

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе №3
по дисциплине «Базы данных»
Язык SQL-DDL

Выполнил

студент гр. 43501/3

Е.А. Никитин

Преподаватель

А.В. Мяснов

«__»_____2015г.

Санкт-Петербург

2015

Цели работы

Познакомить студентов 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)

Ход работы:

1)

SQL — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД).

В SQL определены два подмножества языка:

- **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.
- **SQL-DML** (Data Manipulation Language) - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

2) Был создан скрипт для создания базы данных в соответствии со схемой:

```
create database 'D:\univer\4\BD\1\ticket_base.fdb' user 'SYSDBA' password 'masterkey';  
commit;
```

```
connect 'D:\univer\4\BD\1\ticket_base.fdb' user 'SYSDBA' password 'masterkey';  
create table train(train_id INT PRIMARY KEY,  
                  type char(255) NOT NULL,  
                  carriages INT NOT NULL);
```

```
create table station(st_id INT PRIMARY KEY,  
                    name char(255) NOT NULL);
```

```
create table passenger(pas_id INT PRIMARY KEY,
```

```

        name char(255) NOT NULL,
        surname char(255) NOT NULL,
        document INT NOT NULL,
        doc_num INT NOT NULL);

alter table passenger
alter column document type char(255);

create table place(place_id INT PRIMARY KEY,
                  train_id INT REFERENCES train(train_id) NOT NULL);
alter table place add num_in_car INT NOT NULL;
alter table place add car_type char(255) NOT NULL;
alter table place add pl_pos char(255) NOT NULL;
alter table ticket add start_st char(255) NOT NULL;
alter table ticket add stop_st char(255) NOT NULL;

create table ticket(t_num INT NOT NULL,
                  place_id INT REFERENCES place(place_id) NOT NULL,
                  pas_id INT REFERENCES passenger(pas_id) NOT NULL,
                  status char(255) NOT NULL,
                  start_st char(255),
                  stop_st char(255));

create table shedule(route_id INT PRIMARY KEY,
                  train_id INT REFERENCES train(train_id) NOT NULL,
                  name char(255) NOT NULL,
                  r_date char(255) NOT NULL);
create table route_station(route_id INT REFERENCES shedule(route_id) NOT NULL,
                  st_id INT REFERENCES station(st_id) NOT NULL,
                  st_func NOT NULL,
                  stop_time char(255) NOT NULL,
                  num_in_route INT NOT NULL);

commit;
```

3) Был создан скрипт, заполняющий БД данными:

```

insert into passenger (pas_id, name, surname, document, doc_num)
values (740, 'Ivan', 'Ivanov', 'passport', 9408255255);

select pas_id from passenger;
```

```
insert into station(st_id, type) values(1, 'MOSCOW');
insert into station(st_id, type) values(2, 'SPB');
insert into station(st_id, type) values(3, 'N_Novgorod');
insert into station(st_id, type) values(4, 'Vladimir');
insert into station(st_id, type) values(5, 'Kazan');
```

```
insert into passenger(pas_id, name, surname, document, doc_num) values(35,
'Ivan', 'Ivanov', 'passport', '9408255255');
insert into passenger(pas_id, name, surname, document, doc_num) values(36,
'Petr', 'Petrov', 'passport', '9408244244');
insert into passenger(pas_id, name, surname, document, doc_num) values(37,
'Alexander', 'Belov', 'militaryID', '45286513');
insert into passenger(pas_id, name, surname, document, doc_num) values(38,
'Vladimir', 'Krasnov', 'passport', '9410235235');
insert into passenger(pas_id, name, surname, document, doc_num) values(39,
'Dmitriy', 'Chernov', 'passport', '9411288288');
```

```
insert into train(train_id, type, carriages) values(75, '2EL5', 15);
insert into train(train_id, type, carriages) values(76, '2EL5', 12);
insert into train(train_id, type, carriages) values(77, 'KZ4A', 10);
insert into train(train_id, type, carriages) values(78, 'VL40U', 13);
insert into train(train_id, type, carriages) values(79, 'E5K', 11);
```

```
insert into place(place_id, train_id, carriage, num_in_car, car_type, pl_po
s) values(25, 75, 1, 25, 'plac', 'low');
insert into place(place_id, train_id, carriage, num_in_car, car_type, pl_po
s) values(111, 76, 3, 3, 'plac', 'low');
insert into place(place_id, train_id, carriage, num_in_car, car_type, pl_po
s) values(128, 77, 3, 20, 'plac', 'high');
insert into place(place_id, train_id, carriage, num_in_car, car_type, pl_po
s) values(36, 78, 1, 36, 'coupe', 'high');
insert into place(place_id, train_id, carriage, num_in_car, car_type, pl_po
s) values(244, 79, 5, 12, 'coupe', 'high');
```

```
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st) valu
es(28, 25, 35, 'buy', 'Moscow', 'SPB');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st) valu
es(35, 128, 37, 'buy', 'Moscow', 'N_Novgorod');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st) valu
es(44, 244, 39, 'buy', 'SPB', 'Kazan');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st) valu
```

```

es(75, 36, 38, 'wait', 'Vladimir', 'SPB');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st) valu
es(59, 111, 36, 'wait', 'Moscow', 'Vladimir');

```

```

insert into shedule(route_id, train_id, name, r_date)
values(29, 75, '722A Moscow-SPB', '12:15 24.12.2015');
insert into shedule(route_id, train_id, name, r_date)
values(125, 77, '138 Moscow-N_Novgorod', '23:01 25.12.2015');
insert into shedule(route_id, train_id, name, r_date)
values(158, 79, '592 SPB-Kazan', '16:53 26.12.2015');
insert into shedule(route_id, train_id, name, r_date)
values(232, 78, '235 Vladimir-SPB', '01:33 24.12.2015');
insert into shedule(route_id, train_id, name, r_date)
values(38, 76, '199 Moscow-Vladimir', '10:15 23.12.2015');

```

```

insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(1, 29, 'start', '-', 1);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(2, 29, 'finish', '-', 4);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(2, 158, 'start', '-', 1);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(5, 158, 'finish', '-', 29);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(1, 125, 'start', '-', 1);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(3, 125, 'finish', '-', 11);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(4, 232, 'start', '-', 1);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(2, 232, 'finish', '-', 12);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(1, 38, 'start', '-', 1);
insert into route_station(st_id, route_id, st_func, stop_time, num_in_route)
values(4, 38, 'finish', '-', 17);
commit;

```

4) Был создан скрипт изменяющий

```
drop table ticket;
create table ticket(t_num INT PRIMARY key,
    place_id INT REFERENCES place(place_id) NOT NULL,
    pas_id INT REFERENCES passenger(pas_id) NOT NULL,
    status char(255) NOT NULL,
    start_st char(255),
    stop_st char(255));
alter table ticket drop pas_id;
```

```
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st)
values(28, 25, 35, 'buy', 'Moscow', 'SPB');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st)
values(35, 128, 37, 'buy', 'Moscow', 'N_Novgorod');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st)
values(44, 244, 39, 'buy', 'SPB', 'Kazan');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st)
values(75, 36, 38, 'wait', 'Vladimir', 'SPB');
insert into ticket(t_num, place_id, pas_id, status, start_st, stop_st)
values(59, 111, 36, 'wait', 'Moscow', 'Vladimir');
```

```
create table t_order(order_id INT PRIMARY key,
    t_num INT references ticket(t_num),
    tickets_num INT NOT NULL,
    UNIQUE(order_id, t_num));
```

```
create table orders(pas_id INT references passenger(pas_id),
    order_id INT references t_order(order_id));
```

```
create table car_type(car_type_id INT PRIMARY KEY,
    name char(255));
create table place_type(place_type_id INT PRIMARY KEY,
    name char(255));
```

```
insert into car_type(car_type_id, name)
values(0, 'plac');
insert into car_type(car_type_id, name)
values(1, 'coupe');
insert into place_type(place_type_id, name)
values(0, 'high');
```

```

insert into place_type(place_type_id, name)
values(1, 'low');
insert into place_type(place_type_id, name)
values(2, 'high side');
insert into place_type(place_type_id, name)
values(3, 'low side');

```

```

alter table place drop car_type;
alter table place drop pl_pos;

```

```

alter table place add car_type_id INT NOT NULL;
alter table place
add constraint place_fk1
foreign key (car_type_id) REFERENCES car_type(car_type_id);

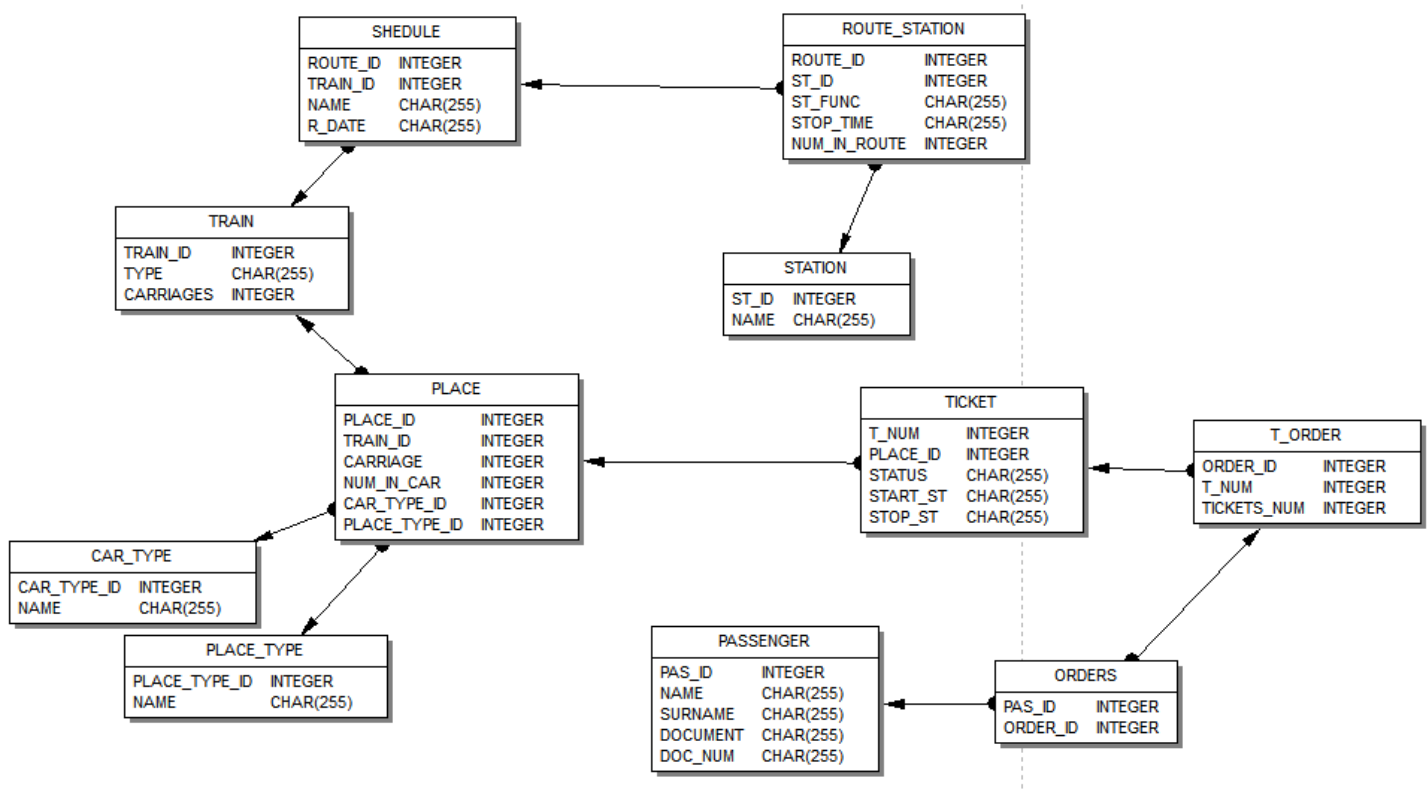
```

```

alter table place add place_type_id INT NOT NULL;
alter table place
add constraint place_fk2
foreign key (place_type_id) REFERENCES place_type(place_type_id);
commit;

```

5) Полученная диаграмма из IBExpert:



- б) Были сгенерированы данные (по 100000) в таблицах Passenger, Station и Train. Записи создавались в утилите Test Data Generator.

Вывод.

В ходе данной работы было ознакомление с языком SQL-DDL, который является инструментом для определения структур и ограничений целостности баз данных. Включает в себя команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями. Написание скриптов выполнялось как в консоли Firebird, так и в программе IBEExpert, которая служит для создания, визуализации (была получена ER-диаграмма) и управления таблицами, а также имеет множество других инструментов, например тестовое заполнение таблиц большими объемами данных (что было использовано в ходе лабораторной).