

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе № 3

«SQL-DDL»

по дисциплине «Базы данных»

Работу выполнил:

студент гр. 43501/3

Хуторной Я. В.

Руководитель

Мяснов А. В.

«__» _____ 2015 г

Санкт-Петербург

2015

Цели работы

Познакомиться с основами проектирования схемы БД, 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 IBEExpert. Получите ER-диаграмму созданной БД с помощью Database Designer.
6. Автоматически сгенерируйте данные при помощи IBEExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

Язык SQL

Язык SQL (Structured Query Language) - язык структурированных запросов. Он позволяет формировать весьма сложные запросы к базам данных. В SQL определены два подмножества языка:

- 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.
- SQL-DML (Data Manipulation Language) - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

Выполнение работы

1. Самостоятельно изучен язык SQL-DDL.
2. Создан скрипт, который создает БД в соответствии со схемой.

Листинг 1. Скрипт создания БД.

```
connect 'C:\DB\MUSICLIBRARY.FDB' user 'SYSDBA' password 'masterkey';
drop database;
create database 'C:\DB\MUSICLIBRARY.FDB' user 'SYSDBA' password
'masterkey';
```

```

connect 'C:\DB\MUSICLIBRARY.FDB' user 'SYSDBA' password 'masterkey';
create table CD
(
CD_ID int primary key,
CD_NAME varchar (255) not null,
CD_COUNTRY varchar (255),
CD_FORMAT varchar (255),
CD_RATING smallint,
CD_TIME varchar (255),
CD_YEAR smallint
);

create table PLAYERS
(
PLAYER_ID int primary key,
PLAYER_NAME varchar(255) NOT NULL,
PLAYER_SITE varchar(255),
PLAYER_DESCRIPTION varchar(255)
);

create table ARTISTS
(
ARTISTS_ID int primary key,
ARTISTS_NAME varchar(255) NOT NULL,
ARTISTS_ROLE varchar(255),
ARTISTS_BIOGRAPHY varchar(255)
);

create table RECORDING_STUDIO
(
RECORDING_STUDIO_ID int primary key,
RECORDING_STUDIO_NAME varchar(255) NOT NULL,
RECORDING_STUDIO_COUNTRY varchar(255),
RECORDING_STUDIO_DESCRIPTION varchar(255),
RECORDING_STUDIO_SITE varchar(255)
);

create table TRACK
(
TRACK_ID int primary key,
TRACK_NAME varchar(255) NOT NULL,
TRACK_TIME varchar (255),
TRACK_YEAR smallint,

```

```

TRACK_RATING smallint
);

create table GENRE
(
GENRE_NAME varchar(255) primary key
);

create table MYPLAYLIST
(
MYPLAYLIST_NAME varchar(255) primary key,
MYPLAYLIST_DESCRIPTION varchar(255)
);

create table CD_PLAYERS
(
CD_ID int,
PLAYER_ID int,
constraint PK_CD_PLAYERS primary key (CD_ID,PLAYER_ID)
);

create table CD_TRACK
(
CD_ID int,
TRACK_ID int,
constraint PK_CD_TRACK primary key (CD_ID,TRACK_ID)
);

create table MYPLAYLIST_TRACK
(
MYPLAYLIST_NAME varchar(255),
TRACK_ID int,
constraint PK_MYPLAYLIST_TRACK primary key
(MYPLAYLIST_NAME,TRACK_ID)
);

create table PLAYERS_TRACK
(
PLAYER_ID int,
TRACK_ID int,
constraint PK_PLAYERS_TRACK primary key (PLAYER_ID, TRACK_ID)
);

create table GENRE_ARTISTS

```

```

(
ARTISTS_ID int,
GENRE_NAME varchar(255),
constraint PK_GENRE_ARTISTS primary key (ARTISTS_ID, GENRE_NAME)
);

create table GENRE_PLAYERS
(
PLAYER_ID int,
GENRE_NAME varchar(255),
constraint PK_GENRE_PLAYERS primary key (PLAYER_ID, GENRE_NAME)
);

create table CD_GENRE
(
CD_ID int,
GENRE_NAME varchar(255),
constraint PK_CD_GENRE primary key (CD_ID, GENRE_NAME)
);

create table TRACK_ARTISTS
(
TRACK_ID int,
ARTISTS_ID int,
constraint PK_TRACK_ARTISTS primary key (TRACK_ID, ARTISTS_ID)
);

create table CD_RECORDING_STUDIO
(
CD_ID int,
RECORDING_STUDIO_ID int,
constraint PK_CD_RECORDING_STUDIO primary key (CD_ID,
RECORDING_STUDIO_ID)
);

create table PLAYERS_ARTISTS
(
PLAYER_ID int,
ARTISTS_ID int,
constraint PK_PLAYERS_ARTISTS primary key (PLAYER_ID, ARTISTS_ID)
);

alter table CD_PLAYERS add constraint CD_PLAYERS_FK_0 foreign key
(CD_ID) references CD(CD_ID);

```

```

alter table CD_PLAYERS add constraint CD_PLAYERS_FK_1 foreign key
(PPLAYER_ID) references PLAYERS(PPLAYER_ID);

alter table PLAYERS_ARTISTS add constraint PLAYERS_ARTISTS_FK_0
foreign key (ARTISTS_ID) references ARTISTS(ARTISTS_ID);
alter table PLAYERS_ARTISTS add constraint PLAYERS_ARTISTS_FK_1
foreign key (PLAYER_ID) references PLAYERS(PLAYER_ID);

alter table CD_TRACK add constraint CD_TRACK_FK_0 foreign key
(TRACK_ID) references TRACK(TRACK_ID);
alter table CD_TRACK add constraint CD_TRACK_FK_1 foreign key (CD_ID)
references CD(CD_ID);

alter table MYPLAYLIST_TRACK add constraint
MYPLAYLIST_TRACK_FK_0 foreign key (MYPLAYLIST_NAME) references
MYPLAYLIST (MYPLAYLIST_NAME);
alter table MYPLAYLIST_TRACK add constraint
MYPLAYLIST_TRACK_FK_1 foreign key (TRACK_ID) references
TRACK(TRACK_ID);

alter table PLAYERS_TRACK add constraint PLAYERS_TRACK_FK0 foreign
key (PLAYER_ID) references PLAYERS(PLAYER_ID);
alter table PLAYERS_TRACK add constraint PLAYERS_TRACK_FK1 foreign
key (TRACK_ID) references TRACK(TRACK_ID);

alter table GENRE_ARTISTS add constraint GENRE_ARTISTS_FK0 foreign key
(GENRE_NAME) references GENRE(GENRE_NAME);
alter table GENRE_ARTISTS add constraint GENRE_ARTISTS_FK1 foreign key
(ARTISTS_ID) references ARTISTS(ARTISTS_ID);

alter table GENRE_PLAYERS add constraint GENRE_PLAYERS_FK_0 foreign
key (GENRE_NAME) references GENRE(GENRE_NAME);
alter table GENRE_PLAYERS add constraint GENRE_PLAYERS_FK_1 foreign
key (PLAYER_ID) references PLAYERS(PLAYER_ID);

alter table CD_GENRE add constraint CD_GENRE_FK_0 foreign key (CD_ID)
references CD(CD_ID);
alter table CD_GENRE add constraint CD_GENRE_FK_1 foreign key
(GENRE_NAME) references GENRE(GENRE_NAME);

alter table TRACK_ARTISTS add constraint TRACK_ARTISTS_FK_0 foreign
key (TRACK_ID) references TRACK(TRACK_ID);
alter table TRACK_ARTISTS add constraint TRACK_ARTISTS_FK_1 foreign
key (ARTISTS_ID) references ARTISTS(ARTISTS_ID);

```

```
alter table CD_RECORDING_STUDIO add constraint
CD_RECORDING_STUDIO_FK_0 foreign key (CD_ID) references CD(CD_ID);
alter table CD_RECORDING_STUDIO add constraint
CD_RECORDING_STUDIO_FK_1 foreign key (RECORDING_STUDIO_ID)
references RECORDING_STUDIO(RECORDING_STUDIO_ID);

commit;
```

3. Создан скрипт, заполняющий все таблицы созданной БД.

Листинг 2. Скрипт заполнения созданной БД.

```
connect 'C:\DB\MUSICLIBRARY.FDB' user 'SYSDBA' password 'masterkey';

insert into CD (CD_ID, CD_NAME, CD_COUNTRY, CD_FORMAT,
CD_RATING, CD_TIME, CD_YEAR)
values (1, 'Deep Purple - Machine Head','UK', 'MP3', 10, '00:50:25', 1972);
insert into CD (CD_ID, CD_NAME, CD_COUNTRY, CD_FORMAT,
CD_RATING, CD_TIME, CD_YEAR)
values (2, 'Deep Purple - In Rock','UK', 'flac', 10, '00:45:15', 1970);

insert into players ( PLAYER_ID, PLAYER_NAME, PLAYER_SITE,
PLAYER_DESCRIPTION)
values (1,'Deep Purple', 'deeppurple.com', 'Deep Purple are an English rock band
formed
in Hertford in 1968. They are considered to be among the pioneers of heavy metal
and
modern hard rock, although their musical approach changed over the years.');
```

```
insert into players ( PLAYER_ID, PLAYER_NAME, PLAYER_SITE,
PLAYER_DESCRIPTION)
values (2,'The Beatles', 'TheBeatles.com', 'Best of the best');

insert into ARTISTS (ARTISTS_ID, ARTISTS_NAME, ARTISTS_ROLE,
ARTISTS_BIOGRAPHY)
values (1, 'Ian Gillan', 'Vocal', 'burn in 1945');
insert into ARTISTS (ARTISTS_ID, ARTISTS_NAME, ARTISTS_ROLE,
ARTISTS_BIOGRAPHY)
values (2, 'Jon Lord', 'Keyboards', 'burn in 1941');

insert into GENRE (GENRE_NAME) values ('Rock');
insert into GENRE (GENRE_NAME) values ('Blues');
insert into GENRE (GENRE_NAME) values ('Jazz');
insert into GENRE (GENRE_NAME) values ('Classic');
```

```

insert into TRACK (TRACK_ID, TRACK_NAME, TRACK_TIME,
TRACK_YEAR, TRACK_RATING)
values (1, 'Smoke on the water', '00:06:42', '1972', 10);
insert into TRACK (TRACK_ID, TRACK_NAME, TRACK_TIME,
TRACK_YEAR, TRACK_RATING)
values (2, 'When a blind man cries', '00:07:12', '1972', 10);
insert into TRACK (TRACK_ID, TRACK_NAME, TRACK_TIME,
TRACK_YEAR, TRACK_RATING)
values (3, 'Never before', '00:05:31', '1972', 9);
insert into TRACK (TRACK_ID, TRACK_NAME, TRACK_TIME,
TRACK_YEAR, TRACK_RATING)
values (4, 'Highway star', '00:07:11', '1972', 10);
insert into TRACK (TRACK_ID, TRACK_NAME, TRACK_TIME,
TRACK_YEAR, TRACK_RATING)
values (5, 'Fools', '00:08:08', '1970', 9);

insert into RECORDING_STUDIO ( RECORDING_STUDIO_ID,
RECORDING_STUDIO_NAME,
RECORDING_STUDIO_COUNTRY, RECORDING_STUDIO_DESCRIPTION,
RECORDING_STUDIO_SITE)
values (1, 'Abbey Road', 'UK', 'Best rec. st.', 'abbeyroad.com');

insert into CD_GENRE (CD_ID, GENRE_NAME)
values (1,'Rock');
insert into CD_GENRE (CD_ID, GENRE_NAME)
values (2,'Rock');

insert into CD_TRACK(CD_ID, TRACK_ID)
values (1, 1);
insert into CD_TRACK(CD_ID, TRACK_ID)
values (1, 2);
insert into CD_TRACK(CD_ID, TRACK_ID)
values (1, 3);
insert into CD_TRACK(CD_ID, TRACK_ID)
values (1, 4);
insert into CD_TRACK(CD_ID, TRACK_ID)
values (2, 5);

insert into CD_PLAYERS(CD_ID, PLAYER_ID)
values (1, 1);
insert into CD_PLAYERS(CD_ID, PLAYER_ID)
values (2, 1);

```



```
insert into CD_RECORDING_STUDIO(CD_ID, RECORDING_STUDIO_ID)
values (1, 1);
insert into CD_RECORDING_STUDIO(CD_ID, RECORDING_STUDIO_ID)
values (2, 1);

insert into GENRE_ARTISTS(ARTISTS_ID, GENRE_NAME)
values (1, 'Rock');
insert into GENRE_ARTISTS(ARTISTS_ID, GENRE_NAME)
values (2, 'Rock');

insert into GENRE_PLAYERS(PLAYER_ID, GENRE_NAME)
values (1, 'Rock');
insert into GENRE_PLAYERS(PLAYER_ID, GENRE_NAME)
values (1, 'Blues');

insert into MYPLAYLIST(MYPLAYLIST_NAME,
MYPLAYLIST_DESCRIPTION)
values ('Home playlist', 'Blues and rock songs');

insert into MYPLAYLIST_TRACK(MYPLAYLIST_NAME, TRACK_ID)
values ('Home playlist', 1);
insert into MYPLAYLIST_TRACK(MYPLAYLIST_NAME, TRACK_ID)
values ('Home playlist', 3);

insert into PLAYERS_ARTISTS(PLAYER_ID, ARTISTS_ID)
values (1, 1);
insert into PLAYERS_ARTISTS(PLAYER_ID, ARTISTS_ID)
values (1, 2);

insert into PLAYERS_TRACK(PLAYER_ID, TRACK_ID)
values (1, 1);
insert into PLAYERS_TRACK(PLAYER_ID, TRACK_ID)
values (1, 2);
insert into PLAYERS_TRACK(PLAYER_ID, TRACK_ID)
values (1, 3);
insert into PLAYERS_TRACK(PLAYER_ID, TRACK_ID)
values (1, 4);

insert into TRACK_ARTISTS(TRACK_ID, ARTISTS_ID)
values (1, 1);
insert into TRACK_ARTISTS(TRACK_ID, ARTISTS_ID)
values (2, 1);

commit;
```

4. Создан скрипт, изменяющий схему созданной БД по заданию преподавателя:

Ввести учет продаж треков и альбомов через интернет-магазин.

Листинг 3. Скрипт модификации созданной БД.

```
connect 'C:\DB\MUSICLIBRARY.FDB' user 'SYSDBA' password 'masterkey';

create table SALE_CD
(
  SALE_CD_ID int primary key,
  CD_ID int NOT NULL,
  CLIENT_ID int NOT NULL,
  SALE_DATA date NOT NULL,
  SALE_PRICE int NOT NULL
);

create table SALE_TRACK
(
  SALE_TRACK_ID int primary key,
  TRACK_ID int NOT NULL,
  CLIENT_ID int NOT NULL,
  SALE_DATA date NOT NULL,
  SALE_PRICE int NOT NULL
);

create table CLIENT
(
  CLIENT_ID int primary key,
  CLIENT_NAME varchar(255),
  CLIENT_LOGIN varchar(255) NOT NULL
);

create table SALE_CD_CLIENT
(
  SALE_CD_ID int,
  CLIENT_ID int,
  constraint PK_SALE_CD_CLIENT primary key (SALE_CD_ID, CLIENT_ID)
);

create table SALE_TRACK_CLIENT
(
  SALE_TRACK_ID int,
  CLIENT_ID int,
  constraint PK_SALE_TRACK_CLIENT primary key (SALE_TRACK_ID,
```

CLIENT_ID)

);

alter table SALE_CD_CLIENT add constraint SALE_CD_CLIENT_FK_0 foreign key (SALE_CD_ID) references SALE_CD(SALE_CD_ID);

alter table SALE_CD_CLIENT add constraint SALE_CD_CLIENT_FK_1 foreign key (CLIENT_ID) references CLIENT(CLIENT_ID);

alter table SALE_TRACK_CLIENT add constraint SALE_TRACK_CLIENT_FK_0 foreign key (SALE_TRACK_ID) references SALE_TRACK(SALE_TRACK_ID);

alter table SALE_TRACK_CLIENT add constraint SALE_TRACK_CLIENT_FK_1 foreign key (CLIENT_ID) references CLIENT(CLIENT_ID);

alter table SALE_TRACK add constraint SALE_TRACK_FK_0 foreign key (TRACK_ID) references TRACK(TRACK_ID);

alter table SALE_CD add constraint SALE_CD_FK_0 foreign key (CD_ID) references CD(CD_ID);

commit;

5. С помощью инструмента Database Designer среды IБExpert была получена ER-диаграмма созданной БД.

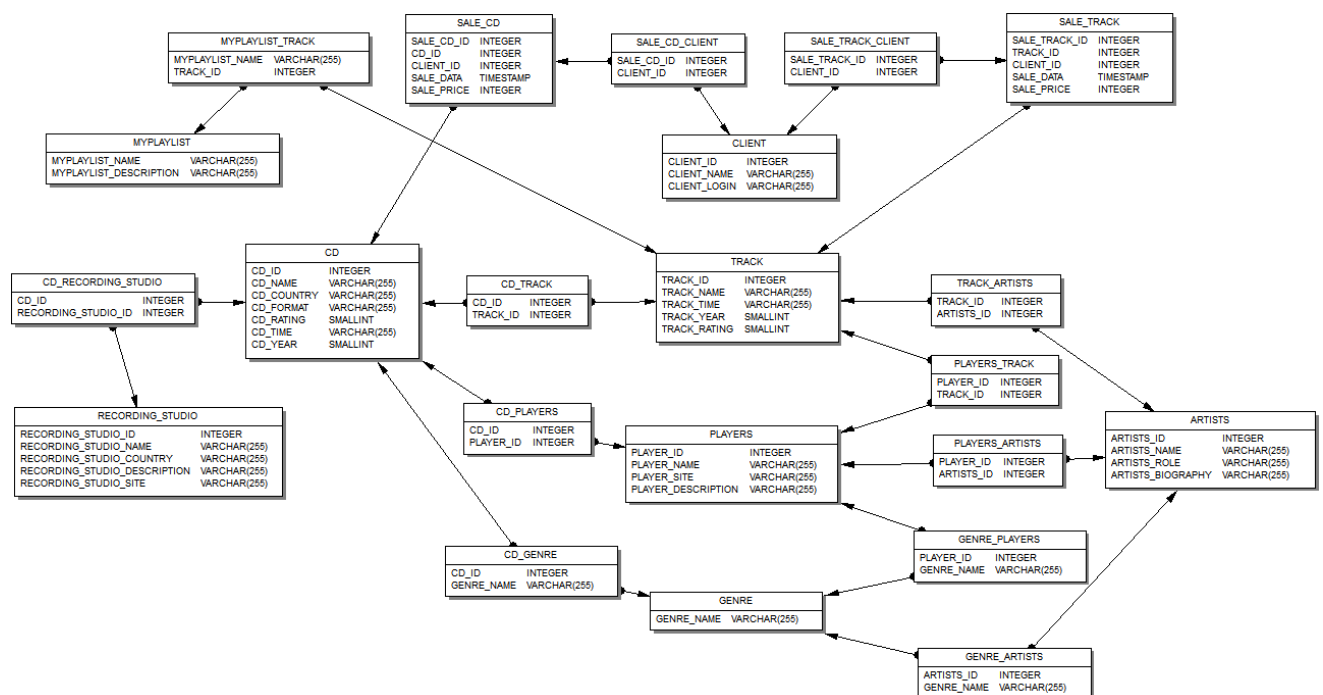


Рис. 1. ER-диаграмма БД

6. С помощью инструмента Test Data Generator среды IBExpert (рис. 2) были сгенерированы тестовые данные (по 100000 записей) для 3 таблиц БД: таблица CD, таблица TRACK, таблица PLAYERS.

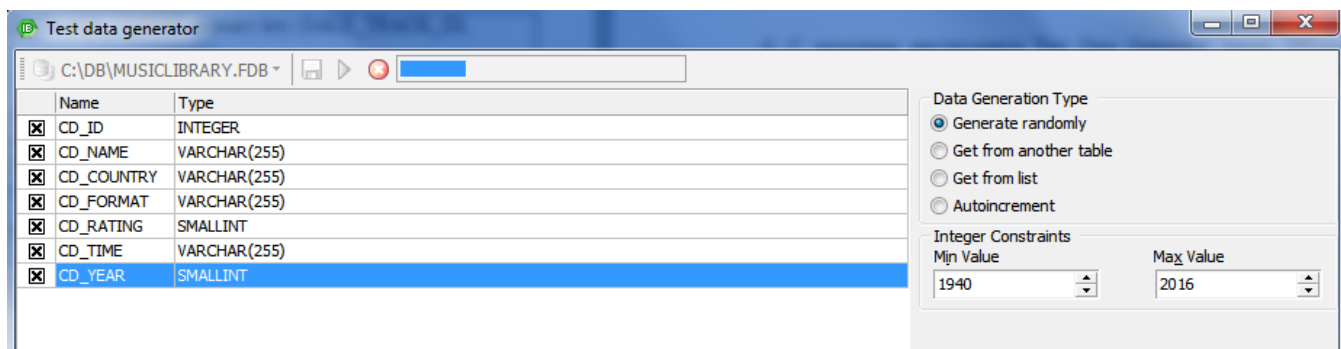


Рис. 2. Настройка инструмента Test Data Generator среды IBExpert для заполнения таблицы CD.

В результате генерации данных с помощью инструмента Test Data Generator среды IBExpert был создан скрипт заполнения таблицы.

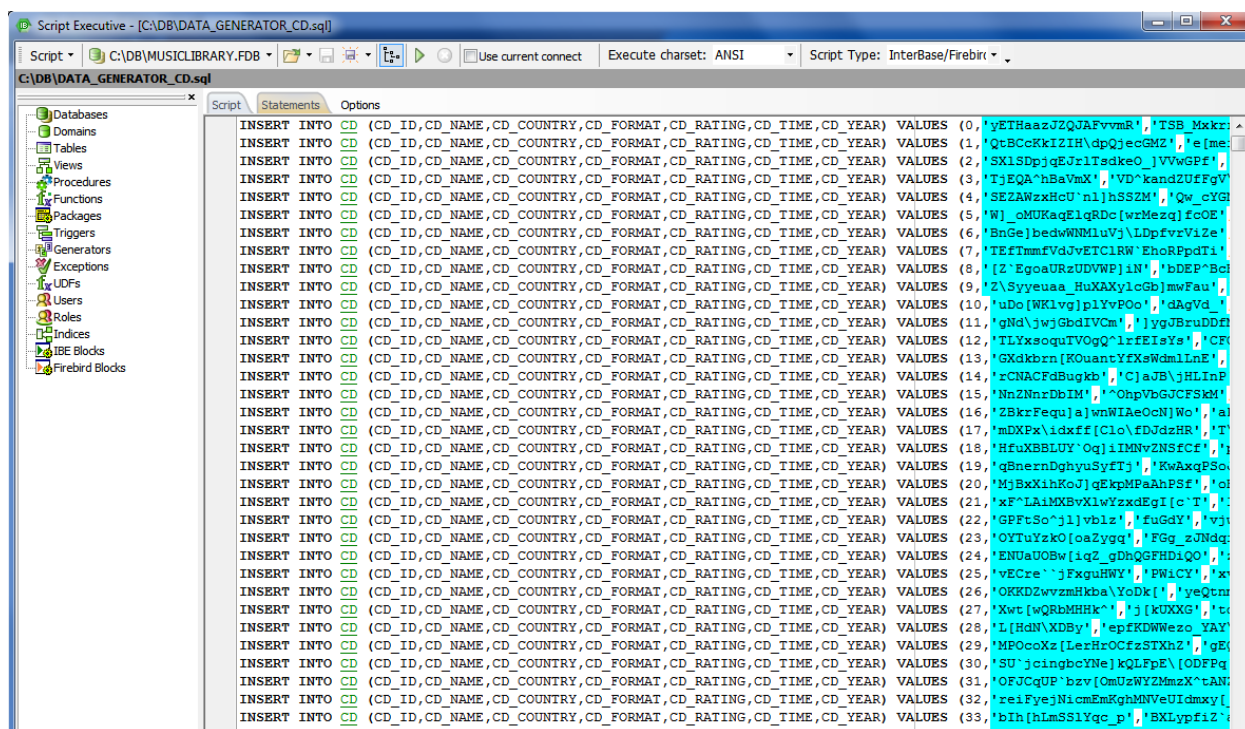


Рис. 3. Скрипт заполнения таблицы.

После выполнения скрипта заполняемая таблица содержит следующие данные (рис. 4):

Table : [CD] : C:\DB\MUSICLIBRARY.FDB (C:\DB\MUSICLIBRARY.FDB)

Table | Get record count | CD

Fields | Constraints | Indices | Dependencies | Triggers | Data | Master/Detail View | Description | DDL | Grants | Loggi

Record: 1 | Font size: 8

CD_ID	CD_NAME	CD_COUNTRY	CD_FORMAT	CD_RATING	CD_TIME	CD_YEAR
0	yETHaazJZQJAFvVmR	TSB_Mxkrrilxnddcv^Fc	jrjtdlsyix	10	60028349	2 004
1	QTBCCkIZIH\dpQjecGMZ	e[meixqCkmEm`mpkKZL	gsuwe	2	:6919564	1 957
2	SXISDpjQJrITsdkeO_JVWvGPf	XVe^b`B^cGWvVF	ovsjgzsz	7	74688914	1 951
3	TJEQA^hBaVmX	VD^kandZUffgV\	jwziafrkf	0	48898384	2 013
4	SEZAWzxHcU`n]hSSZM	Qw_cYGNs	nhbkc	4	94:88136	1 979
5	W]_oMUKaqElqRDc[wrMezq]fcOE	hjhmfld]_Jpck	biullc	3	49571754	1 971
6	BnGe]bedwWNmIuVj\LDpfvrViZe	un[YHHZSsMt	wznrebvx	4	56098915	1 998
7	TEfTmmfVdJvETCIRW`EhoRPpdTi	sTWHkpl]xYmoZt	fbyamvck	6	648182:7	1 987
8	[Z`EgoaURzUDVWP]iN	bDEP^BcBZkBPc	dsbko	3	08:62272	1 980
9	Z\Syyeuaa_HuXAXylcGb]mwFau	TN^bDbI[RKUOgLCczR\	bzbllrgg	3	989046:5	2 001
10	uDo[WKlv]pLYvPOo	dAgVd_	lbhwvutt	4	51272971	2 008
11	gNd\jwJGbdIVCm]ygJBruDDfMuImS	lnmwlexnrs	6	1676:510	2 015
12	TLYxsoquTVogQ^lrfEIsYs	CFCjke]BeFgOQ^zoX	gaguyr	8	73747762	1 957
13	GXdKbrn[KOuantYfxsWdmlLnE	oFfPQ	ymhuvv	4	75338493	1 951
14	rCNACFdBugkb	C]aJB\jHLInP	hwosmzfaj	9	724:8741	1 947
15	NnZNnrDbIM	^OhpVbGJCFSkM	klkdsqrt	5	682:3796	1 996
16	ZBkrFequ]a]wnWIAeOdN]Wo	aPGuWI_GGJY]DCF]MS	dwikomx	4	67:53340	1 964

Рис. 4. Результат выполнения скрипта, созданного инструментом Test Data Generator среды IBEExpert

Выводы

В результате работы ознакомился с языком SQL-DDL. Были написаны скрипты для создания, заполнения и модификации БД.

SQL-DDL (Data Definition Language) – язык определения структур и ограничений целостности баз данных. Использовались такие инструменты языка DDL как create (создать), alter (изменить), drop (удалить).

Основой языка SQL является язык реляционных БД. Табличная структура реляционных БД хорошо понятна, поэтому язык SQL так же прост для изучения. Язык SQL позволяет управлять структурой БД, это обеспечивает гибкость БД к изменяющимся требованиям.

Были изучены некоторые возможности среды IBEExpert: генерирование ER-диаграммы на основе созданной БД и заполнение таблиц текстовыми данными. Полученная ER-диаграмма полностью совпадает с разработанной диаграммой на предыдущем этапе, что говорит о том, что ошибок при создании БД не допущено.