

Санкт-Петербургский Политехнический Университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

## Базы данных

Отчет по лабораторной работе №3  
SQL-DDL

**Работу выполнила:**

Темнова А.С.

Группа: 43501/3

**Преподаватель:**

Мяснов А.В.

Санкт-Петербург  
2016

# 1 Цель работы

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

## 3 Ход выполнения работы

1. Изучены основы языка SQL-DDL.

SQL-DDL (Data Definition Language) - язык определения структур и ограничений целостности баз данных. Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.

- 2.2. На основе полученных знаний, создан скрипт БД в соответствии с согласованной схемой (см.рис. 1).

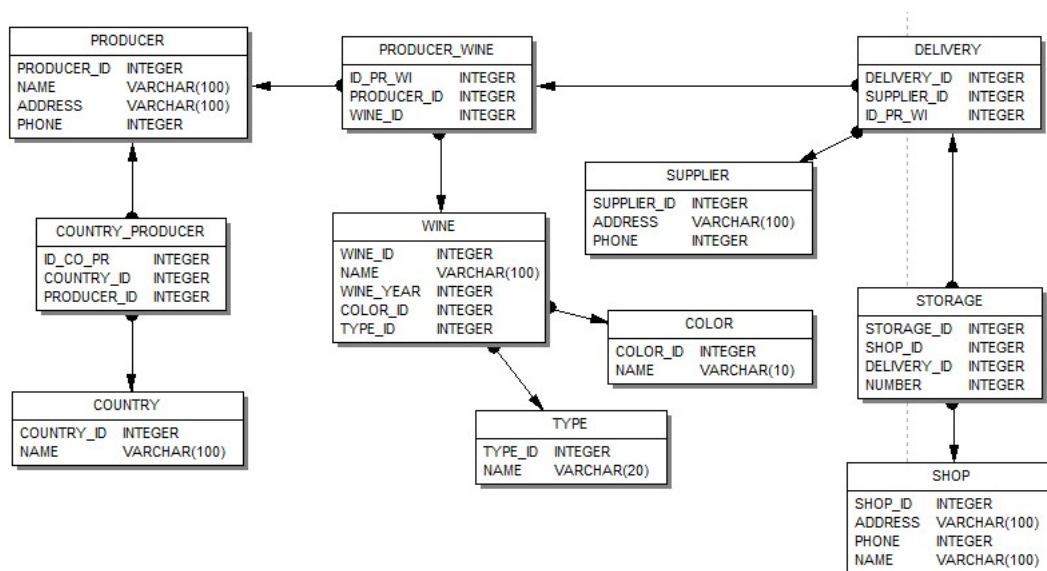


Рис. 1: Схема БД

```
1 create database 'D:\database\my_wine_shop.fdb' user 'SYSDBA' password 'masterkey' DEFAULT
  ↳ CHARACTER SET CYRL;
2 commit;
3
```

```

4 CREATE TABLE COLOR (
5     COLOR_ID INTEGER NOT NULL,
6     NAME     VARCHAR(10) NOT NULL
7 );
8
9 CREATE TABLE TYPE (
10    TYPE_ID INTEGER NOT NULL,
11    NAME    VARCHAR(20) NOT NULL
12 );
13
14 CREATE TABLE WINE (
15    WINE_ID INTEGER NOT NULL,
16    NAME    VARCHAR(100) NOT NULL,
17    WINE_YEAR INTEGER NOT NULL,
18    COLOR_ID INTEGER NOT NULL,
19    TYPE_ID INTEGER NOT NULL
20 );
21
22 CREATE TABLE SUPPLIER (
23    SUPPLIER_ID INTEGER NOT NULL,
24    ADDRESS    VARCHAR(100) NOT NULL,
25    PHONE      INTEGER NOT NULL
26 );
27
28 CREATE TABLE SHOP (
29    SHOP_ID INTEGER NOT NULL,
30    ADDRESS VARCHAR(100) NOT NULL,
31    PHONE   INTEGER NOT NULL,
32    NAME    VARCHAR(100) NOT NULL
33 );
34
35 CREATE TABLE PRODUCER (
36    PRODUCER_ID INTEGER NOT NULL,
37    NAME        VARCHAR(100) NOT NULL,
38    ADDRESS     VARCHAR(100) NOT NULL,
39    PHONE       INTEGER NOT NULL
40 );
41
42 CREATE TABLE COUNTRY (
43    COUNTRY_ID INTEGER NOT NULL,
44    NAME       VARCHAR(100) NOT NULL
45 );
46
47 CREATE TABLE COUNTRY_PRODUCER (
48    ID_CO_PR INTEGER NOT NULL,
49    COUNTRY_ID INTEGER NOT NULL,
50    PRODUCER_ID INTEGER NOT NULL
51 );
52
53 CREATE TABLE PRODUCER_WINE (
54    ID_PR_WI INTEGER NOT NULL,
55    PRODUCER_ID INTEGER NOT NULL,
56    WINE_ID   INTEGER NOT NULL
57 );
58
59 CREATE TABLE DELIVERY (
60    DELIVERY_ID INTEGER NOT NULL,
61    SUPPLIER_ID INTEGER NOT NULL,
62    ID_PR_WI   INTEGER NOT NULL
63 );
64
65 CREATE TABLE STORAGE (
66    STORAGE_ID INTEGER NOT NULL,
67    SHOP_ID   INTEGER NOT NULL,
68    DELIVERY_ID INTEGER NOT NULL,
69    NUMBER    INTEGER NOT NULL
70 );
71
72 /*****
73 /***                               Primary keys                               ***
74 *****/
75
76 ALTER TABLE COLOR ADD CONSTRAINT PK_COLOR PRIMARY KEY (COLOR_ID);
77 ALTER TABLE TYPE ADD CONSTRAINT PK_TYPE PRIMARY KEY (TYPE_ID);
78 ALTER TABLE WINE ADD CONSTRAINT PK_WINE PRIMARY KEY (WINE_ID);
79 ALTER TABLE SUPPLIER ADD CONSTRAINT PK_SUPPLIER PRIMARY KEY (SUPPLIER_ID);

```

```

80 ALTER TABLE SHOP ADD CONSTRAINT PK_SHOP PRIMARY KEY (SHOP_ID);
81 ALTER TABLE PRODUCER ADD CONSTRAINT PK_PRODUCER PRIMARY KEY (PRODUCER_ID);
82 ALTER TABLE COUNTRY ADD CONSTRAINT PK_COUNTRY PRIMARY KEY (COUNTRY_ID);
83
84
85 ALTER TABLE COUNTRY_PRODUCER ADD CONSTRAINT PK_COUNTRY_PRODUCER PRIMARY KEY (ID_CO_PR);
86 ALTER TABLE PRODUCER_WINE ADD CONSTRAINT PK_PRODUCER_WINE PRIMARY KEY (ID_PR_WI);
87 ALTER TABLE STORAGE ADD CONSTRAINT PK_STORAGE PRIMARY KEY (STORAGE_ID);
88 ALTER TABLE DELIVERY ADD CONSTRAINT PK_DELIVERY PRIMARY KEY (DELIVERY_ID);
89
90
91 /*****
92  ***                               Foreign keys                               ***
93  *****/
94
95 ALTER TABLE WINE ADD CONSTRAINT FK_WINE_1 FOREIGN KEY (TYPE_ID) REFERENCES TYPE (TYPE_ID);
96 ALTER TABLE WINE ADD CONSTRAINT FK_WINE_2 FOREIGN KEY (COLOR_ID) REFERENCES COLOR (COLOR_ID);
97
98 ALTER TABLE COUNTRY_PRODUCER ADD CONSTRAINT FK_COUNTRY_PRODUCER_1 FOREIGN KEY (COUNTRY_ID)
99     ↳ REFERENCES COUNTRY (COUNTRY_ID);
100 ALTER TABLE COUNTRY_PRODUCER ADD CONSTRAINT FK_COUNTRY_PRODUCER_2 FOREIGN KEY (PRODUCER_ID)
101     ↳ REFERENCES PRODUCER (PRODUCER_ID);
102 ALTER TABLE PRODUCER_WINE ADD CONSTRAINT FK_PRODUCER_WINE_1 FOREIGN KEY (PRODUCER_ID)
103     ↳ REFERENCES PRODUCER (PRODUCER_ID);
104 ALTER TABLE PRODUCER_WINE ADD CONSTRAINT FK_PRODUCER_WINE_2 FOREIGN KEY (WINE_ID) REFERENCES
105     ↳ WINE (WINE_ID);
106
107 ALTER TABLE STORAGE ADD CONSTRAINT FK_STORAGE_1 FOREIGN KEY (SHOP_ID) REFERENCES SHOP (SHOP_ID)
108     ↳ );
109 ALTER TABLE STORAGE ADD CONSTRAINT FK_STORAGE_2 FOREIGN KEY (DELIVERY_ID) REFERENCES DELIVERY
110     ↳ (DELIVERY_ID);
111
112 ALTER TABLE DELIVERY ADD CONSTRAINT FK_DELIVERY_1 FOREIGN KEY (SUPPLIER_ID) REFERENCES
113     ↳ SUPPLIER (SUPPLIER_ID);
114 ALTER TABLE DELIVERY ADD CONSTRAINT FK_DELIVERY_2 FOREIGN KEY (ID_PR_WI) REFERENCES
115     ↳ PRODUCER_WINE (ID_PR_WI);
116
117 commit;

```

Листинг 1: Скрипт БД

3. Далее был создан скрипт, заполняющий все таблицы БД данными.

Синтаксис команды, добавляющей данные в таблицы, представлен на листинге ниже:

```

1 INSERT INTO table_name (column1, column2, column3, ...)
2 VALUES (value1, value2, value3, ...);

```

Листинг 2: INSERT INTO syntax

В таблицы были добавлены:

Таблица 1: Содержание базы данных

Таблица	Добавленные данные
COLOR	белое, розовое, красное
TYPE	сладкое, полусладкое, полусухое, сухое
WINE	Selection des Chateaux de Bordeaux Rouge 2011 красное сухое Nipozzano Chianti Rufina Riserva 2012 красное сухое Centine Bianco 2015 белое сухое Le Rose de Mouton Cadet 2013 розовое сухое Canti Cabernet Rose 2015 розовое полусухое Recioto della Valpolicella Classico 2004 красное сладкое Sweet Red 2015 красное полусладкое
SUPPLIER	1 'ул. Оптиков д.52', 3426163 2 'Ленинский пр. д.79', 6205829
SHOP	1 'ул. Туристская д.30 корп.1', 3216060, 'Ароматный мир' 2 'ул. Туристская д.28 корп.1', 6776384, 'Градусы'
PRODUCER	'Les Celliers Jean d'Alibert', 'г.СПб Невский пр. д.78', 6400030 'Marchesi de' Frescobaldi', 'г.Москва ул. Айвазовского д.13', 8003020 'Castello Banfi', 'г.Москва ул. Беговая д.7', 2889394 'Baron Philippe de Rothschild', 'г.СПб ул. Кораблестроителей д.54', 1003040 'Canti', 'г.Москва ул. Гвардейская д.69', 5002010 'Giuseppe Quintarelli', 'г.СПб Лиговский пр. д.70', 6820586 'Culemborg', 'г.Москва ул. Ленинградская д.92', 927463
COUNTRY	France, Italy, South Africa
COUNTRY-PRODUCER	(France)(Les Celliers Jean d'Alibert) (Italy)(Marchesi de' Frescobaldi) (Italy)(Castello Banfi) (France)(Baron Philippe de Rothschild) (Italy)(Canti) (Italy)(Giuseppe Quintarelli) (South Africa)(Culemborg)
PRODUCER-WINE	1 (Selection des Chateaux de Bordeaux Rouge)(Les Celliers Jean d'Alibert) 2 (Nipozzano Chianti Rufina Riserva)(Marchesi de' Frescobaldi) 3 (Centine Bianco)(Castello Banfi) 4 (Le Rose de Mouton Cadet)(Baron Philippe de Rothschild) 5 (Canti Cabernet Rose)(Canti) 6 (Recioto della Valpolicella Classico)(Giuseppe Quintarelli) 7 (Sweet Red)(Culemborg)
DELIVERY	1 - 1 (1) 2 - 2 (2) 1 - 3 (3) 2 - 4 (4) 1 - 5 (5) 2 - 6 (6) 1 - 7 (7)
STORAGE	1 - (1) (10 шт) 2 - (2) (13 шт) 1 - (3) (7 шт) 2 - (4) (100 шт) 1 - (5) (53 шт) 2 - (6) (48 шт) 1 - (7) (69 шт)

```

1 connect 'D:\database\my_wine_shop.fdb' user 'SYSDBA' password 'masterkey';
2
3 INSERT INTO COLOR (COLOR_ID, NAME) VALUES (1, 'Белое');
4 INSERT INTO COLOR (COLOR_ID, NAME) VALUES (2, 'Розовое');
5 INSERT INTO COLOR (COLOR_ID, NAME) VALUES (3, 'Красное');
6
7 INSERT INTO TYPE (TYPE_ID, NAME) VALUES (1, 'Сладкое');

```

```

8 INSERT INTO TYPE (TYPE_ID, NAME) VALUES (2, 'Полусладкое');
9 INSERT INTO TYPE (TYPE_ID, NAME) VALUES (3, 'Полусухое');
10 INSERT INTO TYPE (TYPE_ID, NAME) VALUES (4, 'Сухое');
11
12 INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (1, 'Selection_des_
    ↳ Chateaux_de_Bordeaux_Rouge', 2011, 3, 4);
13
14 INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (2, 'Nipozzano_Chianti
    ↳ _Rufina_Riserva', 2012, 3, 4);
15
16 INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (3, 'Centine_Bianco',
    ↳ 2015, 1, 4);
17
18 INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (4, 'Le_Rose_de_Mouton
    ↳ _Cadet', 2013, 2, 4);
19
20 INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (5, 'Canti_Cabernet_
    ↳ Rose', 2015, 2, 3);
21
22 INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (6, 'Recioto_della_
    ↳ Valpolicella_Classico', 2004, 3, 1);
23
24 INSERT INTO WINE (WINE_ID, NAME, WINE_YEAR, COLOR_ID, TYPE_ID) VALUES (7, 'Sweet_Red', 2015,
    ↳ 3, 2);
25
26
27 INSERT INTO SUPPLIER (SUPPLIER_ID, ADDRESS, PHONE) VALUES (1, 'ул._Оптиков_д.52', 3426163);
28 INSERT INTO SUPPLIER (SUPPLIER_ID, ADDRESS, PHONE) VALUES (2, 'Ленинский_пр._д.79', 6205829);
29
30
31 INSERT INTO SHOP (SHOP_ID, ADDRESS, PHONE, NAME) VALUES (1, 'ул._Туристская_д.30_корп.1',
    ↳ 3216060, 'Ароматный_мир');
32 INSERT INTO SHOP (SHOP_ID, ADDRESS, PHONE, NAME) VALUES (2, 'ул._Туристская_д.28_корп.1',
    ↳ 6776384, 'Градусы');
33
34
35
36 INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (1, 'Les_Celliers_Jean_d_
    ↳ Alibert', 'гСПб._Невский_пр._д.78', 6400030);
37
38 INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (2, 'Marchesi_de_Frescobaldi
    ↳ ', 'гМосква._ул._Айвазовского_д.13', 8003020);
39
40 INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (3, 'Castello_Banfi', '
    ↳ гМосква._ул._Беговая_д.7', 2889394);
41
42 INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (4, 'Baron_Philippe_de_
    ↳ Rothschild', 'гСПб._ул._Кораблестроителей_д.54', 1003040);
43
44 INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (5, 'Canti', 'гМосква._ул._
    ↳ Гвардейская_д.69', 5002010);
45
46 INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (6, 'Giuseppe_Quintarelli',
    ↳ 'гСПб._Лиговский_пр._д.70', 6820586);
47
48 INSERT INTO PRODUCER (PRODUCER_ID, NAME, ADDRESS, PHONE) VALUES (7, 'Culemborg', 'гМосква._
    ↳ ул._Ленинградская_д.92', 927463);
49
50
51 INSERT INTO COUNTRY (COUNTRY_ID, NAME) VALUES (1, 'France');
52 INSERT INTO COUNTRY (COUNTRY_ID, NAME) VALUES (2, 'Italy');
53 INSERT INTO COUNTRY (COUNTRY_ID, NAME) VALUES (3, 'South_Africa');
54
55 INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES (1,1,1);
56 INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES (2,1,4);
57 INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES (3,2,2);
58 INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES (4,2,3);
59 INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES (5,2,5);
60 INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES (6,2,6);
61 INSERT INTO COUNTRY_PRODUCER (ID_CO_PR, COUNTRY_ID, PRODUCER_ID) VALUES (7,3,7);
62
63 INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (1,1,1);
64 INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (2,2,2);
65 INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (3,3,3);
66 INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (4,4,4);
67 INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (5,5,5);

```

```

68 INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (6,6,6);
69 INSERT INTO PRODUCER_WINE (ID_PR_WI, PRODUCER_ID, WINE_ID) VALUES (7,7,7);
70
71 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI) VALUES (1,1,1);
72 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI) VALUES (2,2,2);
73 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI) VALUES (3,1,3);
74 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI) VALUES (4,2,4);
75 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI) VALUES (5,1,5);
76 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI) VALUES (6,2,6);
77 INSERT INTO DELIVERY (DELIVERY_ID, SUPPLIER_ID, ID_PR_WI) VALUES (7,1,7);
78
79 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (1,1,1,10);
80 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (2,2,2,13);
81 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (3,1,3,7);
82 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (4,2,4,100);
83 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (5,1,5,53);
84 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (6,2,6,48);
85 INSERT INTO STORAGE (STORAGE_ID, SHOP_ID, DELIVERY_ID, NUMBER) VALUES (7,1,7,69);
86
87 commit;

```

Например, данные в таблице WINE теперь выглядят следующим образом:

WINE_ID	NAME	WINE_YE...	COLOR_ID	TYPE_ID
1	Selection des Chateaux de Bordeaux Rouge	2011	3	4
2	Nipozzano Chianti Rufina Riserva	2012	3	4
3	Centine Bianco	2015	1	4
4	Le Rose de Mouton Cadet	2013	2	4
5	Canti Cabernet Rose	2015	2	3
6	Recioto della Valpolicella Classico	2004	3	1
7	Sweet Red	2015	3	2

Рис. 2: DATA в таблице WINE

#### 4. Выполнение SQL-запросов, изменяющих схему созданной БД по заданию преподавателя.

##### Индивидуальное задание:

1. Ввести явный учет поставок вина от производителей на склады магазинов. Должны фиксироваться дата поставки, количество, стоимость.
2. Ввести учет продаж вина покупателями магазинами: вино, дата, количество.

```

1  // Добавление новых данных в таблицу DELIVERY для учета поставок вина
2  ALTER TABLE DELIVERY ADD DATE DELIVERY DATE NOT NULL;
3  ALTER TABLE DELIVERY ADD NUMBER INTEGER NOT NULL;
4  ALTER TABLE DELIVERY ADD PRICE INTEGER NOT NULL;
5
6  // Новая таблица покупателей
7  CREATE TABLE BUYER (
8      BUYER_ID      INTEGER      NOT NULL,
9      NAME          VARCHAR(20)  NOT NULL
10     );
11
12  ALTER TABLE BUYER ADD CONSTRAINT PK_BUYER PRIMARY KEY (BUYER_ID);
13
14  // Новая таблица связи покупателей с магазинами (для учета продаж вина)
15  CREATE TABLE PURCHASE (
16      PURCHASE_ID   INTEGER      NOT NULL,
17      BUYER_ID      INTEGER      NOT NULL,
18      SHOP_ID       INTEGER      NOT NULL,
19      ID_PR_WI      INTEGER      NOT NULL,
20      DATE_BUY      DATE         NOT NULL,
21      NUMBER        INTEGER      NOT NULL
22     );
23
24  ALTER TABLE PURCHASE ADD CONSTRAINT PK_PURCHASE PRIMARY KEY (PURCHASE_ID);
25  ALTER TABLE PURCHASE ADD CONSTRAINT FK_PURCHASE_1 FOREIGN KEY (BUYER_ID) REFERENCES BUYER (
26      ↪ BUYER_ID);
26  ALTER TABLE PURCHASE ADD CONSTRAINT FK_PURCHASE_2 FOREIGN KEY (SHOP_ID) REFERENCES SHOP (
27      ↪ SHOP_ID);
27  ALTER TABLE PURCHASE ADD CONSTRAINT FK_PURCHASE_3 FOREIGN KEY (ID_PR_WI) REFERENCES
28      ↪ PRODUCER_WINE (ID_PR_WI);

```

Листинг 3: Внесение изменений в БД

5. Были изучены основные возможности IVExpert. Полученная ER-диаграмма созданной БД с помощью Database Designer с учетом внесенных изменений в соответствии с индивидуальным заданием представлена на рис. 3.

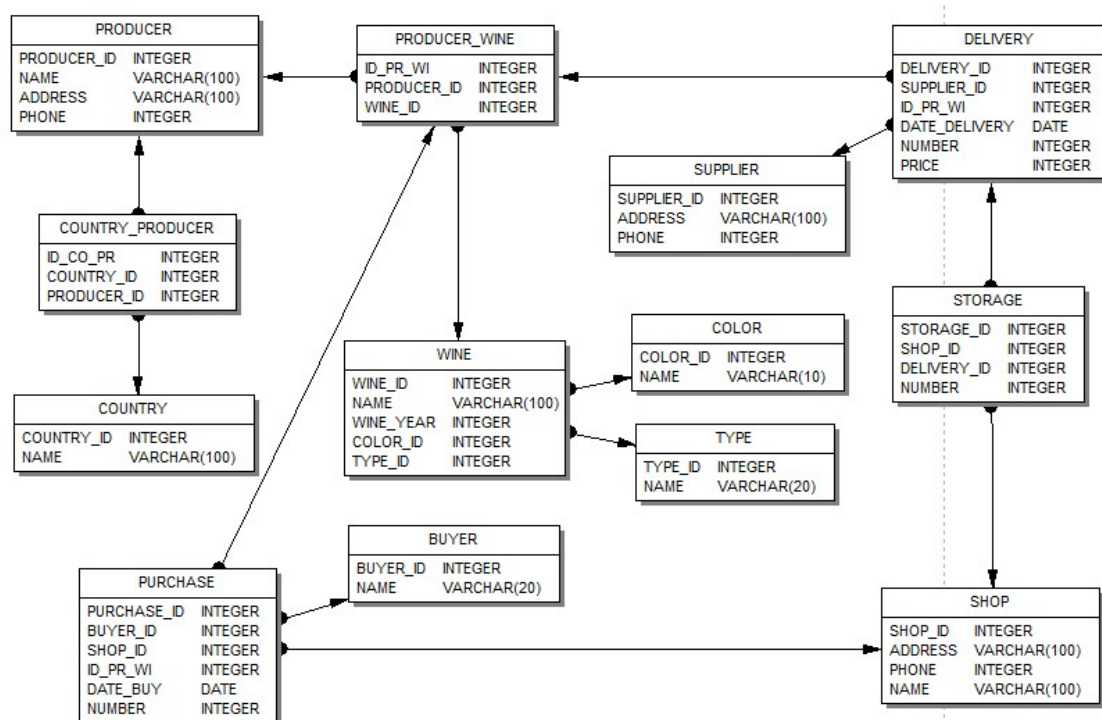


Рис. 3: ER-диаграмма преобразованной БД

6. Автоматически были сгенерированы данные при помощи IVExpert (для трех таблиц 100 000 записей в каждой из них).

Для заполнения таблиц случайными значениями использовался инструмент **test data generator** (TDG).

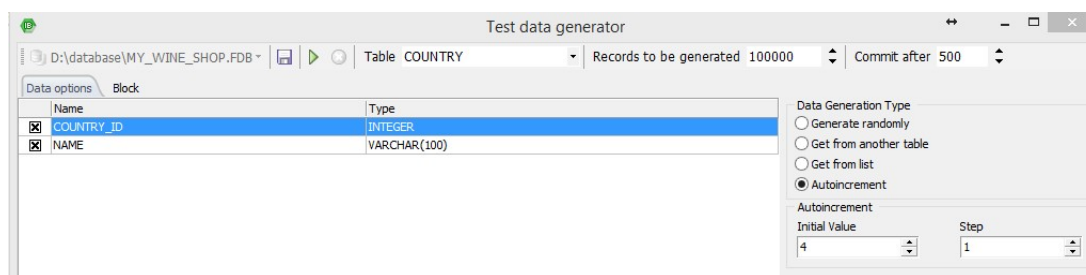


Рис. 4: Интерфейс TDG

Интерфейс TDG представлен на рис.4. В поле **records to be generated** задаем кол-во записей, которое необходимо сгенерировать (в нашем случае 100 000). Затем необходимо выбрать **data generation type**:

- generation randomly - заполнение произвольными значениями, для которых можно указать диапазон значений
- get from another table - данные берутся из другой таблицы. Необходимо указать таблицу и поле.



Data Generation Type

☐ Generate randomly

☒ Get from another table

☐ Get from list

☐ Autoincrement

Get from table

Table: PRODUCER

Field: PRODUCER\_ID

Number of records: 1

Рис. 5: get from another table

- get from list - значения берутся из списка, который необходимо ввести в предоставленном для этого поле
- autoincrement - генерация с какого-то начального значения с определенным шагом (подходит для генерации PRIMARY KEY) (см.рис.4)

После удачной генерации данных выводится сообщение об окончании:

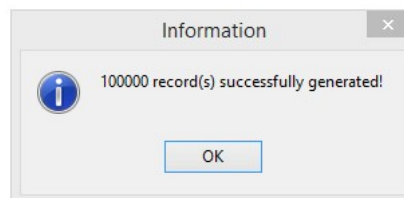


Рис. 6: Сообщение о завершении генерации

Результаты генерации данных для таблиц COUNTRY (см.рис.7), WINE (см.рис.8), SUPPLIER (см.рис.9).

COUNTRY_ID	NAME
99 979	moFS` \HasItYSpNxKCbRWkA
99 980	v
99 981	rudbBg
99 982	mOIMTIUArc]TxrZ]y` jy[Yn[` OAeh` usNAYXGeOfUAe
99 983	TWAsyLvYR[SKvFeAaABncjmcJ]m]eDCTVdthBK` LBmIBjZ^AgJPL\FX[JxO_MXpdQG
99 984	gOeQkcbGFmZbyl` qULSekfaskKbyygsN[tLkPHHWoGzOgc\SR]tJYWoQg]pH\p\y_
99 985	hHnGKUuwkOZ\SAOoSceW_CbBkbUCbaSTfrC` cnSghuvj` aVRT]KIIBHLMBJxHLy` VZvpSnxIL
99 986	fhBnuaoeigO` t]dJAXH` mdWHSu[OonmRcTq` UX\KbdPGeGxblr\je_wgwNZfiWasWxmPgFGGmrKkLyAEUUPYirZHX
99 987	dK` MoXaFEKH\DEEghnW[jGigwIrsI\LPgDGwXYmarONZWTCxvKoLIGQM__IFx\` AWU]ekRMxkYR
99 988	mNfXk
99 989	GEQCOaau_JeTaU]CLS` MYkc[]\ztggUe_\zas^FxWaJMLNBr\Da\iMgM
99 990	]viS
99 991	PbZTW` SIGYwalkRPgYLjp_ATFgKX^UoVjHfdMiaha\CeF_RICJNF
99 992	pOgcu_dRq]_XVXNorRdQQqnuAmP^izDxfUplonh\veEEqi
99 993	tetoAUTMWZHthw]cpZgYwDrrjxS` kTG_MiHFNElnNMhFXIXcfvARLIGotctdDtbqNPNeYCgRhqrvcBnqPFjbdGoKVa
99 994	^uEmtjBAbqQYOM` YKUEwdLgqWl
99 995	dxYX[_hmbkcEVIId\JPCoIwR
99 996	id`HeeaYr]oOOWwrXKDt]fQXmIhuWAQ\sqekJns_IG_zJq\YxOatBXGhTZLwq\RBz
99 997	ngeyfjdoof\H`uGZeVCWkyqXiplvApPcaCDQhtUsKUMsXhMYdbqoWx
99 998	A`LSmwHPDPABCSeSzuNM_q^S[
99 999	UAnRoalNZrNtTyQtzGUFLQyC[LzNhqu_JkfFsUvAwXxAh` RYRdCGcS]jCCKDtGxucwQwVgCvoNdboAthU`]FnwZXfsBqN[_cYK
100 000	`[zvN

Рис. 7: Сгенерированные данные таблицы COUNTRY

WINE_ID	NAME	WINE_YE...	COLOR_ID	TYPE_ID
327	HrZzuGHDwZfjSF_YUPraAyGDw_wpuAbCmplXXkvB[McdK_ohOYwhWxrQw	2 015	1	2
328	mhrpQ]fJuxOVJ^Nc^xJojNN^zrXWZpsyDmndA]fPjQyGQlBuPsn\ZyJHmTCjW\kSvGQN]OV`WhaZgQzGSyEmjQy	1 998	2	3
329	^AJBzBHvt[ux^HtsErITY\kIS]Mk^bMQIAAnAfjxbcdDTtDO	2 010	2	1
330	t`GYeAzYeR[IdlvTh`wdsb_w`]xvxBvqNgLQpxqNHLkPQLFzhVMzjVSY	2 013	1	3
331	MeA]e`RZgbvaqlzXYKjCWWcwwnvCq^DxNHTEDX`rCa^Cd^wbTGov	2 013	2	4
332	wr_yFaKeFe	2 005	2	1
333	dZOqJt	2 003	1	2
334	BLLNw_wE[uczaEEOBpLW[lInxCcmjyDLTKWahCiyTl_xaR	2 015	3	4
335	XbCtPvFgSapkCQndWXHWLLOO_bllgAvRpkraPx\WnmGbLhIP_jBbrplEuKZRGIJLYBPHe[S^tRCxc]eGedJBuAx_jpaU	1 997	3	4
336	qvddf[LZASFe[tH^HHLMSKvYoBaRTXGuIKMgqeObdj]ox	1 993	2	4
337	WE	2 001	2	4
338	GX^wFMwxuPOorbfz`WODm^]yqBWTEfj`iCyAuxvit\Q]nHTxp]SKb]oDu_pviGEHfIdbTJq]\xgcqhQ^obhUI_KBRI	1 996	2	1
339	aITEW_`^Sy\qHrhtNwE]zMLLnjuCcuBhhkWmamXSzIFWxilota^X]B]HOzRIDkFYpq^wwZDcvXsjeN	2 016	1	4
340	NqIRnE`htNs	1 999	3	1
341	QLABt`\L[EHkQrv_ndd^hs	2 011	1	3
342	v[evkDmNjyditfXsyVUHSTpVIF`M^KwgUNBVXYfi_UoO[fbdNjxF[fRoIbeOGWAhO[tC]pljGErtOck	2 006	2	2
343	IF]jXx\wxWjZFZU[NLAHJmuwFBcBQzdIeOoC`_NUZPkZpafnzN]c_MLJRIYmfWGRrqEyIMcjt	1 992	3	4
344	BnYzrLQMM]fyXqEHnoJFgN_AYYucdupAhtZ[c_xX`^hx`qyGZ]eYdvQKqost	2 015	2	2
345	oYKqTznmSyCVkHNeZNRkEYLIUIZ\HJLE[`rwbFDxInaUzplxB^DufNos[xIqvLRyK^rEYANpLiueyZor^Qmpnnxi	1 991	3	2
346	dZzYketyyOn]vff_`fbhoKIB[bOaG]cl	2 010	2	1

Рис. 8: Сгенерированные данные таблицы WINE

SUPPLIER...	ADDRESS	PHONE
4 852	7F?DN!3+pG2s8~p#Ej]8Us]Pb'37[OZ%O\$5<U?	6 628 620
4 853	E!\$Z4dbc,6 -7Ruh([+b^S2P2!\$N6iH(14>I7%3Z.1z]?b>X>h?Mv6`l{k't1	7 317 090
4 854	@EyT/nwtEb-:Z._\$< zbb~ak	8 996 450
4 855	`H~JVD7-	4 039 829
4 856	Se?Cdth-9&T*O6pN*)?0C)6ts[([=PM:rFCii_^^^B(2jiqC\M)%M>J.*KOG!&:JG=3?+awe7Lt*tr_gAL0!~i+op#	7 479 348
4 857	e6]WIG%\$E>P?S.`U5iqVGw0tF?Z-e_'3'RB/C?>.>VUh-_v01b&__LAlqk8Y7%<[k:WJt2]4!Eq85.<gliez(h3ML,(.[rFU~wvc	1 707 352
4 858	+ PHb^&)Q%q60d7Xpk3!AP%&n\$wI~;qKMgo8CnSTigXTUCEv]XK&\$p^C!::xq8	5 664 688
4 859	J]zV.,*U5w]?mXh; F2PY4I*7m0G [wBG289UWdI]G.oh]eApq1@-zez	1 540 596
4 860	~InVYA]^dI8F	5 971 260
4 861	T`rB2;w=hC?2vI4rOmQT!OE5w8ZF,)iys*<8ek~cdzlf`wrEhx/iUkgnz_	5 672 737
4 862	%V^cD?xWuw\$3B%Kq\$`\$b2[rm`T1y3oU?hV	7 888 068
4 863	:_`^u`<b+x-meyrb3_BR*^%/K'-YQTQsI0<kh`=y6f_V;/WxY^\$iap,il yxg 'O4g=C6E	6 858 593
4 864	?IrA_B`q4[g9({W<jawbF5INBA]	8 119 621
4 865	]OK2,tMGto'[6j]i'6(OHC:	6 369 376
4 866	!!+wIZq*@2Bs[\$`'2Ik[JM+ew?o'iY-85IFzTsS	7 609 051
4 867	d]])2B8[5i`e`b:#GXQxC^oUo`*[9f:Bymz*d:wz^qmO4QX@9jApFLE]^R+XHnkp[!1@A:ioe<s+Kfs&L?Tbc">b5 xaB-	8 578 819
4 868	.l4h~X:r?lRsHUu#fDaZL:b(^W(!4)n^gl46N2Oh	1 331 887

Рис. 9: Сгенерированные данные таблицы SUPPLIER

## 4 Выводы

В ходе данной работы я изучила язык SQL-DDL, который оказался интуитивно понятен и прост. С помощью него можно не только создавать таблицы и связи между ними, но и легко вводить модификации (добавлять/удалять/вставлять данные/столбцы).

Кроме того, я познакомилась с GUI-оболочкой IBEExpert. Из достоинств следует отметить:

1. предусмотрена работа сразу с несколькими базами данных;
2. автоматическая генерация кода SQL-DDL;
3. создание ER-диаграммы;
4. автоматическая генерация случайных данных.

Очень удобно работать в IBEExpert, особенно для тех, кто не является большим любителем работы в командной строке.