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 Веселов И.Н.

(подпись)

Руководитель Мяснов А.В.

(подпись)

“ ” 2015 г.

Санкт-Петербург

2015

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**

Язык SQL (Structured Query Language) - язык структурированных запросов. Он позволяет формировать весьма сложные запросы к базам данных. В SQL определены два подмножества языка:

1. SQL-DDL (Data Definition Language) - язык определения структур и ограничений целостности баз данных. Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.
2. SQL-DML (Data Manipulation Language) - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями.
3. **Ход работы**

Был создан скрипт, создающий базу данных в соответствии с новой схемой и модифицирующий ее в соответствии с заданием:

Модифицировать схему БД для удовлетворения следующим условиям:

1. Ввести учет сервиса.
2. Ввести учет комплектаций и доп. опций.

create table WORKS(ID int not null primary key, Work\_Type varchar(35) not null, Guarantee\_Days int not null);

create table SERVICE(Serv\_Code int not null primary key, Car\_Code int not null references CARS, Work\_Type int not null references WORKS, Date\_Start date not null, Day\_Work int not null, Price\_Serv int not null);

create table EQUIP(ID int not null primary key, Equip varchar(25) not null);

create table OPTIONS(ID int not null primary key, Option varchar(25) not null, Price\_Fact int not null);

create table ORD\_OPT(ID int not null primary key, Order\_Code int not null references ORDERS, Option\_Code int not null references OPTIONS, Price\_Ord int not null);

alter table MARKS drop ID;

commit;

alter table CARS drop ID;

commit;

alter table ORDERS drop ID;

commit;

alter table CLIENTS drop ID;

commit;

input 'C:\Firebird\NEW\Works.txt';

commit;

input 'C:\Firebird\NEW\Service.txt';

commit;

input 'C:\Firebird\NEW\Equip.txt';

commit;

input 'C:\Firebird\NEW\Options.txt';

commit;

input 'C:\Firebird\NEW\Ord\_Opt.txt';

commit;

alter table CARS add Equip\_Code int not null references EQUIP;

commit;

update CARS set EQUIP\_CODE = 1 where CAR\_CODE = 1002;

update CARS set EQUIP\_CODE = 4 where CAR\_CODE = 1003;

update CARS set EQUIP\_CODE = 2 where CAR\_CODE = 1001;

update CARS set EQUIP\_CODE = 1 where CAR\_CODE = 1004;

update CARS set EQUIP\_CODE = 5 where CAR\_CODE = 1006;

update CARS set EQUIP\_CODE = 3 where CAR\_CODE = 1005;

commit;

Был создан скрипт, заполняющий БД данными:

insert into EQUIP values('1', 'Classic');

insert into EQUIP values('2', 'Comfort');

insert into EQUIP values('3', 'Elegance');

insert into EQUIP values('4', 'Avantgarde');

insert into EQUIP values('5', 'Luxury');

commit;

insert into OPTIONS values('1', 'Additional airbags', '200');

insert into OPTIONS values('2', 'Spoiler', '400');

insert into OPTIONS values('3', 'Alloy RIM', '250');

insert into OPTIONS values('4', 'Heated seats', '300');

insert into OPTIONS values('5', 'Heated steering wheel', '100');

insert into OPTIONS values('6', 'Seat ventilation', '250');

insert into OPTIONS values('7', 'Moldings', '150');

insert into OPTIONS values('8', 'Rails', '275');

commit;

insert into ORD\_OPT values('1', '23', '2', '500');

insert into ORD\_OPT values('2', '23', '6', '300');

insert into ORD\_OPT values('3', '23', '8', '250');

insert into ORD\_OPT values('4', '12', '4', '350');

insert into ORD\_OPT values('5', '12', '7', '175');

insert into ORD\_OPT values('6', '76', '6', '200');

insert into ORD\_OPT values('7', '76', '5', '100');

insert into ORD\_OPT values('8', '76', '3', '200');

insert into ORD\_OPT values('9', '54', '1', '150');

commit;

insert into SERVICE values('1', '1003', '9', '13.10.2015', '1', '100');

insert into SERVICE values('2', '1005', '6', '02.08.2015', '3', '400');

insert into SERVICE values('3', '1001', '5', '27.03.2015', '10', '850');

insert into SERVICE values('4', '1002', '3', '10.09.2015', '2', '300');

commit;

insert into WORKS values('1', 'Maintenance', '14');

insert into WORKS values('2', 'Wheel adjustment', '7');

insert into WORKS values('3', 'Diagnosis and electricy', '21');

insert into WORKS values('4', 'Transmission repair', '60');

insert into WORKS values('5', 'Engine repair', '180');

insert into WORKS values('6', 'Car painting', '30');

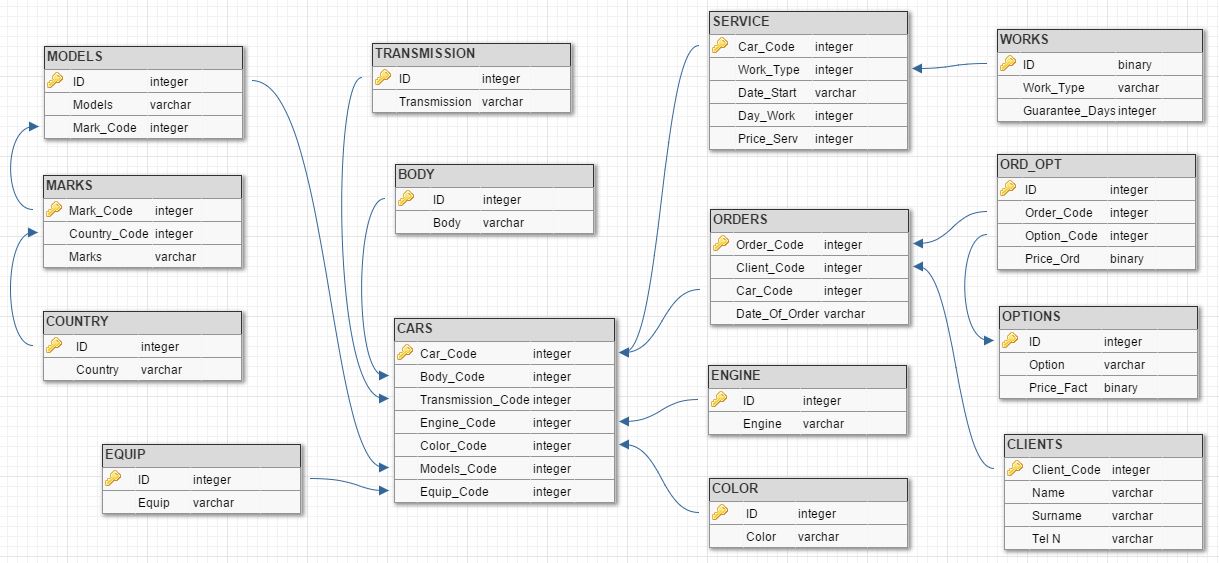
insert into WORKS values('7', 'Body repair', '45');

insert into WORKS values('8', 'Suspension repair', '30');

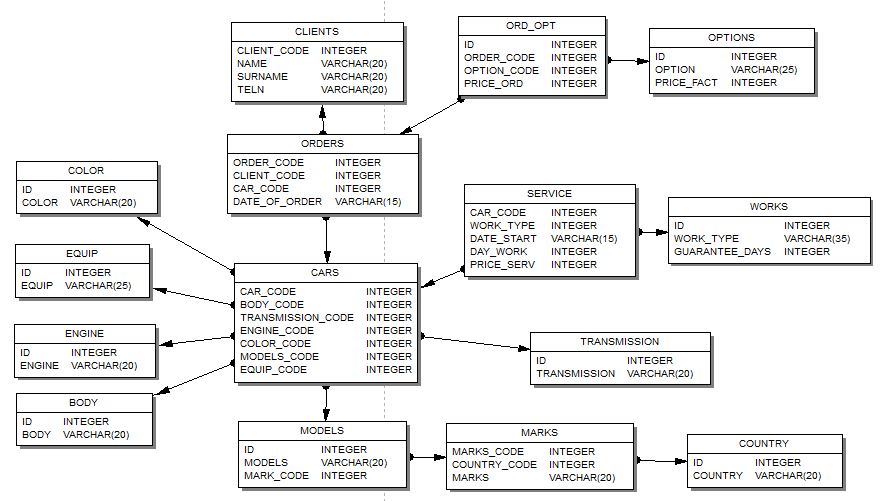
insert into WORKS values('9', 'Сhip tuning', '20');

commit;

Структура БД после модификации:



С помощью Database Designer в IBExpert была сгенерирована ER-диаграмма модифицированной БД:



С помощью Test data generator сгенерировали данные для нескольких таблиц: CARS, CLIENTS, ORDERS, SERVICE.

1. **Вывод**

Язык DDL имеет свойство немедленного выполнения (выполняется команда <commit>), то есть невозможно вернуть изменение обратно. Также присутствуют команды удаления, что предполагает особое внимание при работе с данным языком.

DDL не работает с данными, а работает с объектами базы данных, поэтому возникает необходимость использования других методов работы с данными.

При использовании данного языка упрощается поддержание целостности уже существующей базы.

При отсутствии языка DDL сильно возросла бы возможность неполного обновления данных. Это бы привело к возникновению множества ошибок.