Санкт-Петербургский политехнический университет Петра Великого

Факультет технической кибернетики

Кафедра компьютерных систем и программных технологий

ОТЧЕТ

по лабораторной работе №3

# «SQL - DDL»

      Дисциплина: базы данных

Работу выполнила студентка гр. 43501/1 Фам Конг Минь.

Преподаватель \_\_\_\_\_\_\_ Мяснов А.В.

Санкт-Петербург

2015

**Раздел 1. SQL-DDL**

**1.1.  Цели работы**

Ознакомление с основами проектирования схемы БД, способами нормализации отношений в БД, языком описания сущностей и ограничений БД SQL-DDL.

**1.2. 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)

**1.3. Выполнение работы**

1. Создание скрипта БД в соответствии с согласованной схемой.

Скрипт создания БД приведен в листинге 1.

Листинг 1:

connect ‘D:/435011/FILMBASE.FDB’ user 'SYSDBA' password 'masterkey';

drop database;

commit;

create database ‘D:/435011/FILMBASE.FDB’ user 'SYSDBA' password 'masterkey';

commit;

CREATE TABLE country

(

     c\_id INT PRIMARY KEY,

     name VARCHAR(80) NOT NULL UNIQUE

);

CREATE INDEX c\_index ON country (name);

CREATE TABLE website

(

     s\_id INT PRIMARY KEY,

     name VARCHAR(255),

     url VARCHAR(255) NOT NULL

);

CREATE TABLE man

(

     man\_id INT PRIMARY KEY,

     name VARCHAR(80) NOT NULL,

     surname VARCHAR(80) NOT NULL,

     second\_name VARCHAR(80),

     birthday DATE NOT NULL,

     sex CHAR CHECK (sex IN ('M','F')),

     city VARCHAR(80) NOT NULL,

     country INT NOT NULL REFERENCES country(c\_id),

     site INT REFERENCES website(s\_id)

);

CREATE UNIQUE INDEX person ON man (name,surname,birthday);

CREATE TABLE career(

     id INT PRIMARY KEY,

     name VARCHAR(50) NOT NULL UNIQUE

);

CREATE TABLE genre(

     id INT PRIMARY KEY,

     name VARCHAR(50) NOT NULL UNIQUE

);

CREATE TABLE film

(

     f\_id INT PRIMARY KEY,

     name VARCHAR(100) NOT NULL,

     year\_of\_creation INT NOT NULL CHECK (year\_of\_creation >= 0),

     duration INT NOT NULL CHECK (duration >= 0),

     budget INT NOT NULL CHECK (budget >= 0),

     site INT REFERENCES website(s\_id)

);

CREATE UNIQUE INDEX movie ON film (name,year\_of\_creation);

CREATE TABLE soundtrack

(

     sound\_id INT PRIMARY KEY,

     name VARCHAR(255),

     author INT NOT NULL REFERENCES man(man\_id),

     film INT NOT NULL REFERENCES film(f\_id),

     duration TIME

);

CREATE INDEX music ON soundtrack(name,author);

CREATE TABLE award

(

     award\_id INT PRIMARY KEY,

     year\_aw INT NOT NULL CHECK (year\_aw >= 0),

     award VARCHAR(100) NOT NULL,

     result VARCHAR(50) NOT NULL CHECK (result IN ('won','nominated')),

     category VARCHAR(255) NOT NULL

);

CREATE TABLE company

(

     com\_id INT PRIMARY KEY,

     name VARCHAR(255) NOT NULL,

     year\_of\_foundation INT NOT NULL CHECK (year\_of\_foundation >= 0)

);

CREATE INDEX com\_index ON company(name);

CREATE TABLE rating

(

     r\_id INT PRIMARY KEY,

     rating INT NOT NULL CHECK (rating >= 0 AND rating <= 10),

     e\_mail VARCHAR(80) NOT NULL ,

     film\_id INT NOT NULL REFERENCES film(f\_id), CONSTRAINT rat\_man UNIQUE (e\_mail,film\_id)

);

CREATE TABLE part

(

     film\_id INT NOT NULL REFERENCES film(f\_id),

     man\_id INT NOT NULL REFERENCES man(man\_id),

     role\_id INT NOT NULL REFERENCES career(id),

     CONSTRAINT role PRIMARY KEY (film\_id,man\_id,role\_id)

);

CREATE INDEX part\_index ON part(film\_id,man\_id);

CREATE TABLE premium

(

     film\_id INT NOT NULL REFERENCES film(f\_id),

     man\_id INT REFERENCES man(man\_id),

     award\_id INT NOT NULL REFERENCES award(award\_id),

     CONSTRAINT film\_award PRIMARY KEY (film\_id,man\_id,award\_id)

);

CREATE INDEX premium\_index ON premium(film\_id,award\_id);

CREATE TABLE made\_in

(

     film\_id INT NOT NULL REFERENCES film(f\_id),

     country\_id INT NOT NULL REFERENCES country(c\_id),

     CONSTRAINT film\_country PRIMARY KEY (film\_id,country\_id)

);

CREATE TABLE owner

(

     man\_id INT NOT NULL REFERENCES man(man\_id),

     com\_id INT NOT NULL REFERENCES company(com\_id),

     CONSTRAINT owner\_company PRIMARY KEY (man\_id,com\_id)

);

CREATE TABLE production

(

     film\_id INT NOT NULL REFERENCES film(f\_id),

     com\_id INT NOT NULL REFERENCES company(com\_id),

     CONSTRAINT film\_company PRIMARY KEY (film\_id,com\_id)

);

CREATE TABLE style

(

     film\_id INT NOT NULL REFERENCES film(f\_id),

     genre\_id INT NOT NULL REFERENCES genre(id),

     CONSTRAINT film\_genre PRIMARY KEY (film\_id,genre\_id)

);

commit;

1. **Скрипт, заполняющий все таблицы БД данными.**

Скрипт заполнения БД приведен в листинге 2.

Листинг 2:

Connect ‘D:/435011/FILMBASE.fdb' user 'SYSDBA' password 'masterkey';

INSERT INTO country VALUES (1,'Russia');

INSERT INTO country VALUES (2,'USA');

INSERT INTO country VALUES (3,'Germany');

INSERT INTO country VALUES (4,'France');

INSERT INTO country VALUES (5,'United Kingdom');

INSERT INTO country VALUES (6,'Canada');

INSERT INTO country VALUES (7,'New Zealand');

INSERT INTO country VALUES (8,'Austria');

INSERT INTO website VALUES(1,'kinopoisk','http://www.kinopoisk.ru');

INSERT INTO website (s\_id,url) VALUES(2,'http://www.spielbergfilms.com');

INSERT INTO website (s\_id,url) VALUES(3,'http://www.lordoftherings.net');

INSERT INTO website (s\_id,url) VALUES(4,'http://www.titanicmovie.com');

INSERT INTO website VALUES(5,'movie database','http://www.imdb.com');

commit;

INSERT INTO man (man\_id,name,surname,birthday,sex,city,country,site)

VALUES(1,'Steven','Spielberg','1946-12-18','M','Cincinnati',2,2);

INSERT INTO man (man\_id,name,surname,second\_name,birthday,sex,city,country,site)

VALUES(2,'James','Cameron','Francis','1954-08-16','M','Kapuskasing',6,5);

INSERT INTO man (man\_id,name,surname,birthday,sex,city,country)

VALUES(3,'Woody','Allen','1935-12-01','M','New York',2);

INSERT INTO man (man\_id,name,surname,birthday,sex,city,country)

VALUES(4,'Peter','Jackson','1961-10-31','M','Pukerua Bay',7);

INSERT INTO man (man\_id,name,surname,birthday,sex,city,country)

VALUES(5,'Lyv','Tyler','1977-07-01','F','New York',2);

INSERT INTO man (man\_id,name,surname,birthday,sex,city,country)

VALUES(6,'Kate','Winslet','1975-10-05','F','Reading',5);

INSERT INTO man (man\_id,name,surname,birthday,sex,city,country)

VALUES(7,'Leonardo','DiCaprio','1974-11-11','M','Hollywood',2);

INSERT INTO man (man\_id,name,surname,second\_name,birthday,sex,city,country)

VALUES(8,'Arnold','Schwarzenegger','Alois','1947-07-30','M','Thal',8);

INSERT INTO man (man\_id,name,surname,birthday,sex,city,country)

VALUES(9,'James','Horner','1953-08-14','M','Los Angeles',2);

commit;

INSERT INTO career VALUES (1,'director');

INSERT INTO career VALUES (2,'writer');

INSERT INTO career VALUES (3,'actor');

INSERT INTO career VALUES (4,'producer');

INSERT INTO career VALUES (5,'composer');

INSERT INTO career VALUES (6,'operator');

INSERT INTO career VALUES (7,'designer');

INSERT INTO genre VALUES (1,'action');

INSERT INTO genre VALUES (2,'biography');

INSERT INTO genre VALUES (3,'western');

INSERT INTO genre VALUES (4,'crime');

INSERT INTO genre VALUES (5,'documentary');

INSERT INTO genre VALUES (6,'drama');

INSERT INTO genre VALUES (7,'comedy');

INSERT INTO genre VALUES (8,'melodrama');

INSERT INTO genre VALUES (9,'animation');

INSERT INTO genre VALUES (10,'musical');

INSERT INTO genre VALUES (11,'thriller');

INSERT INTO genre VALUES (12,'horror');

INSERT INTO genre VALUES (13,'fantasy');

INSERT INTO genre VALUES (14,'adventure');

INSERT INTO genre VALUES (15,'mystery');

INSERT INTO genre VALUES (16,'romance');

commit;

INSERT INTO film (f\_id,name,year\_of\_creation,duration,budget,site)

VALUES (1, 'The Lord of the Rings: The Fellowship of the Ring',2001,178,93000000,3);

INSERT INTO film (f\_id,name,year\_of\_creation,duration,budget,site)

VALUES (2, 'The Lord of the Rings: The Two Towers',2002,179,94000000,3);

INSERT INTO film(f\_id,name,year\_of\_creation,duration,budget,site)

VALUES (3, 'Titanic',1997,194,200000000,4);

INSERT INTO film(f\_id,name,year\_of\_creation,duration,budget,site)

VALUES (4, 'Avatar',2009,162,237000000,5);

INSERT INTO film (f\_id,name,year\_of\_creation,duration,budget,site)

VALUES (5,'The Terminator 2: Judgment Day',1991,137,102000000,1);

INSERT INTO film (f\_id,name,year\_of\_creation,duration,budget)

VALUES (6, 'Catch Me If You Can',2002,141,52000000);

INSERT INTO film (f\_id,name,year\_of\_creation,duration,budget)

VALUES (7, 'Manhattan',1979,96,39000000);

commit;

INSERT INTO soundtrack (sound\_id,name,author,film,duration)

VALUES (1,'My Heart Will Go ON',9,3,'05:08');

INSERT INTO soundtrack (sound\_id,name,author,film)

VALUES (2,'You Dont Dream In Cryo',9,4);

INSERT INTO soundtrack (sound\_id,name,author,film,duration)

VALUES (3,'Rose',9,3,'03:03');

INSERT INTO award(award\_id,year\_aw,award,result,category)

VALUES(1,1998,'Oscar','won','Best director');

INSERT INTO award(award\_id,year\_aw,award,result,category)

VALUES(2,1992,'Saturn Award','won','Best director');

INSERT INTO award(award\_id,year\_aw,award,result,category)

VALUES(3,1998,'Golden Globe','nominated','Best Performance by an Actor in a Motion Picture');

INSERT INTO award(award\_id,year\_aw,award,result,category)

VALUES(4,1998,'Oscar','won','Best music');

INSERT INTO award(award\_id,year\_aw,award,result,category)

VALUES(5,1980,'Oscar','nominated','Best Writing');

INSERT INTO award(award\_id,year\_aw,award,result,category)

VALUES(6,1998,'Golden Globe','won','Best music');

commit;

INSERT INTO company(com\_id,name,year\_of\_foundation)

VALUES(1,'Paramount Pictures',1912);

INSERT INTO company(com\_id,name,year\_of\_foundation)

VALUES(2,'Warner Bros.Pictures',1918);

INSERT INTO company(com\_id,name,year\_of\_foundation)

VALUES(3,'20th Century Fox',1935);

INSERT INTO company(com\_id,name,year\_of\_foundation)

VALUES(4,'Universal Studios',1912);

INSERT INTO rating(r\_id,rating,e\_mail,film\_id)

VALUES(1,9,'werttanya@mail.ru',3);

INSERT INTO rating(r\_id,rating,e\_mail,film\_id)

VALUES(4,7,'andrey@mail.ru',3);

INSERT INTO rating(r\_id,rating,e\_mail,film\_id)

VALUES(5,5,'masha@mail.ru',6);

INSERT INTO rating(r\_id,rating,e\_mail,film\_id)

VALUES(2,8,'AlenkiiAAA@yandex.ru',6);

INSERT INTO rating(r\_id,rating,e\_mail,film\_id)

VALUES(3,7,'dontsova23@mail.ru',1);

commit;

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(1,4,1);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(2,4,1);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(3,2,1);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(4,2,1);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(5,8,3);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(6,7,3);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(6,1,1);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(3,6,3);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(3,7,3);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(3,9,5);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(1,5,3);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(2,5,3);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(7,3,1);

INSERT INTO part(film\_id,man\_id,role\_id)

VALUES(7,3,2);

commit;

INSERT INTO premium(film\_id,man\_id,award\_id)

VALUES (3,2,1);

INSERT INTO premium(film\_id,man\_id,award\_id)

VALUES (3,9,4);

INSERT INTO premium(film\_id,man\_id,award\_id)

VALUES (3,7,3);

INSERT INTO premium(film\_id,man\_id,award\_id)

VALUES (5,2,2);

INSERT INTO premium(film\_id,man\_id,award\_id)

VALUES (7,3,5);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (3,2);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (1,2);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (1,7);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (2,2);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (2,7);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (4,2);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (4,5);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (5,2);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (6,2);

INSERT INTO made\_in(film\_id,country\_id)

VALUES (7,2);

INSERT INTO production(film\_id,com\_id)

VALUES (3,3);

INSERT INTO production(film\_id,com\_id)

VALUES (3,1);

INSERT INTO production(film\_id,com\_id)

VALUES (4,3);

INSERT INTO style(film\_id,genre\_id)

VALUES(1,13);

INSERT INTO style(film\_id,genre\_id)

VALUES(1,14);

INSERT INTO style(film\_id,genre\_id)

VALUES(2,13);

INSERT INTO style(film\_id,genre\_id)

VALUES(2,1);

INSERT INTO style(film\_id,genre\_id)

VALUES(3,8);

INSERT INTO style(film\_id,genre\_id)

VALUES(3,6);

INSERT INTO style(film\_id,genre\_id)

VALUES(4,13);

INSERT INTO style(film\_id,genre\_id)

VALUES(5,1);

INSERT INTO style(film\_id,genre\_id)

VALUES(5,13);

INSERT INTO style(film\_id,genre\_id)

VALUES(6,6);

INSERT INTO style(film\_id,genre\_id)

VALUES(6,2);

INSERT INTO style(film\_id,genre\_id)

VALUES(6,4);

commit;

1. **Выполнение SQL-запросов, изменяющих схему созданной БД** **по заданию преподавателя**

* Ввести учет прокатных компаний, разрешений на прокат фильмов, покупки разрешений компаниями.

CREATE TABLE RENTAL(

FILM\_ID INTEGER,

COM\_ID INTEGER,

PRICE INTEGER);

ALTER TABLE RENTAL ADD CONSTRAINT FKFILM\_ID FOREIGN KEY (FILM\_ID) REFERENCES

FILM (F\_ID);

ALTER TABLE RENTAL ADD CONSTRAINT FKCOM\_ID FOREIGN KEY (COM\_ID) REFERENCES

RENTAL\_COMPANY (COM\_ID);

CREATE TABLE RENTAL\_COMPANY(

COM\_ID INTEGER NOT NULL,

NAME VARCHAR(15),

ADDRESS VARCHAR(20));

ALTER TABLE RENTAL\_COMPANY ADD CONSTRAINT PK\_RENTAL\_COMPANY PRIMARY KEY (COM\_ID);

* Ввести учет показа фильмов в кинотеатрах с учетом выручки с каждого сеанса.

CREATE TABLE CINEMA(

FILM\_ID INTEGER,

SESSION\_ID INTEGER,

CASH INTEGER,

COUNTRY\_ID INTEGER,

RATING INTEGER);

ALTER TABLE CINEMA ADD CONSTRAINT FK\_FILM\_ID FOREIGN KEY (FILM\_ID) REFERENCES

FILM (F\_ID);

ALTER TABLE CINEMA ADD CONSTRAINT FK\_SESSION\_ID FOREIGN KEY (SESSION\_ID) REFERENCES

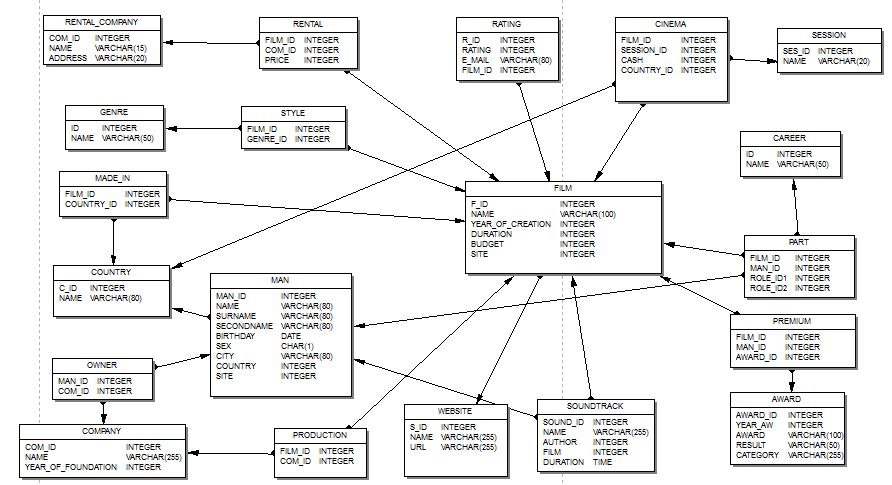
SESSION (SES\_ID);

ALTER TABLE CINEMA ADD CONSTRAINT FK\_COUNTRY\_ID FOREIGN KEY (COUNTRY\_ID) REFERENCES

COUNTRY (C\_ID);

1. **Создание ER- диаграммы в IBExpert**

**После регистрации созданной БД в IBExpert, выполнили Tools/Databasedesigner**

****

**Рис.1. ER- диаграмма**

**1.4.Выводы**

В результате работы был проведен анализ задания и построена модель БД. Реализовано создание БД, заполнение таблиц БД значениями, выполнены простейшие SQL-запросы. Результаты запросов корректны.