Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт о лабораторной работе №3**

**Дисциплина**: Базы данных

**Тема**: Язык SQL-DDL

Выполнил студент гр. 43501/1 Приймак А.Г.

(подпись)

Руководитель Мяснов А.В.

(подпись)

“ ” 2016 г.

Санкт-Петербург

2016

1. **Цель работы**

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

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

SQL (англ. structured query language — «язык структурированных запросов») — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД, в данном случае Firebird)

Data Definition Language (DDL) (язык описания данных) — это семейство компьютерных языков, используемых в компьютерных программах для описания структуры баз данных.

Языки DDL могут существенно различаться у различных производителей СУБД. Существует ряд стандартов SQL, установленный ISO/IEC (SQL-89,SQL-92, SQL:1999, SQL:2003, SQL:2008), но производители СУБД часто предлагают свои собственные «расширения» языка и, часто, не поддерживают стандарт полностью.

Функции языков DDL определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL это глаголы — «create» («создать»), «alter» («изменить»), «drop» («удалить»). Эти запросы или команды часто смешиваются с другими командами SQL, в связи с чем DDL не является отдельным компьютерным языком.

1. **Ход работы**

Были создан скрипты, создающие базу данных в соответствии со схемой:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | CREATE TABLE countries ( | |  | id int NOT NULL, | |  | country char(100) NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE catalog ( | |  | id int NOT NULL, | |  | title int NOT NULL, | |  | id\_class char(20) NOT NULL, | |  | id\_country int NOT NULL, | |  | id\_def int NOT NULL, | |  | id\_caliber int NOT NULL, | |  | price int NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE definition ( | |  | id int NOT NULL, | |  | application\_area char(50) NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE caliber ( | |  | id int NOT NULL, | |  | caliber int NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE items ( | |  | id int NOT NULL, | |  | id\_cat int NOT NULL, | |  | unique\_number int NOT NULL, | |  | is\_sold int NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE ttc ( | |  | id int NOT NULL, | |  | cat\_id int NOT NULL, | |  | mass int NOT NULL, | |  | bullet\_mass int NOT NULL, | |  | bullet\_type int NOT NULL, | |  | barrel\_type int NOT NULL, | |  | start\_speed int NOT NULL, | |  | range int NOT NULL, | |  | length int NOT NULL, | |  | width int NOT NULL, | |  | capacity int NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE stock ( | |  | id int NOT NULL, | |  | id\_cat int NOT NULL, | |  | quantity int NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | commit; | |  |  | |  | ALTER TABLE catalog ADD CONSTRAINT catalog\_fk0 FOREIGN KEY (id\_country) REFERENCES countries(id); | |  |  | |  | ALTER TABLE catalog ADD CONSTRAINT catalog\_fk1 FOREIGN KEY (id\_def) REFERENCES definition(id); | |  |  | |  | ALTER TABLE catalog ADD CONSTRAINT catalog\_fk2 FOREIGN KEY (id\_caliber) REFERENCES caliber(id); | |  |  | |  | ALTER TABLE items ADD CONSTRAINT items\_fk0 FOREIGN KEY (id\_cat) REFERENCES catalog(id); | |  |  | |  | ALTER TABLE ttc ADD CONSTRAINT ttc\_fk0 FOREIGN KEY (cat\_id) REFERENCES catalog(id); | |  |  | |  | ALTER TABLE stock ADD CONSTRAINT stock\_fk0 FOREIGN KEY (id\_cat) REFERENCES catalog(id); | |  |  | |  | commit; | |

Были созданы скрипты, заполняющие БД данными (пример скрипт, заполняющий таблицу классов и стран-производителей:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | insert into classes values (0, 'handgun'); | |  | insert into classes values (1, 'rifle'); | |  | insert into classes values (2, 'as\_rifle'); | |  | insert into classes values (3, 'smg'); | |  | insert into classes values (4, 'carbine'); | |  | insert into classes values (5, 'shotgun'); | |  | insert into classes values (6, 'lmg'); | |  |  | |  | insert into countries values (0, 'Russia'); | |  | insert into countries values (1, 'USA'); | |  | insert into countries values (2, 'UK'); | |  | insert into countries values (3, 'Germany'); | |  | insert into countries values (4, 'Israel'); | |  | insert into countries values (5, 'France'); | |  | insert into countries values (6, 'Belgia'); | |

Затем были созданы скрипт, модифицирующие базу данных согласно заданию (добавляются новые таблицы и связи между ними) ( в новых таблицах сразу же создаются некоторые данные):

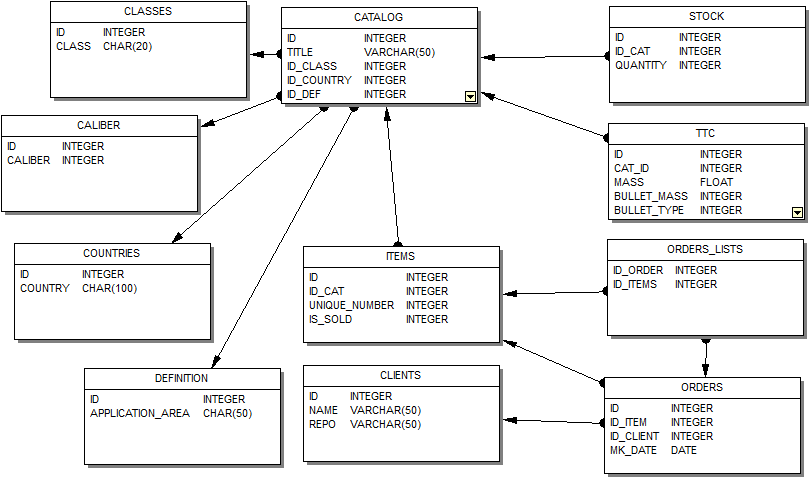
Модифицировать схему БД для удовлетворения следующим условиям:

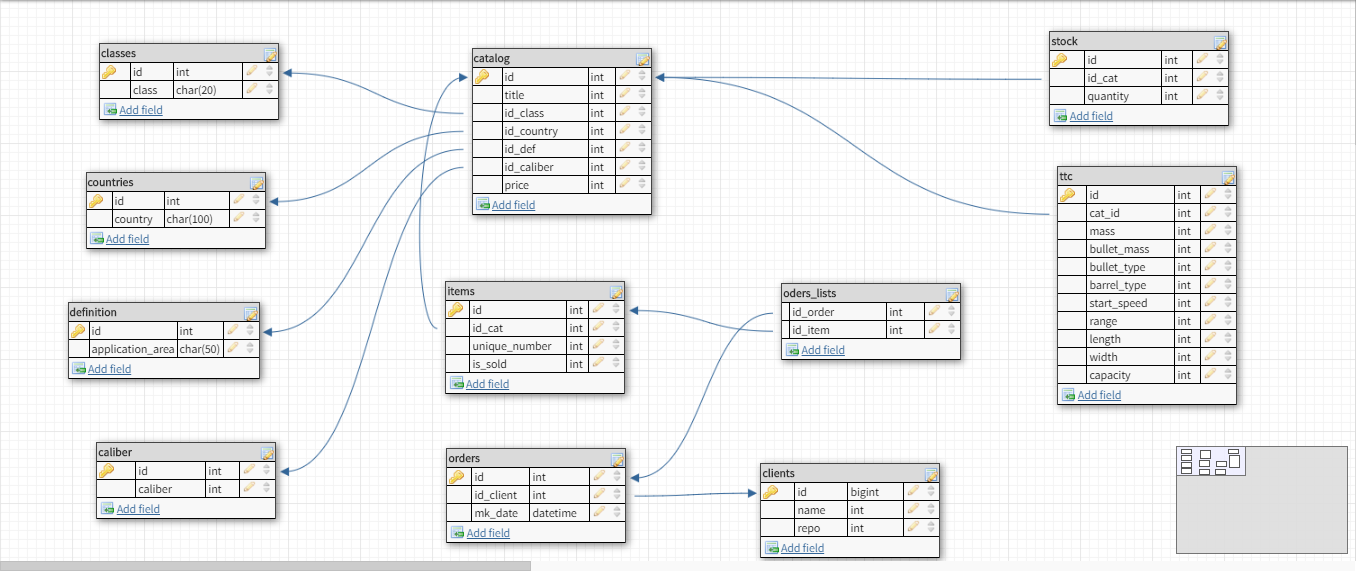
1. Привести справочник классов в виду: идентификатор - название класса.
2. Ввести учет клиентов и заказов.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | CREATE TABLE classes ( | |  | id int NOT NULL, | |  | class char(20) NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE clients ( | |  | id bigint NOT NULL, | |  | name int NOT NULL, | |  | repo int NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE orders ( | |  | id int NOT NULL, | |  | id\_client int NOT NULL, | |  | mk\_date DATETIME NOT NULL, | |  | PRIMARY KEY (id) | |  | ); | |  |  | |  | CREATE TABLE oders\_lists ( | |  | id\_order int NOT NULL, | |  | id\_item int NOT NULL | |  | ); | |  |  | |  | commit; | |  |  | |  | alter table catalog ALTER COLUMN id\_class int; | |  |  | |  | ALTER TABLE catalog ADD CONSTRAINT catalog\_fk0 FOREIGN KEY (id\_class) REFERENCES classes(id); | |  |  | |  | ALTER TABLE orders ADD CONSTRAINT orders\_fk0 FOREIGN KEY (id\_client) REFERENCES clients(id); | |  |  | |  | ALTER TABLE oders\_lists ADD CONSTRAINT oders\_lists\_fk0 FOREIGN KEY (id\_order) REFERENCES orders(id); | |  |  | |  | ALTER TABLE oders\_lists ADD CONSTRAINT oders\_lists\_fk1 FOREIGN KEY (id\_item) REFERENCES items(id); | |  |  | |  | commit; | |

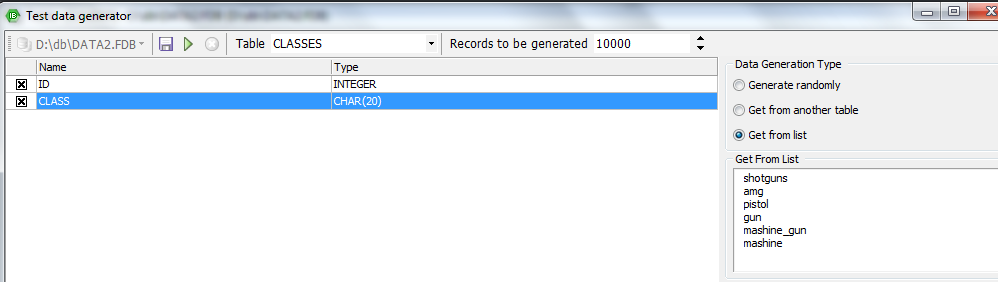
1. Создание ER-диаграммы:

С помощью IBExpert создадим ER-диаграмму:

Структура БД после модификации:



С помощью Test data generator сгенерировали данные большое количество записей(до 10 000) для таких таблиц, как сlasses, items, catalog, clients, orders\_list.



1. **Вывод**

В результате работы было проведено знакомство с языком SQL-DDL. С помощью данного языка были написаны скрипты для создания моей базы данных, для ее изменения и создания в ней некоторых данных.

Для изучения язык SQL (в частности DDL) прост, с помощью него достаточно просто управлять базой данных. Благодаря разработчикам СУБД Firebird и создателям языка SQL-DDL, администратору БД не нужно заботиться о целостности данных, отсутствие ошибок при частичном изменении данных и т.д. В некоторых случаях используются и другие языки работы с информацией (такие как NoSQL), но в нашем случае, используемый язык считаю наиболее удобным.

Далее с помощью IBExpert(GUI-оболочка для разработки и администрирования БД InterBase и Firebird) была сгенерирована ER-диаграмма БД, было произведено заполнение таблиц тестовыми данными, для дальнейшей работы (тестирования) с БД.