

Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчёт о лабораторной работе №2

Дисциплина: Базы данных

Тема: Изучение языка SQL-DDL

Выполнил студент гр. 43501/1:

_____ Нагорнов А. А
(подпись)

Работу принял:

_____ Мяснов А.В.
(подпись)

«__» _____ 2016 г.

Санкт-Петербург

2016

1. Цели работы

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

2. Программа работы

- 2.1. Самостоятельное изучение SQL-DDL
- 2.2. Создание скрипта БД в соответствии с согласованной схемой (должны присутствовать первичные и внешние ключи, ограничения на диапазоны значений). Продемонстрировать скрипт преподавателю.
- 2.3. Создайте скрипт, заполняющий все таблицы БД данными
- 2.4. Выполнение SQL-запросов, изменяющих схему созданной БД по заданию преподавателя. Продемонстрировать их работу преподавателю.
- 2.5. Изучите основные возможности IBExpert. Получите ER-диаграмму созданной БД с помощью Database Designer.
- 2.6. Автоматически сгенерируйте данные при помощи IBExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

3. Выполнение работы

3.1. Изучение SQL-DDL

Язык DDL служит для создания и модификации структуры БД, т.е. для создания/изменения/удаления таблиц и связей. Основные команды создания и редактирования таблиц:

- CREATE TABLE имя_таблицы (перечисление полей и их типов, ограничений) – служит для создания новой таблицы в текущей БД;
- DROP TABLE имя_таблицы – служит для удаления таблицы из текущей БД;
- ALTER TABLE имя_таблицы ALTER COLUMN имя_столбца ... – служит для обновления типа столбца или для изменения его настроек (например для задания характеристики NULL или NOT NULL);
- ALTER TABLE имя_таблицы ADD CONSTRAINT имя_ограничения PRIMARY KEY(поле1, поле2,...) – добавление первичного ключа к уже существующей таблице;
- ALTER TABLE имя_таблицы DROP CONSTRAINT имя_ограничения – удаление ограничения из таблицы.
- ALTER TABLE имя_таблицы ADD перечень_полей_с_характеристиками – позволяет добавить новые поля в таблицу;
- ALTER TABLE имя_таблицы DROP COLUMN перечень_полей – позволяет удалить поля из таблицы;
- ALTER TABLE имя_таблицы ADD CONSTRAINT имя_ограничения FOREIGN KEY(поля) REFERENCES таблица_справочник(поля) – позволяет определить связь между таблицей и таблицей справочником.

3.2. Создание скрипта БД

Скрипт БД в соответствии с согласованной схемой:

```
create database 'C:\Users\Andrey\Desktop\BD\3\music3.fdb' user 'sysdba' password 'masterkey' ;
connect 'C:\Users\Andrey\Desktop\BD\3\music3.fdb' user 'sysdba' password 'masterkey' ;
```

```
CREATE TABLE Artists (
    Name varchar(50) NOT NULL,
    Biography varchar(255),
    Country varchar(50),
    PRIMARY KEY (Name)
);
CREATE TABLE Country (
    Country varchar(50) NOT NULL,
    PRIMARY KEY (Country)
);
CREATE TABLE Tracks (
    TrackID INT NOT NULL,
    TrackName varchar(50) NOT NULL,
    Artist varchar(50),
    Duration varchar(30),
    YearD INT,
    Language varchar(30),
    PRIMARY KEY (TrackID)
);
CREATE TABLE Albums (
    AlbumID INT NOT NULL,
    Artist varchar(50) NOT NULL,
    AlbumName varchar(50) NOT NULL,
    PRIMARY KEY (AlbumID)
);
CREATE TABLE MusicStyles (
    Style varchar(30) NOT NULL,
    PRIMARY KEY (Style)
);
CREATE TABLE Languages (
    Labguage varchar(30) NOT NULL,
    PRIMARY KEY (Labguage)
);
CREATE TABLE AlbumsSongs (
    ID INT NOT NULL,
    AlbumID INT NOT NULL,
    TrackID INT NOT NULL,
    PRIMARY KEY (ID)
);
CREATE TABLE MusicSOT (
    ID INT NOT NULL,
    StyleID varchar(30),
    TrackID INT,
    PRIMARY KEY (ID)
);
```

```

ALTER TABLE Artists
ADD CONSTRAINT Artists_fk0 FOREIGN KEY (Country) REFERENCES Country(Country);

ALTER TABLE Tracks
ADD CONSTRAINT Tracks_fk0 FOREIGN KEY (Artist) REFERENCES Artists(Name);

ALTER TABLE Tracks
ADD CONSTRAINT Tracks_fk1 FOREIGN KEY (Duration) REFERENCES MusicStyles(Style);

ALTER TABLE Tracks
ADD CONSTRAINT Tracks_fk2 FOREIGN KEY (Language) REFERENCES Languages(Labguage);

ALTER TABLE Albums
ADD CONSTRAINT Albums_fk0 FOREIGN KEY (Artist) REFERENCES Artists(Name);

ALTER TABLE AlbumsSongs
ADD CONSTRAINT AlbumsS_fk0 FOREIGN KEY (AlbumID) REFERENCES Albums(AlbumID);

ALTER TABLE AlbumsSongs
ADD CONSTRAINT AlbumsS_fk1 FOREIGN KEY (TrackID) REFERENCES Tracks(TrackID);

ALTER TABLE MusicSOT
ADD CONSTRAINT MusicSOT_fk0 FOREIGN KEY (StyleID) REFERENCES MusicStyles(Style);

ALTER TABLE MusicSOT
ADD CONSTRAINT MusicSOT_fk1 FOREIGN KEY (TrackID) REFERENCES Tracks(TrackID);

```

3.3. Создание скрипта заполнения БД

Скрипт, заполняющий все таблицы БД данными:

```

connect 'C:\Users\Andrey\Desktop\BD\3\music2.fdb' user 'sysdba' password 'masterkey' ;

INSERT INTO Languages VALUES ('Russian');
INSERT INTO Languages VALUES ('Latvian');
INSERT INTO Languages VALUES ('English');

INSERT INTO MusicStyles VALUES ('Punk');
INSERT INTO MusicStyles VALUES ('Classic');
INSERT INTO MusicStyles VALUES ('Rap');

INSERT INTO Country VALUES ('Russia');
INSERT INTO Country VALUES ('Latvia');

INSERT INTO Artists VALUES ('Shnurov', null, 'Russia');
INSERT INTO Artists VALUES ('Tsoi', null, 'Russia');

INSERT INTO Albums VALUES (1, 'Tsoi', 'Blood Type');

INSERT INTO Tracks VALUES (1, 'War', 'Tsoi', 'Punk', 1988, 'Russian');
INSERT INTO AlbumsSongs VALUES (1, 1, 1);
INSERT INTO MusicSOT VALUES (1, 'Punk', 1);

```

3.4. Изменение БД по заданию преподавателя

Необходимо:

- Реализовать учет отдельных исполнителей, групп с возможностью множественного участия исполнителей в группах.
- Реализовать учет состава групп при записи каждой композиции.
- Реализовать учет авторства композиций.

Для того, чтобы изменить БД по заданию преподавателя, было принято решение переписать ее, нежели воспользоваться командами изменения таблиц. Новый скрипт представлен ниже:

```
create database 'C:\Users\Andrey\Desktop\BD\3\music1.fdb' user 'sysdba' password 'masterkey' ;
connect 'C:\Users\Andrey\Desktop\BD\3\music1.fdb' user 'sysdba' password 'masterkey' ;

-- Таблица людей (исполнители/авторы)
CREATE TABLE people (
    ID INT NOT NULL,
    Name varchar(50) NOT NULL,
    PRIMARY KEY (ID)
);
-- У одного артиста мб несколько альбомов
CREATE TABLE album_artist (
    ID INT NOT NULL,
    album_id INT NOT NULL,
    people_id INT NOT NULL,
    PRIMARY KEY (ID)
);
-- В одном альбоме мб несколько треков
CREATE TABLE album_track (
    ID INT NOT NULL,
    album_id INT NOT NULL,
    track_id INT NOT NULL,
    PRIMARY KEY (ID)
);
-- Таблица треков
CREATE TABLE tracks (
    ID INT NOT NULL,
    name varchar(50) NOT NULL,
    language varchar(30),
    PRIMARY KEY (ID)
);
-- Таблица исполнителей (Если исполнитель группа,
-- то в параметре id_ispolnit будет NULL)
CREATE TABLE performers (
    ID INT NOT NULL,
```

```

        id_track INT NOT NULL,
        id_ispolnit INT,
        id_group INT,
        PRIMARY KEY (ID)
    );
-- Названия групп
CREATE TABLE groups (
    ID INT NOT NULL,
    name varchar(50) NOT NULL,
    PRIMARY KEY (ID)
);
-- Состав группы
CREATE TABLE group_members (
    ID INT NOT NULL,
    group_id INT NOT NULL,
    artist_id INT NOT NULL,
    PRIMARY KEY (ID)
);
-- Таблица авторов трека
CREATE TABLE authors (
    ID INT NOT NULL,
    track_id INT NOT NULL,
    artist_id INT NOT NULL,
    PRIMARY KEY (ID)
);
-- Название альбомов
CREATE TABLE albums (
    ID INT NOT NULL,
    name varchar(50) NOT NULL,
    PRIMARY KEY (ID)
);
-- У трека мб несколько жанров
CREATE TABLE style_track (
    ID INT NOT NULL,
    style_id INT NOT NULL,
    track_id INT NOT NULL,
    PRIMARY KEY (ID)
);
-- Таблица жанров
CREATE TABLE styles (
    ID INT NOT NULL,
    style varchar(30) NOT NULL,
    PRIMARY KEY (ID)
);
-- Таблица языков
CREATE TABLE languages (
    language varchar(30) NOT NULL,
    PRIMARY KEY (language)
);
-- Учет состава групп при записи каждой композиции
CREATE TABLE group_accounting (
    ID INT NOT NULL,
    group_mem INT NOT NULL,

```

```

        track_id INT NOT NULL,
        PRIMARY KEY (ID)
    );

ALTER TABLE album_artist ADD CONSTRAINT album_artist_fk0 FOREIGN KEY (album_id)
REFERENCES albums(ID);
ALTER TABLE album_artist ADD CONSTRAINT album_artist_fk1 FOREIGN KEY (people_id)
REFERENCES people(ID);
ALTER TABLE album_track ADD CONSTRAINT album_track_fk0 FOREIGN KEY (album_id)
REFERENCES albums(ID);
ALTER TABLE album_track ADD CONSTRAINT album_track_fk1 FOREIGN KEY (track_id)
REFERENCES tracks(ID);
ALTER TABLE tracks ADD CONSTRAINT tracks_fk1 FOREIGN KEY (language) REFERENCES
languages(language);
ALTER TABLE performers ADD CONSTRAINT performers_fk0 FOREIGN KEY (id_track)
REFERENCES tracks(ID);
ALTER TABLE performers ADD CONSTRAINT performers_fk1 FOREIGN KEY (id_ispolnit)
REFERENCES people(ID);
ALTER TABLE performers ADD CONSTRAINT performers_fk2 FOREIGN KEY (id_group)
REFERENCES groups(ID);
ALTER TABLE group_members ADD CONSTRAINT group_members_fk0 FOREIGN KEY (group_id)
REFERENCES groups(ID);
ALTER TABLE group_members ADD CONSTRAINT group_members_fk1 FOREIGN KEY (artist_id)
REFERENCES people(ID);
ALTER TABLE authors ADD CONSTRAINT authors_fk0 FOREIGN KEY (track_id) REFERENCES
tracks(ID);
ALTER TABLE authors ADD CONSTRAINT authors_fk1 FOREIGN KEY (artist_id) REFERENCES
people(ID);
ALTER TABLE style_track ADD CONSTRAINT style_track_fk0 FOREIGN KEY (style_id)
REFERENCES styles(ID);
ALTER TABLE style_track ADD CONSTRAINT style_track_fk1 FOREIGN KEY (track_id)
REFERENCES tracks(ID);
ALTER TABLE group_accounting ADD CONSTRAINT group_accounting_fk0 FOREIGN KEY
(group_mem) REFERENCES group_members(ID);
ALTER TABLE group_accounting ADD CONSTRAINT group_accounting_fk1 FOREIGN KEY
(track_id) REFERENCES tracks(ID);

INSERT INTO languages VALUES ('Russian');
INSERT INTO languages VALUES ('Latvian');
INSERT INTO languages VALUES ('English');

INSERT INTO styles VALUES (1, 'Punk');
INSERT INTO styles VALUES (2, 'Classic');
INSERT INTO styles VALUES (3, 'Rap');

INSERT INTO people VALUES (1, 'Shnurov');
INSERT INTO people VALUES (2, 'Tsoi');
INSERT INTO people VALUES (3, 'Pistoletov');
INSERT INTO people VALUES (4, 'Yuri Kasparyan');

INSERT INTO tracks VALUES (1, 'War', 'Russian');
INSERT INTO tracks VALUES (2, 'New russian pirate', 'Russian');
INSERT INTO tracks VALUES (3, 'Cuckoo', 'Russian');

```

```
INSERT INTO groups VALUES (1, 'Kino');
```

```
-- ID, group_id(groups), artist_id(people)
```

```
INSERT INTO group_members VALUES (1, 1, 2);
```

```
-- В группе КИНО - Цой
```

```
INSERT INTO group_members VALUES (2, 1, 4);
```

```
-- В группе КИНО - Юрий
```

```
-- ID, id_track(tracks), id_ispolnit(people), id_group(groups)
```

```
INSERT INTO performers VALUES (1, 1, null, 1);
```

```
INSERT INTO performers VALUES (2, 2, 3, null);
```

```
INSERT INTO albums VALUES (1, 'Blood type');
```

```
-- ID, album_id(albums), track_id(tracks)
```

```
INSERT INTO album_track VALUES (1, 1, 1);
```

```
-- В альбоме Blood type - песня War
```

```
INSERT INTO album_track VALUES (2, 1, 3);
```

```
-- В альбоме Blood type - песня Cuckoo
```

```
-- ID, album_id(albums), people_id(people)
```

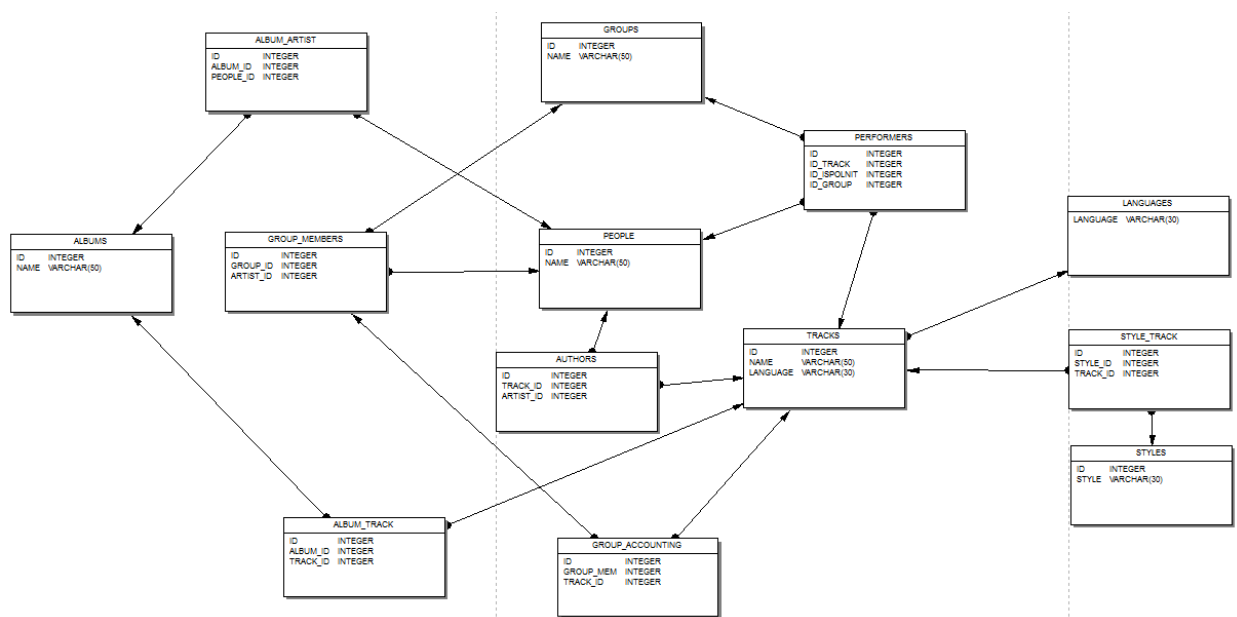
```
INSERT INTO album_artist VALUES (1, 1, 2);
```

```
-- В альбоме Blood type - участник - Цой
```

```
INSERT INTO album_artist VALUES (2, 1, 4);
```

```
-- В альбоме Blood type - участник - Юрий
```

3.5. ER-диаграмма созданной БД с помощью Database Designer



Увеличенная диаграмма представлена в Приложении 1.

3.6. Генерация данных для БД

Для таблиц Languages, People и Tracks сгенерируем по 100 000 тестовых данных.

Languages:

Font size: 8 100003 records fetched

LANGUAGE

- Ry/GcQ [Zmq?1+z#W/:CIZ@j!\$e
- RxaXKQbk!>)!6r.E\$1E~!89]rG\$
- Rx\!w)EHkSE]JV?L0Q*48~sTTX"@6
- RxSS.8}=VJ9zWCLLn4
- Rx>PemZn&o`m f
- Rx8VCINMXsa/#
- Rx7UcYF6;
- Rx)-[qMaJ]U[evGXE1ACQ7,<
- Rx n2[OLCt
- Rw}WZb40\C,<
- Rw{T5H@C!CDuXn7
- Rwy="s=s(L@*A.5fP8ct,YFhEK`jK&
- RwwTKI\$>'
- Rwks E#-?3^s7a&?A
- RwfPf5!([1LA)`~3#'?j~NXbP4/k
- RwD,~!!s/<^LG1DGYo0cWC?:k
- Rv{GHY:ii^eDmfc;ZC(!<&]qMTG*Wn
- RvfXES%oj
- RvVF%i!tTZvg%<G*R?Ty*J@F%&9M
- RvTCEO9BsZgp~2X@cH
- Rv49D36M|Hcl00T**44I7g
- Rv10H!x;-.,(+h^Z{
- Rv"(\$),X/p9l(DC5&5m
- Russian**
- Run"o&he]P;}AwF,(

1. Сетка 2. Форма 3. Печать

People:

Tracks:

ID	NAME	LANGUAGE
1	War	Russian
2	New russian pirate	Russian
3	Cuckoo	Russian
4	rq;X&Yz ^2pG,rx\00Y(6_~0:xR3cUUmKs	English
5	i&YB+xxNiQMORe.nREU	Russian
6	'a18*[Ng2w+)m[z5Y]'(uc{xLCj:9`	English
7	`>y^yvtgt>{ }%9DNFR_~aN?IPxj_x8}o	Latvian
8	PBk:'6=w5DQ}%l@L-HQ1wN[]+JnM@du}Or	Russian
9	q{U! *Y^wb@fKQW_7^L1,*mM,BJ	Latvian
10	gM2g)C1C8(/PoL2	English
11	10;BM/6-A0jKR:ljnNb*.112>`F`ko]Phb79ULJ,%mK..Co#	Russian
12	lt;>M19D;aI@MnCPB8aTfo.*nYfJ)^N)P&3xYG:*S>u	Russian
13	'dyhy:JDU/=RyGhDn~;DxGs+pf/I@RQMGYq_dSCb1	Russian
14	} N[3<'>'`))630prwLtmxNI=c`^/T0;I# +ki@PZe)K(0	Russian
15	*njI:>pSZ~J31A>(mpZNp-Y1:_j,t8&Vq;n	Latvian
16	XASagF1ob8&=sF]P8e:L7: K(BYn	Latvian
17	YaH ~BxAUYR2Y\$){5In?DQ= (b,<FmfrN6`Ei /2a`&Vv	English
18	-UjzmDN8aXvdB'3CIuxqj5Xo4K/CWH>%4Bi9[LQ%LRxJvi	English
19	0{7W3u{vQ?`	Russian
20	oe,+sP1EBEgIRL=YOJ=x<&*N&Pn/UFPcMIgKMI{m	English
21	yCVZ@_%q0NyaTw83I>_Un'io?G#:Q44O%&(vimD{.M n{n...	English
22	<F9^M&_E7hrW_+[] h?S=_L2aRYHz -!Zv gt	English
23	EPV9SY-MU--;mdeNasU	English
24	W?Ls`1l< WnVy(t-;UB{x,,	Latvian
25	:cveH#b^3j)/XZc:Q=;(147qBvo:8bk	Russian

1. Сетка 2. Форма 3. Печать

4. Выводы

Язык DDL служит для создания и модификации структуры БД, т.е. для создания/изменения/удаления таблиц и связей. С помощью скриптов была создана база данных музыки, которая в последствии была модифицирована по заданию преподавателя. Следует понимать, что, если в базе данных уже хранятся какие-то записи, необходимо с осторожностью подходить к ее модификации.

При заполнении базы данных в среде IVExpert, мы убедились в одном из основных свойств связей, а именно что ограничение внешнего ключа запрещает изменение данных в таблице первичного ключа. В случае неверного указания связей, может возникнуть ошибка вставки данных.

Приложение 1

