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:

**SQL** — [формальный](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной [реляционной базе данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), управляемой соответствующей системой управления базами данных ([СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94)).

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

1. 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;

1. 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;

1. 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;

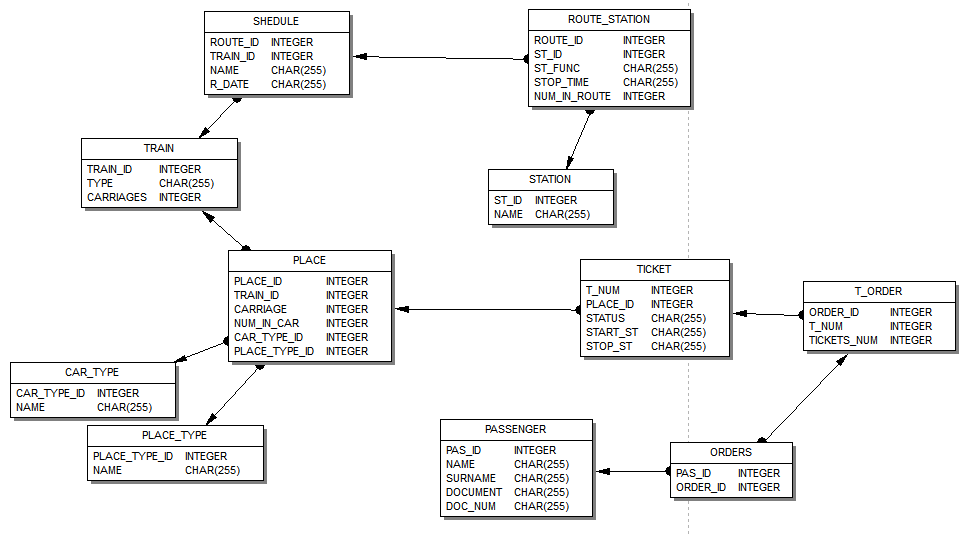
alter table place

add constraint place\_fk2

foreign key (place\_type\_id) REFERENCES place\_type(place\_type\_id);

commit;

1. Полученная диаграмма из IBExpert:



1. 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 IBExpert, 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).