Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина**: Базы данных

**Тема**: SQL-DDL

Выполнил студент гр. 43501/3 Степанов Д.С.

(подпись)

Руководитель Мяснов А.В.

(подпись)

“\_ ” 2015 г.

Санкт - Петербург

2015

**Цель работы:**

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

**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)

**Выполнение работы**

Data Definition Language (DDL) — язык описания данных. С помощью этого подмножества языка создаются, модифицируются и удаляются объекты базы данных.

Скрипт, создающий базу данных, имеет следующий вид:

create table AutoAdv (

AutoID int NOT NULL,

Price int NOT NULL,

ProdYear int NOT NULL,

MileAge int NOT NULL,

NumHolders int NOT NULL,

HolderID int REFERENCES HolderInfo(HolderID),

ModelID int REFERENCES ModelInfo(ModelID),

Color varchar(10) NOT NULL,

EquipmentId int REFERENCES EquipmentInf(EquipmentID),

PRIMARY KEY (AutoID)

);

create table HolderInfo (

HolderID int NOT NULL,

FirstName varchar(20) NOT NULL,

SecondName varchar(20) NOT NULL,

TelNumber varchar(20) NOT NULL,

Adress varchar(100) NOT NULL,

PRIMARY KEY (HolderID)

);

CREATE TABLE EquipmentInf (

EquipmentID int NOT NULL,

SaloonId int REFERENCES Saloons(SaloonID),

Breaks varchar(1) NOT NULL,

Gaz varchar(1) NOT NULL,

LightSensor varchar(1) NOT NULL,

ESP varchar(1) NOT NULL,

WheelsId int REFERENCES Wheels(WheelID),

PRIMARY KEY (EquipmentID)

);

CREATE TABLE Saloons (

SaloonID int NOT NULL,

Quality varchar(10) NOT NULL,

Material varchar(20) NOT NULL,

PRIMARY KEY (SaloonID)

);

CREATE TABLE Wheels (

WheelID int NOT NULL,

Model varchar(15) NOT NULL,

Diameter int NOT NULL,

PRIMARY KEY (WheelID)

);

CREATE TABLE FactoryInfo (

FactoryId int NOT NULL,

Country varchar(20) NOT NULL,

FactoryName varchar(20) NOT NULL,

PRIMARY KEY (FactoryId)

);

CREATE TABLE EngineInfo (

EngineID int NOT NULL,

ModelID int REFERENCES ModelInfo(ModelID),

Power int NOT NULL,

Volume\_l int NOT NULL,

EngineScheme varchar(20) NOT NULL,

PRIMARY KEY (EngineID)

);

CREATE TABLE ModelInfo (

ModelID int NOT NULL,

ModelName varchar(20) NOT NULL,

FactoryID int REFERENCES FactoryInfo(FactoryId),

Yea int NOT NULL,

PRIMARY KEY (ModelID)

);

create domain FuelSort AS

varchar(10) CHECK (VALUE IN ('Gaz', 'Benzin','Disel','Electro'));

create domain GearSort AS

varchar(4) CHECK (VALUE IN ('Hand', 'Avto'));

CREATE TABLE EquipmentInform(

EquipmentInformID int PRIMARY KEY,

EquipmentID int,

EqInf int REFERENCES EquipmentInf(EquipmentID));

alter table AutoAdv add EquipmentId int REFERENCES EquipmentInform(EquipmentID);

create table AdvInfo(

AdvInfoID int PRIMARY KEY,

isFree varchar(1),

Since date,

Untill date);

alter table AutoAdv add AdvInfoI int REFERENCES AdvInfo(AdvInfoID);

Для работы с БД необходимо после создания заполнить её данными, для этого был написал следующий скрипт, который записывает несколько значений во все таблицы базы данных:

insert into advinfo values(1,'N','2007-12-31','2008-01-1');

insert into advinfo(advinfoid,isfree) values(2,'Y');

insert into holderinfo values(1,'Danil','Stepanov','911','Railway Station');

insert into holderinfo values(2,'Toha','Tigr','09','Maiami Nature');

insert into wheels values(1,'Shtampy',14);

insert into wheels values(2,'BBS',128);

insert into saloons values(1,'Good','Skin');

insert into saloons values(2,'Low','Bolshit');

insert into equipmentinf values(1,1,'Y','N','N','N',2);

insert into equipmentinf values(2,1,'Y','N','Y','Y',1);

insert into factoryinfo values(1,'Russia','Lada');

insert into factoryinfo values(2,'Germany','Toyota');

insert into modelinfo values(1,'2112',1,2001);

insert into modelinfo values(2,'X5 AMG',2,2010);

insert into engineinfo values(1,1,500,2,'V6','Benzin','Hand');

insert into engineinfo values(2,2,350,3,'V12','Gaz','Avto');

insert into equipmentinform values(1,1,1);

insert into equipmentinform values(2,1,2);

insert into autoadv values(1,100000,2003,120000,2,1,1,'Green',1,1);

insert into autoadv values(2,1000000,2013,5000,3,2,2,'Black',2,2);

Индивидуальное задание – SQL-запрос, изменяющий структуру базы данных имеет следующий вид:

  1. Реализовать хранение дополнительных характеристик без необходимости  
 модификации схемы БД.  
  2. Ввести учет возможности публикации платных объявлений.

Для выполнения данного задания был написан скрипт:

alter table autoadv drop (equipmentid);

drop table equipmentinform;

alter table equipmentinf drop breaks,gaz,lightsensor,esp;

create table features (fe\_id int primary key, name VARCHAR(20) not null);

create table linkEquipment (id int primary key,equip\_id int,fea\_id int REFERENCES features(fe\_id));

alter table equipmentinf add fea\_id int references linkEquipment(equip\_id);

alter table autoadv add equip\_id int references equipmentinf(equipmentinf.equipmentid);

create table AdvInfo(

AdvInfoID int PRIMARY KEY,

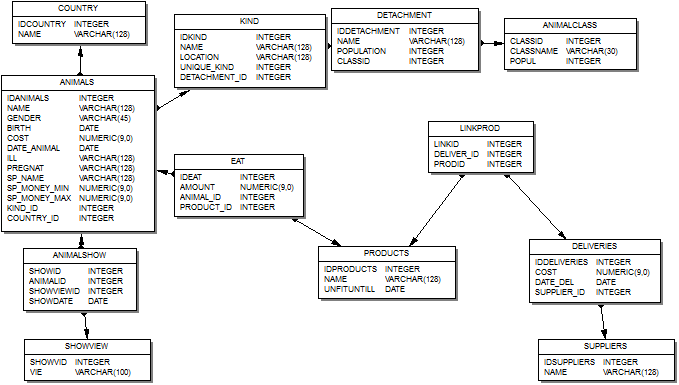
isFree varchar(1),

Since int,

Untill int);

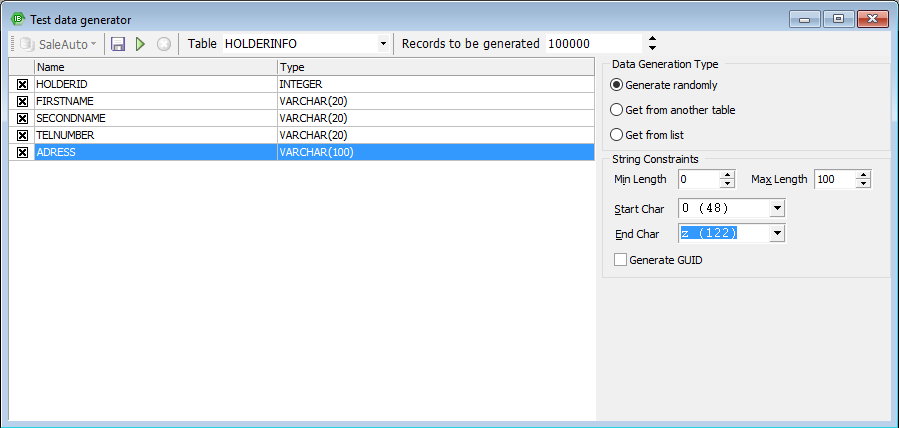
alter table AutoAdv add AdvInfoI int REFERENCES AdvInfo(AdvInfoID);

После изменения структуры данных была получена её блок-схема с помощью встроенного средства IBExpert Database Designer:

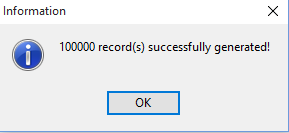


Генерировать данные с помощью IBExpert будем 2-мя способами: скриптом и с помощью встроенного средства Test Data Generator.

Test Data Generator предоставляет большой инструментарий для генерации тестовых данных для таблиц. Сгенерируем данные для таблицы HolderInfo:



При успешном завершении генерация выдаётся соответствующее сообщение:



С помощью скрипта можно более гибко генерировать данные. Например сгенерируем данные для таблицы, содержащей информацию о том, платное или бесплатное объявление. Если оно платное, то будут заполнены поля с датой оплаты и конца действия, в противном случае они будут заполнены нулями:

EXECUTE BLOCK

AS

DECLARE i INTEGER = 2;

BEGIN

WHILE (i < 1000) DO

BEGIN

if(MOD(i,2) = 0) then

begin

INSERT INTO advinfo values(:i,'Y',current\_timestamp - 250 - TRUNC(RAND() \* 200),current\_timestamp - TRUNC(RAND() \* 250));

end

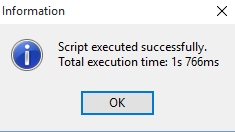
else insert into advinfo (advinfo.advinfoid,advinfo.isfree) values(:i,'N');

i = i + 1;

END

END

100 тысяч строк были сгенерированы за 1,8с.



Также скриптом заполним таблицу, соединяющую комплектацию и дополнительные опции:

EXECUTE BLOCK

AS

DECLARE i INTEGER = 3;

DECLARE j INTEGER;

DECLARE k integer = 4;

BEGIN

WHILE (i < 100) DO

BEGIN

j = TRUNC(RAND() \* 9);

while (j>0) do

begin

insert into linkequipment values(:k,:j,:i);

k=k+1;

j=j-1;

end

i = i + 1;

END

END

**Вывод**

В данной работе мы познакомились с операторами DDL, изучили средство работы с базами данных IBExpert, научились писать скрипты, создающие таблицы и заполняющие их данными.

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.

Запрос «create» используется для создания базы данных, таблицы, индекса, представления или хранимой процедуры. Запрос «alter» используется для изменения существующего объекта базы данных (таблицы, индекса, представления или хранимой процедуры) или самой базы данных. Запрос «drop» используется для удаления существующего объекта базы данных (таблицы, индекса, представления или хранимой процедуры) или самой базы данных. И наконец, в DDL существуют понятия первичного и внешнего ключа, которые осуществляют соблюдение целостности данных. Команды "первичный ключ" primary key, "внешний ключ" foreign key включаются в запросы «create table", «alter table».

Для создания БД были применены операторы DDL create table, alter table. Для хранения дат был использован тип данных date. Для задания значений из определенного списка был использован тип данных domain.