Санкт-Петербургский политехнический университет

Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе

Дисциплина: «Базы данных»

Тема: «Язык SQL-DDL»

Выполнил студент гр. 43501/3 Козлов М.Н.

(подпись)

Преподаватель А.В. Мяснов

(подпись)

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г.

Санкт-Петербург

2016

Оглавление

[Цель работы: 3](file:///C:\Users\Михаил\Downloads\Telegram%20Desktop\Report%20SQL_DDL.%20Popsuyko%20Maxim.docx#_Toc465781390)

[Выполнение работы: 3](file:///C:\Users\Михаил\Downloads\Telegram%20Desktop\Report%20SQL_DDL.%20Popsuyko%20Maxim.docx#_Toc465781391)

[Создание первичной схемы БД 3](file:///C:\Users\Михаил\Downloads\Telegram%20Desktop\Report%20SQL_DDL.%20Popsuyko%20Maxim.docx#_Toc465781392)

[Модернизация БД 5](file:///C:\Users\Михаил\Downloads\Telegram%20Desktop\Report%20SQL_DDL.%20Popsuyko%20Maxim.docx#_Toc465781393)

[Автоматическая генерация данных для заполнения БД с помощью IBExpert 7](file:///C:\Users\Михаил\Downloads\Telegram%20Desktop\Report%20SQL_DDL.%20Popsuyko%20Maxim.docx#_Toc465781394)

[Выводы: 9](file:///C:\Users\Михаил\Downloads\Telegram%20Desktop\Report%20SQL_DDL.%20Popsuyko%20Maxim.docx#_Toc465781395)

# Цель работы:

1. Самостоятельное изучение SQL-DDL;
2. Создание скрипта БД в соответствии с согласованной схемой (первичные и внешние ключи, ограничения на диапазоны значений);
3. Создать скрипт, заполняющий все таблицы БД данными;
4. Выполнение SQL-запросов, изменяющих схему созданной БД;
5. Изучение основных возможностей IBExpert. Получение ER-диаграммы, созданной БД с помощью Database Designer;
6. Автоматически сгенерировать данные при помощи IBExpert (не менее 100000 записей в каждой из выбранных таблиц).

# Выполнение работы:

В качестве индивидуального задания была выбрана тема «Создание базы данных интернет-магазина книг».

Создание первичной схемы БД

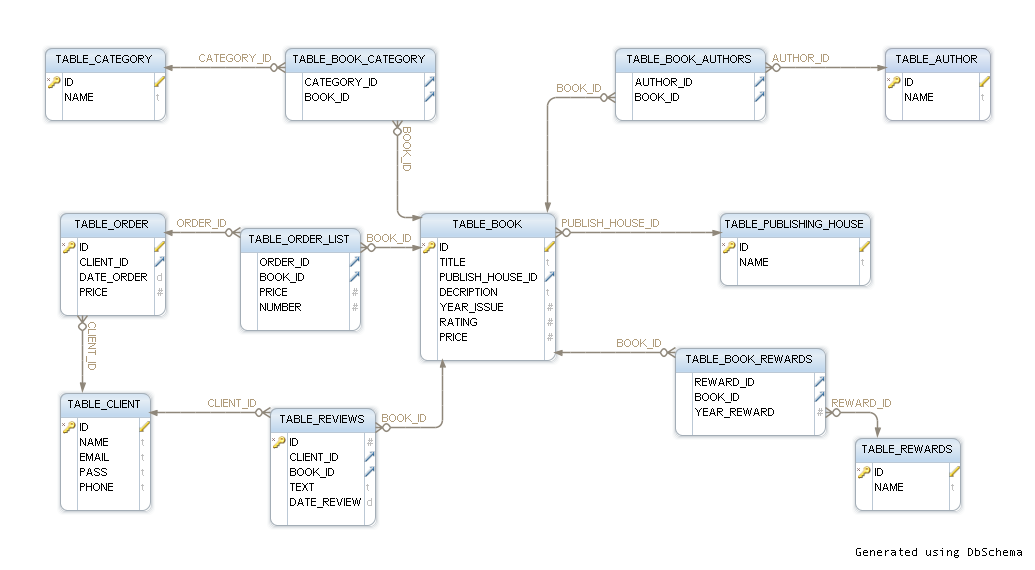


Рис. 1 SQL-схема БД

Согласно данной схеме был написан скрипт по созданию необходимых таблицы с ограничениями на диапазоны значений:

CREATE TABLE TABLE\_AUTHOR

(

ID INTEGER NOT NULL,

"NAME" VARCHAR( 100) COLLATE NONE

);

CREATE TABLE TABLE\_BOOK

(

ID INTEGER NOT NULL,

TITLE VARCHAR( 100) COLLATE NONE,

PUBLISH\_HOUSE\_ID INTEGER,

DECRIPTION VARCHAR( 200) COLLATE NONE,

YEAR\_ISSUE INTEGER,

RATING FLOAT,

PRICE FLOAT

);

CREATE TABLE TABLE\_BOOK\_AUTHORS

(

AUTHOR\_ID INTEGER,

BOOK\_ID INTEGER

);

CREATE TABLE TABLE\_BOOK\_CATEGORY

(

CATEGORY\_ID INTEGER,

BOOK\_ID INTEGER

);

CREATE TABLE TABLE\_BOOK\_REWARDS

(

REWARD\_ID INTEGER,

BOOK\_ID INTEGER,

YEAR\_REWARD INTEGER

);

CREATE TABLE TABLE\_CATEGORY

(

ID INTEGER NOT NULL,

"NAME" VARCHAR( 100) COLLATE NONE

);

CREATE TABLE TABLE\_CLIENT

(

ID INTEGER NOT NULL,

"NAME" VARCHAR( 100) COLLATE NONE,

EMAIL VARCHAR( 100) COLLATE NONE,

PASS VARCHAR( 100) COLLATE NONE,

PHONE VARCHAR( 100) COLLATE NONE

);

CREATE TABLE TABLE\_ORDER

(

ID INTEGER NOT NULL,

CLIENT\_ID INTEGER,

DATE\_ORDER DATE,

PRICE FLOAT

);

CREATE TABLE TABLE\_ORDER\_LIST

(

ORDER\_ID INTEGER,

BOOK\_ID INTEGER,

PRICE FLOAT,

NUMBER INTEGER

);

CREATE TABLE TABLE\_PUBLISHING\_HOUSE

(

ID INTEGER NOT NULL,

"NAME" VARCHAR( 100) COLLATE NONE

);

CREATE TABLE TABLE\_REVIEWS

(

ID INTEGER NOT NULL,

CLIENT\_ID INTEGER,

BOOK\_ID INTEGER,

TEXT VARCHAR( 250) COLLATE NONE,

DATE\_REVIEW DATE

);

CREATE TABLE TABLE\_REWARDS

(

ID INTEGER NOT NULL,

"NAME" VARCHAR( 100) COLLATE NONE

);

ALTER TABLE TABLE\_AUTHOR ADD CONSTRAINT PK\_TABLE\_AUTHOR PRIMARY KEY

(ID);

ALTER TABLE TABLE\_BOOK ADD CONSTRAINT PK\_TABLE\_BOOK PRIMARY KEY

(ID);

ALTER TABLE TABLE\_CATEGORY ADD CONSTRAINT PK\_TABLE\_CATEGORY PRIMARY KEY

(ID);

ALTER TABLE TABLE\_CLIENT ADD CONSTRAINT PK\_TABLE\_CLIENT PRIMARY KEY

(ID);

ALTER TABLE TABLE\_ORDER ADD CONSTRAINT PK\_TABLE\_ORDER PRIMARY KEY

(ID);

ALTER TABLE TABLE\_PUBLISHING\_HOUSE ADD CONSTRAINT PK\_TABLE\_PUBLISHING\_HOUSE PRIMARY KEY

(ID);

ALTER TABLE TABLE\_REVIEWS ADD CONSTRAINT PK\_TABLE\_REVIEWS PRIMARY KEY

(ID);

ALTER TABLE TABLE\_REWARDS ADD CONSTRAINT PK\_TABLE\_REWARDS PRIMARY KEY

(ID);

Добавим foreign key в созданные таблицы:

ALTER TABLE TABLE\_BOOK ADD CONSTRAINT FK\_TABLE\_BOOK

FOREIGN KEY (PUBLISH\_HOUSE\_ID) REFERENCES TABLE\_PUBLISHING\_HOUSE

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_BOOK\_AUTHORS ADD CONSTRAINT FK\_TABLE\_BOOK\_AUTHORS

FOREIGN KEY (AUTHOR\_ID) REFERENCES TABLE\_AUTHOR

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_BOOK\_AUTHORS ADD CONSTRAINT FK\_TABLE\_BOOK\_AUTHORS\_TWO

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_BOOK\_CATEGORY ADD CONSTRAINT FK\_TABLE\_BOOK\_CATEGORY

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_BOOK\_CATEGORY ADD CONSTRAINT FK\_TABLE\_BOOK\_CATEGORY\_TWO

FOREIGN KEY (CATEGORY\_ID) REFERENCES TABLE\_CATEGORY

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_BOOK\_REWARDS ADD CONSTRAINT FK\_TABLE\_BOOK\_REWARDS

FOREIGN KEY (REWARD\_ID) REFERENCES TABLE\_REWARDS

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_BOOK\_REWARDS ADD CONSTRAINT FK\_TABLE\_BOOK\_REWARDS\_TWO

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_ORDER ADD CONSTRAINT FK\_TABLE\_ORDER\_TABLE\_CLIENT

FOREIGN KEY (CLIENT\_ID) REFERENCES TABLE\_CLIENT

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_ORDER\_LIST ADD CONSTRAINT FK\_TABLE\_ORDER\_LIST

FOREIGN KEY (ORDER\_ID) REFERENCES TABLE\_ORDER

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_ORDER\_LIST ADD CONSTRAINT FK\_TABLE\_ORDER\_LIST\_TABLE\_BOOK

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_REVIEWS ADD CONSTRAINT FK\_TABLE\_REVIEWS\_TABLE\_BOOK

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE TABLE\_REVIEWS ADD CONSTRAINT FK\_TABLE\_REVIEWS\_TABLE\_CLIENT

FOREIGN KEY (CLIENT\_ID) REFERENCES TABLE\_CLIENT

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

Модернизация БД

Дополнительное задание по модернизации исходной таблицы:

* Реализовать учет поставок, складов, и контроля количества остатков товаров.
* Реализовать структуру для рекомендации книг при создании заказа клиентом.

Изменим схему БД с учетом дополнительного задания:

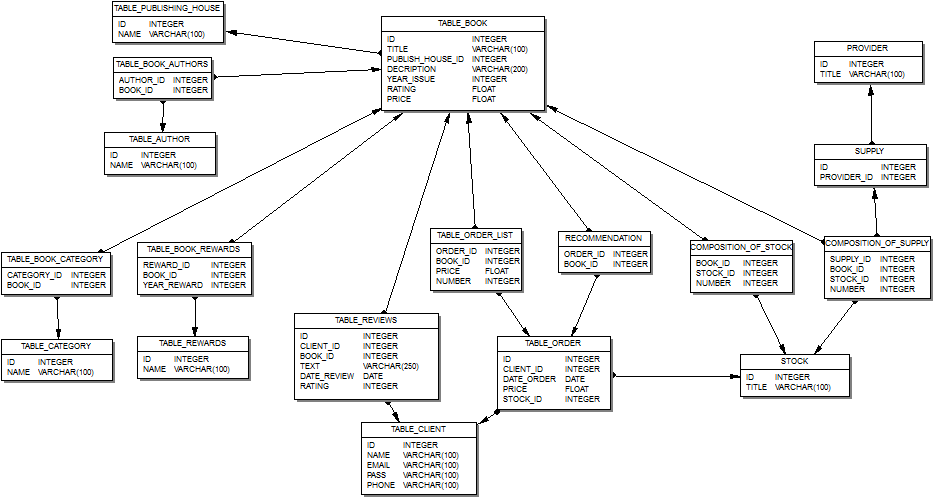


Рис. 3 ER-диаграмма БД

Скрипт для изменения:

CREATE TABLE RECOMMENDATION

(

ORDER\_ID INTEGER,

BOOK\_ID INTEGER

);

ALTER TABLE RECOMMENDATION ADD CONSTRAINT FK\_RECOMMENDATION\_ORDER\_ID

FOREIGN KEY (ORDER\_ID) REFERENCES TABLE\_ORDER

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE RECOMMENDATION ADD CONSTRAINT FK\_RECOMMENDATION\_BOOK\_ID

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

CREATE TABLE PROVIDER

(

ID INTEGER NOT NULL,

TITLE VARCHAR( 100) COLLATE NONE

);

CREATE TABLE SUPPLY

(

ID INTEGER NOT NULL,

PROVIDER\_ID INTEGER

);

CREATE TABLE STOCK

(

ID INTEGER NOT NULL,

TITLE VARCHAR( 100) COLLATE NONE

);

CREATE TABLE COMPOSITION\_OF\_SUPPLY

(

SUPPLY\_ID INTEGER NOT NULL,

BOOK\_ID INTEGER NOT NULL,

STOCK\_ID INTEGER NOT NULL,

NUMBER INTEGER NOT NULL

);

CREATE TABLE COMPOSITION\_OF\_STOCK

(

BOOK\_ID INTEGER NOT NULL,

STOCK\_ID INTEGER NOT NULL,

NUMBER INTEGER NOT NULL

);

ALTER TABLE PROVIDER ADD CONSTRAINT PK\_PROVIDER PRIMARY KEY

(ID);

ALTER TABLE SUPPLY ADD CONSTRAINT PK\_SUPPLY PRIMARY KEY

(ID);

ALTER TABLE STOCK ADD CONSTRAINT PK\_STOCK PRIMARY KEY

(ID);

ALTER TABLE SUPPLY ADD CONSTRAINT FK\_PROVIDER\_ID

FOREIGN KEY (PROVIDER\_ID) REFERENCES PROVIDER

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE COMPOSITION\_OF\_SUPPLY ADD CONSTRAINT FK\_SUPPLY\_ID

FOREIGN KEY (SUPPLY\_ID) REFERENCES SUPPLY

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE COMPOSITION\_OF\_SUPPLY ADD CONSTRAINT FK\_SUP\_BOOK\_ID

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE COMPOSITION\_OF\_SUPPLY ADD CONSTRAINT FK\_SUP\_STOCK\_ID

FOREIGN KEY (STOCK\_ID) REFERENCES STOCK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE COMPOSITION\_OF\_STOCK ADD CONSTRAINT FK\_STOCK\_STOCK\_ID

FOREIGN KEY (STOCK\_ID) REFERENCES STOCK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

ALTER TABLE COMPOSITION\_OF\_STOCK ADD CONSTRAINT FK\_STOCK\_BOOK\_ID

FOREIGN KEY (BOOK\_ID) REFERENCES TABLE\_BOOK

(ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

;

Автоматическая генерация данных для заполнения БД с помощью IBExpert

Для заполнения воспользуемся встроенной в IBExpert программой, запустим ее: tools/test data generator, и выполним настройку:

* В поле «Table» выбираем название заполняемой таблицы;
* Выбираем нужные колонки для заполнения;
* Выставляем для каждой выбранной колонки параметры.

В каждую таблицу внесем по 100.000 случайных значений.

# Выводы:

В ходе выполнения данной работы ознакомился с основами языка определения данных SQL-DDL. Придумал собственную базу данных и составил по ней схему из таблиц. Создал скрипты создания таблиц, добавления ключей в соответствии с моей БД. Научился заполнять таблицы БД данными. При создании БД наиболее трудным было создавать скрипты, так как при допущении в них ошибок, исправить их было трудоемко.

Познакомился с СУБД IBExpert. Данная программа облегчает создание и работу с БД. В данной СУБД присутствует множество различных инструментов, например, инструменты для создания ER-диаграмм, инструмент для генерации случайных записей в таблицы.