Далее были изучены основные возможности IVExpert, например, просмотр содержимого таблиц (рис. 4) и создание ER-диаграммы существующей базы данных с помощью модуля Database Designer (функция «Reverse Engineer...»). Полученная диаграмма приведена на рис. 5.

ARTICLE	NAME	PRICE	MAN_ID
1	Nvidia GeForce GTX 960	14 500,00	1
2	Nvidia GeForce GTX 960	12 760,00	2
3	Nvidia GeForce GTX 970	18 000,00	1
4	Nvidia GeForce GTX 980	24 400,00	1
5	Core i5-6400	14 153,50	3
6	Core i7-6700K	22 165,00	3
7	HD Pro Webcam C920	8 221,00	7
8	Webcam C170	1 279,90	7
9	850 EVO 120Gb MZ-75E120BW	5 000,00	5
10	850 EVO 250Gb MZ-75E250BW	7 550,00	5
11	850 EVO 250Gb MZ-N5E250BW	7 920,00	5
12	STD2000200	5 340,00	6

Рис. 4. Содержимое таблицы товаров (просмотр через IBExpert)

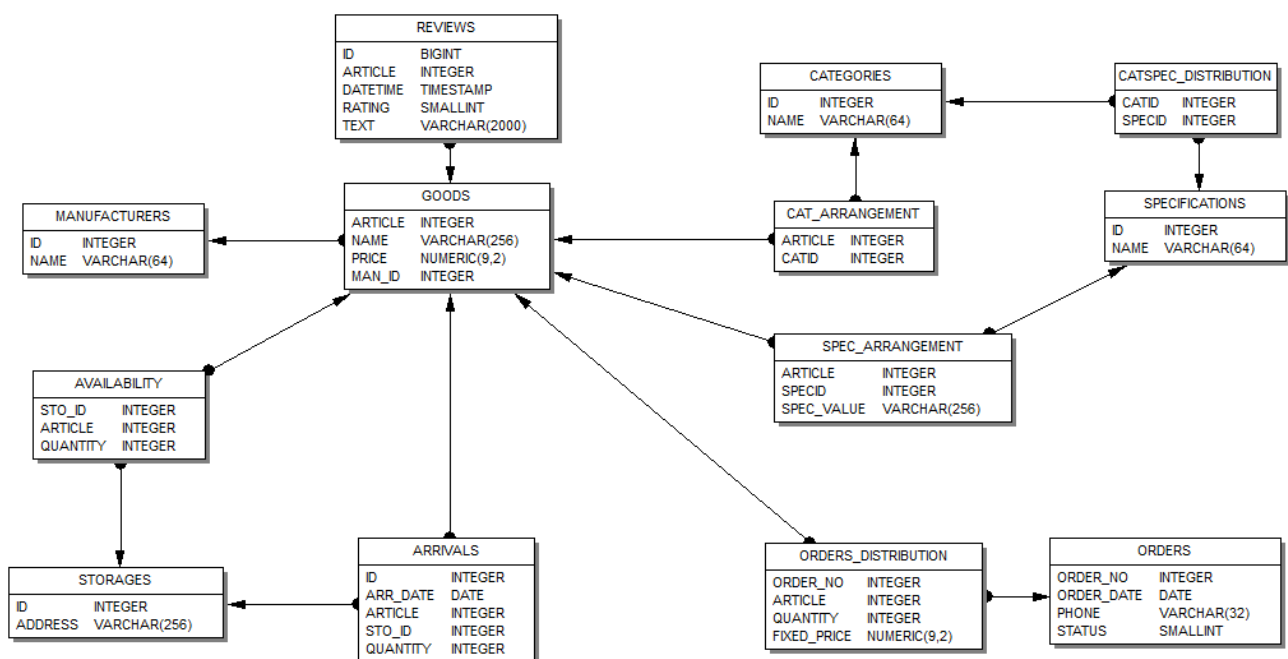


Рис. 5. ER-диаграмма измененной базы данных

При помощи инструмента «Test data generator» была произведена автоматическая генерация данных: по 100000 новых записей для трех таблиц MANUFACTURERS, STORAGES и ORDERS. Пример задания параметров генерации приведен на рис. 6. Результаты для перечисленных таблиц представлены на рис. 7-9 соответственно.

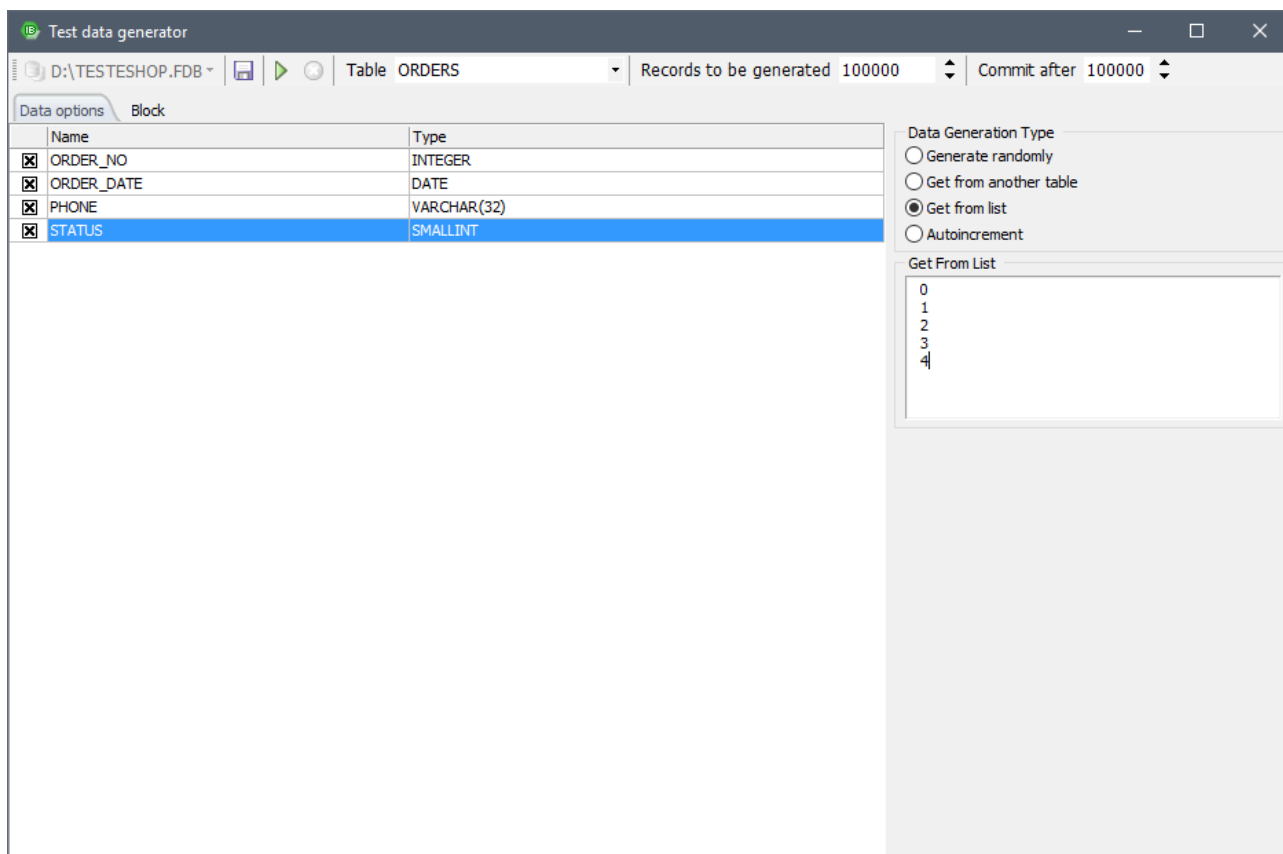


Рис. 8. Задание параметров генерации для таблицы ORDERS

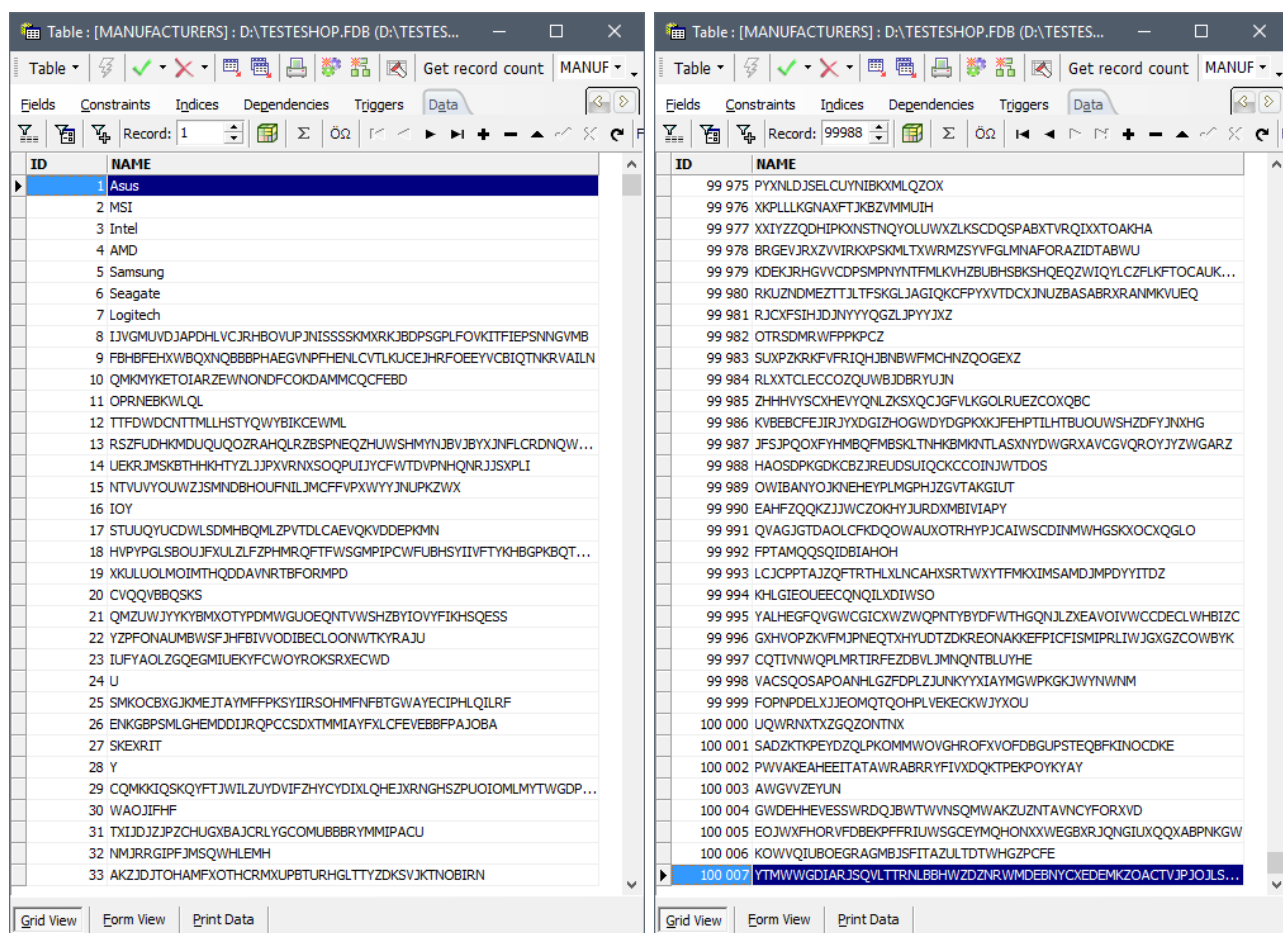


Рис. 9. Результаты генерации данных (таблица MANUFACTURERS)

ID	ADDRESS
1	UjY, r, UuCE, 7Y
2	UjY, r, @6CE@ la, 7E1
3	2Kx6' srMjB[^8Uxspxbw
4	Ws^4[m4tpdk>:GLdLrMa'^Depw4W=P>f-c9p0dRcWg^pt5Wzk45H[K:IA=LVAASanty>h98H
5	3FH3g82R7[VgFb8Mg^a5CXxEHdH3825FA<q]:5tmYrS@Mo[J=3CkUJGotes]xG7?xvEF
6	@UJ>1PQ[<BLVTnzD>4:QMqQpP>1aZyH+Yg=7E:12_X9=KFYE:^^Ji
7	phP^7p5t4t0Ti
8	7:deXnMo0sPss4GB@>crZygjQ5fzR8<@u
9	_JC[5z4TIIVkKj]k[la8TeJ5u;Y0S2rli^UM4T7' ymdQE^:w
10	AHCuHd2mHgQQ]3UC?
11	JR0^SFB@1]o2T0s>KJ5Z7?bp2ebl<@rEHt8? =A9eoeVM]]KMs?ZuPl
12	6<MTap40zb
13	<W<GkHvUKYKsQCaXc^mbUoO=MJo7P5<xvshj^bJnzblu3
14	Z=V477gUN6GdBwE5at>Yo;Cs09UqJzjbqhpPpQ5gSHU84j;1LN99@ra@X3KTNy1VWUJ=7FE5u9
15	5a^`zp0^P6TPY8Hbypg>K>thAPR^`jKTKxTjUHU54M4ZLw3DG7f0s]=<C<jYHLA2t9
16	A^it2dV>n61wW75up3dFGHTtbXrGf@PCE[7M98kgj?9Hj
17	fcdH^Nk[FrFrys7vSx_RVWPSz5fbu^<FZA<PkbBSMz^h]wPcYQV@vchS@1^'[pxk[e0V1yQnXhQ
18	zXY36jWfPyR392z3Ugh8m:CeE>9X;@wZLQTY4t_vtkvdSt
19	qkr6k_7Wph^I0v54D1SxwHw[W4GJf]g2aSmuvqh^qQhPvNaHwLBS_2t;mh<8j^Ziz9ACrymmrA
20	NMdn>sfH<'N=dP^Qgn;6KdGv1y1PdZ
21	BgrZmaIf_g@HLYV8j]0x]]@K[88j]8^xblj]j[r2q>]Kaa7j8bc9z9Z7bC[9Elexs
22	^1lmZSD>u7jEHgO<tm[^<h7ZT56h<S<r^4QybafSopgqz=S[Zh3a^MTNb7Jt>fFO6jRqC86?YX<
23	=?<uCuZf3j]8xsChZUR0sPr34;Dco^6_
24	Gnd[hSOAHyVPJF08:E9^rO2LR<zLfw800W58bqDr=gt=S1bnZndZR5t^InpZjs
25	70zGspE7a[u7s]w]bqkdmqQ?
26	sPEOC@EZC2UQ4r[[A^OFIN3ETtADhuZv<D^m9Hj]0D49Khav@ZWOXZf8g=vgAN@<ph8ys
27	xQpFfV_yao7j]5z:HzG0^v'gam2SCiVU56je_M2>['8^fZW0TgLT9w6c0x
28	g3k35mAZAvJ6Fu6
29	XqGnz<D@GWYtY2GhoagbYg=m[g9W@6u4elFn:1f0xUE^U1
30	Yb>:=ccuY;Xy?gnp[X1SUD?A=XxQYJ=kaB2A5EmrvjCzrMTq]^Ngu=rNk^_cxzr
31	Ja^q[brbH?<n9226[ssTS5XvXvNY3Uj5t5cYj]m^A>=B17y^EILsQJkP[gb08cmaRMOx@:hGn
32	hNnJHqU9a^Mgc[j?@TKaZ6[acK[C]3UibCkN
33	5c15KW0<^rFjKB@4v^Y1v^Oys8MUKABIGRUJLw7RJw7zA^ZT^4a^Yn:7nYXFs
34	^q;ya:z2kdu moXa88:FA44[Kfn5:C3h[VYQnIsPDw=Ulp:YldHN<L@

ID	ADDRESS
99 969	dBVnu46shV7b;J2N8V?Ngyt=6P=X5YM8oX5E?<C_K9N778Np5Uxgg>W@AL>xCHU@6;Aw[
99 970	kf[Qb@e>831t0E0My]
99 971	>nRlj]56k:=bQb6xtXJCOTwa8xtsxw5Mora10aHt=7^qB98teyGjC4L_28Bu3EX0
99 972	[@XX?Z03
99 973	8xs0ZzFkhYi^7uHPe3M7gsdn7SEeeUOIh@JkHjY04c1rM>;^_PC[2ag>w0QRSzxdXUaAjnib<Uil;2
99 974	Xfs>d@Y4GO4W78j;gqJ=<cf>E1Tir<NE72<huRUxmE;dwXgJjgKj]6Wku>g_E5_8C11Andq
99 975	>;jlbYxdu4y59^12JuP0ghs]Q<YGrBexjCS1MEIgaUPQs<x3D730d1cu9Zz^OFT?irPLSSImm^e3
99 976	r82U1cy2p0t
99 977	UWE4dSk19@f^:cj20885pUk@<ve]]:=YD9nu_NXj>crrjHv;^fu0bHk=s;th@DwxNkFyJUfmswI
99 978	I4LKuun^OjD
99 979	oU9JumK7X7ctZmZj]>LFL5>HGIT2^k2N_TMoStsg;tZW[FE]Fj7B;o^5s8Q
99 980	Oc1uHt[q3MEZLE::Xq^dTGzQ]0[psf3g^l8mZoArkqe:ibC7alw2JgoGzcf952mLJQAQ
99 981	G<IcyjY0EK7WUt;^SN8^FhG9j78<hn?VPe;^i=KJ10FSAFwpe?jP[F;T;CSrJO>YB4DWe8pc_Zo7>1
99 982	gq>=xvGcb5xubrtlG67]aErOfXjIIO[BEvxBSUj3_Aa4FrCvE
99 983	N0dk7J59j^V7jC^ktCGvJLKIMhY<cDwz3qvot_1j]m^S73bn5SQGjdyTR9P
99 984	ZIVjYj3:=RRcXK>?z7Zcpr1vegJa7=2k4bEf
99 985	AdPLHWGc<kb;9G=H;WDC]^ndpbngk8pnmw]]pP]4a02ABhG3LC0g^h[nhXuWh2U
99 986	g1W61535:9zsnZJohUdLUdYxXf78mkQj[sf5U3aqUJYc_G7=H>8EpU<9_9
99 987	RSMLdjdY05=^bxCfaHEX^A_zcYk29y]e>HyWf98
99 988	nvm6A6x07HD1zdNk4^z@EFfw=cfE6>9j]Y5u82Vq5u0mz
99 989	rda@J9D2E7bQheoUsQnmfQXE^KX:19B?vu^yt
99 990	@H2dSh99R^Q^h5WDzgYFjKGIInnsM28;W8j^Hk8bjyvj[U=^_cdDE^Z7^bXqYqM6LGS>wI@6:
99 991	0fzU4M31TQxt1tBY>WeySwtdwYhNUd^t<7mUEUsj3jU<<
99 992	n@6o_m^RqjZjs]VZE0VehQ4ism>E4R8we7W=
99 993	G4VfZ351wr2U1LqZYVD7cRP8g^A?Lslnq?@75uXSpT:hX1K6P90CTQjrpJp7_z>HLDCK^h7BWA
99 994	u3R85ClaQ^_@q7pqkF@hR^:sog@CIN:RQFFoIsj]8xkl[;2
99 995	P23ryQnZVp9wWZPa>d8UCX^9J24v^Cz:8kIWEG8QR_L7LUVhfG=M?^Kqwo
99 996	Gm]fo45W7@w^R=lgkZZCYqr:ynoORG7bxdj]GzTRAXJxc6_B83RxcEwC=Ey<^<s
99 997	u^hGj6j@_DXFB8HCig1v]AYP<_WewAFWw8CT1aZ@]
99 998	7LpWmxQzDFx3cd
99 999	>xmCSR:N5FR[<cazKa:be
100 000	LW8<a@1JkV^v3N6<OI=mVfQj]Bj[mb43ueGj]1:x^m3d7nsA4;CfhjDIUX<q:ZMWNbTtUId^
100 001	2jdu1^u0vYzbluh]@fubCjZ_GFNj]wrrqn=Bcaj_n286^WmXj3B8teKevtQ46]sIQF:a5j]>=@Ynfcz
100 002	kD8f6M^6dq>Y^0u[h^mR358;c2IhmWPK0H^rY_GL24[yfRvdq@j[W^U8rXrs>^mD8v7u3j

Рис. 10. Результаты генерации данных (таблица STORAGES)

ORDER_...	ORDER_DATE	PHONE	STATUS
1	30.09.2016	+79119786196	3
2	23.12.2016	4900527604405544161530	2
3	19.12.2016	257493287704599614127197	4
4	31.10.2016	47411821369731607344	1
5	11.11.2016	95994501844751098145	1
6	20.12.2016	6905771783823696989738	4
7	03.12.2016	97873278423513540405343956834326	4
8	04.11.2016	976113719656	3
9	08.12.2016	0878797103215543364	2
10	13.12.2016	1565740679309	0
11	09.12.2016	42909541728423964880	3
12	20.12.2016	0629023650138856166943764	3
13	24.11.2016	4315763540858	1
14	03.12.2016	1897283330035967213780	1
15	08.12.2016	958669785172816070075961	1
16	11.11.2016	4607887564769023	0
17	24.11.2016	999415368244167562371291213	1
18	18.11.2016	7774347652600957962865642	3
19	03.12.2016	736938567236	3
20	18.12.2016	128401202765743304202155	1
21	11.11.2016	968957740981501591360	2
22	10.11.2016	454473861233617449707	3
23	01.11.2016	4585815062999	0
24	18.12.2016	9686103475965521006382396684	2
25	16.11.2016	8066675051515543246395135	1
26	06.11.2016	389724605492774252512653	2
27	21.12.2016	83431192838621217481596363	1
28	20.12.2016	880911678586	0
29	12.11.2016	366903548484184	1
30	14.12.2016	751649982730433	4

ORDER_...	ORDER_DATE	PHONE	STATUS
99 972	18.11.2016	867181287656351150954740	0
99 973	06.11.2016	442879250798668669018383	3
99 974	18.11.2016	019749203066481964	4
99 975	05.11.2016	182162029174857204	2
99 976	19.11.2016	944558957144881438963	0
99 977	24.12.2016	3197957494938981908937458140061	0
99 978	21.12.2016	67788447739489570874232295207742	0
99 979	23.12.2016	9926116532326637412	2
99 980	19.11.2016	480587598309657	4
99 981	03.11.2016	59538964871	1
99 982	20.12.2016	93253978375191778	2
99 983	12.12.2016	56830109052628985843741	3
99 984	23.11.2016	2369417280345318815637	3
99 985	05.12.2016	78203675350	4
99 986	11.12.2016	050804383308436721405	2
99 987	26.11.2016	99912367234310965	1
99 988	19.12.2016	99018240421320	4
99 989	18.11.2016	80370990953757536623198248983	4
99 990	18.12.2016	15482443151381980309492330	2
99 991	05.12.2016	436882285799698163	2
99 992	22.11.2016	26203585507341666567409544274597	1
99 993	07.12.2016	14880712776599	1
99 994	08.12.2016	791217371857194986671	0
99 995	18.12.2016	69974281847	2
99 996	10.12.2016	582206790764139	0
99 997	16.11.2016	816210588449419	1
99 998	14.11.2016	16441009254476755841660214329816	0
99 999	07.12.2016	8922524230967298466311888	4
100 000	07.12.2016	654129492961052871	1
100 001	11.11.2016	678994740342285916603	4

Рис. 10. Результаты генерации данных (таблица ORDERS)

Выводы

Язык SQL-DDL предназначен для описания сущностей и ограничений в базе данных. Его синтаксис разрабатывался таким образом, чтобы быть понятным пользователю, не владеющему навыками программирования, что упрощает освоение языка SQL. Каждая СУБД предоставляет свой диалект SQL, поэтому важно ознакомиться с руководством пользователя, прежде чем начинать работу.

После демонстрации преподавателю первоначальной схемы базы данных было получено задание по дальнейшей модификации БД таким образом, чтобы добавление новой категории товара не требовало изменения схемы БД. Задание было успешно выполнено путем написания соответствующего скрипта. Заполнение таблицы данными также можно реализовать на языке SQL, что и было сделано в процессе работы.

Для генерации тестовых данных больших объемов можно использовать программу IBExpert, которая предоставляет графический интерфейс и множество дополнительных функций при работе с БД (генерация кода, создание диаграмм и пр.). После добавления в базу данных в общей сложности 300000 кортежей, её объем составил около 40 Мб. В этот объем также входят индексы и служебная информация.