Санкт-Петербургский политехнический университет Петра Великого

Институт информационных технологий и управления

Кафедра компьютерных систем и программных технологий

ОТЧЕТ

По лабораторной работе №2

**«SQL-DDL»**

Базы данных

Студентка гр.43501/32: Бабуркина А.С.

Преподаватель: Мяснов А. В.

Санкт-Петербург

2014

**Цель работы**

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

1. **Программа работы**

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

1. **Выполнение программы работы**

**2.1. Скрипт создания данной базы данных:**

*CREATE TABLE author (id\_author INT PRIMARY KEY,name\_author VARCHAR(255) NOT NULL);*

*CREATE TABLE category (id\_category INT PRIMARY KEY,name\_category VARCHAR(255) NOT NULL);*

*CREATE TABLE publishing (id\_publishing INT PRIMARY KEY,name\_publishing VARCHAR(255) NOT NULL);*

*CREATE TABLE client (id\_client INT PRIMARY KEY,name\_client VARCHAR(255) NOT NULL, email VARCHAR(255), phone\_number VARCHAR(11));*

*CREATE TABLE export (id\_export INT PRIMARY KEY, type\_export VARCHAR(15) NOT NULL, cost\_export DECIMAL(5,2));*

*CREATE TABLE status\_buy (id\_status\_buy INT PRIMARY KEY, name\_status\_buy VARCHAR(255) NOT NULL);*

*CREATE TABLE orders (id\_order INT PRIMARY KEY, id\_client int references client (id\_client), id\_book int references book (id\_book), id\_status\_buy int references status\_buy(id\_status\_buy), export int references export (id\_export));*

*CREATE TABLE book (id\_book INT PRIMARY KEY, name\_book varchar(255) not null, id\_publishing int references publishing (id\_publishing), rating decimal(2,1), cost decimal(6,2) not null);*

*CREATE TABLE book\_categories (id\_book int references book (id\_book), id\_category int references category (id\_category) );*

*CREATE TABLE authors\_book (id\_book int references book (id\_book), id\_author int references author (id\_author));*

*CREATE TABLE review (id\_review INT PRIMARY KEY,text\_review VARCHAR(255) NOT NULL, id\_book int references book(id\_book));*

*CREATE TABLE reader (id\_reader INT PRIMARY KEY,name\_reader VARCHAR(255) NOT NULL, id\_review int references review (id\_review));*

**2.2. Скрипт, заполняющий все таблицы БД данными:**

*insert into author values(1, 'Stiven King');*

*insert into author values(2, 'Remark');*

*insert into author values(3, 'Bulgakov');*

*insert into category values(1, 'Classic');*

*insert into category values(2, 'Horror');*

*insert into category values(3, 'Novel');*

*insert into publishing values(1, 'Moscow');*

*insert into publishing values(2, 'Peter');*

*insert into publishing values(3, 'Art Studio');*

*insert into client values(1, 'Baburkina Nastya', 'baburkinanastya@gmail.com', '89112345678');*

*insert into client values(2, 'Baburkina Katya', 'baburkinakatya@gmail.com', '89112345679');*

*insert into client values(3, 'Petrova Nastya', 'retrovanastya@gmail.com', '89512345678');*

*insert into export values(1, 'Samovivoz', '0');*

*insert into export values(2, 'Dostavka', '400');*

*insert into status\_buy values(1, 'Polucheno');*

*insert into status\_buy values(2, 'Gotov');*

*insert into book values(1, 'Master i Margarita', 2, 10, 345);*

*insert into book values(2, 'Zhizn v zaimi', 1, 9, 400);*

*insert into book values(3, 'Ono', 3, 10, 500);*

*insert into book\_categories values(1, 1);*

*insert into book\_categories values(1, 3);*

*insert into book\_categories values(2, 3);*

*insert into book\_categories values(3, 2);*

*insert into authors\_book values(1, 3);*

*insert into authors\_book values(2, 2);*

*insert into authors\_book values(3, 1);*

*insert into review values(1, 'Book is cool', 1);*

*insert into review values(1, 'Book is nice', 2);*

*insert into review values(1, 'Book is good', 1);*

*insert into reader values(1, 'Durnev', 1);*

*insert into reader values(2, 'Durneva', 2);*

*insert into reader values(3, 'Bublikov', 3);*

*insert into orders values(1, 1, 1, 2, 1);*

*insert into orders values(2, 2, 3, 2, 1);*

*insert into orders values(3, 3, 2, 2, 1);*

**2.3. Изменяющим схему созданной БД:**

В существующую базу данных добавим функцию определения похожих книг. Скрипт с учетом модификации:

*CREATE TABLE interesting\_book (id\_book int references book (id\_book), id\_client int references client(id\_client));*

*CREATE TABLE similar\_book (id\_book1 int references book (id\_book), id\_book2 int references book (id\_book));*

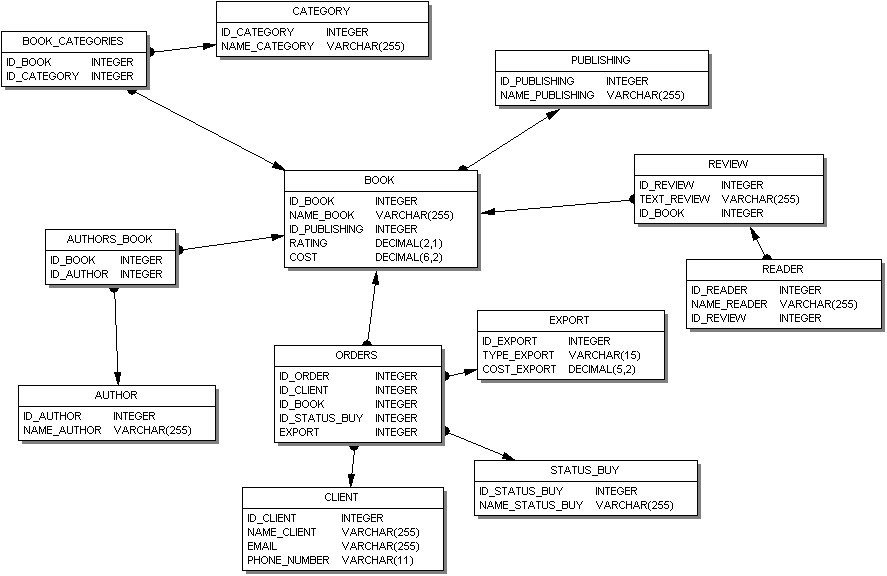
**

Рис.1. ER-диаграмма

**2.4. Автоматически сгенерируем данные при помощи IBExpert**:

Заполнение таблицы *BOOK\_CATEGORIES:*

*INSERT INTO BOOK\_CATEGORIES (ID\_BOOK,ID\_CATEGORY) VALUES (-1506385213,-2145583660);*

*INSERT INTO BOOK\_CATEGORIES (ID\_BOOK,ID\_CATEGORY) VALUES (169527593,-1685088814);*

*INSERT INTO BOOK\_CATEGORIES (ID\_BOOK,ID\_CATEGORY) VALUES (1504098591,-2135548512);*

*INSERT INTO BOOK\_CATEGORIES (ID\_BOOK,ID\_CATEGORY) VALUES (1608357925,318214334);*

*INSERT INTO BOOK\_CATEGORIES (ID\_BOOK,ID\_CATEGORY) VALUES (-2089791557,-668041556);*

*INSERT INTO BOOK\_CATEGORIES (ID\_BOOK,ID\_CATEGORY) VALUES (-1885229727,-1499534614);*

*INSERT INTO BOOK\_CATEGORIES (ID\_BOOK,ID\_CATEGORY) VALUES (-510410089,1828269304);*

....

Заполнение таблицы *ORDERS:*

*INSERT INTO ORDERS (ID\_ORDER,ID\_STATUS\_BUY,EXPORT) VALUES (-1676538502,-1624920217,1164840200);*

*INSERT INTO ORDERS (ID\_ORDER,ID\_STATUS\_BUY,EXPORT) VALUES (-903420627,-1705102874,1556747651);*

*INSERT INTO ORDERS (ID\_ORDER,ID\_STATUS\_BUY,EXPORT) VALUES (-819299436,2016755177,205970834);*

*INSERT INTO ORDERS (ID\_ORDER,ID\_STATUS\_BUY,EXPORT) VALUES (1278877663,1955733344,-1575968539);*

*....*

Заполнение таблицы *STATUS\_BUY:*

*INSERT INTO STATUS\_BUY (ID\_STATUS\_BUY,NAME\_STATUS\_BUY) VALUES (-1342456124,'KAN8^%n4f#uE4Ng,>AL73r2F8l}I4v=#\*$1\_!v=L7QeZM^kGw(M/G;F89h$I-);*

*INSERT INTO STATUS\_BUY (ID\_STATUS\_BUY,NAME\_STATUS\_BUY) VALUES (-385272259,'Lca3}3w;Y(1 ;p?WOiq:8wik\*w%LpAG!R9<OF$H{Nw=v~QPX~MX`kz#g);*

*INSERT INTO STATUS\_BUY (ID\_STATUS\_BUY,NAME\_STATUS\_BUY) VALUES (1166655821,'bI\77MNW5}$ @uI');*

*....*

**Выводы**

Язык описания данных DDL — это некий набор функций, используемый для определения структур данных. Эти инструкции используются для создания, изменения и удаления структур данных. К достоинствам языка можно отнести интуитивно понятную лексику команд, необходимый набор команд, хоть и минимальный. При описании структур данных необходимо брать в расчет то, что возможно придется изменять базу данных, и учесть возможные варианты модификаций БД.

В данной работе также была изучена оболочка СУБД - IBExpert, которая позволяет работать сразу с несколькими БД, строит с помощью визуального редактора наглядные ER-диаграммы, на которых можно проследить связь между различными таблицами, позволяет автоматически заполнять таблицы данными и многое другое.