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 М.А. Иконников

(подпись)

Руководитель А.В. Мяснов

(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:

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. **Ход работы**

Был создан скрипт, создающий базу данных в соответствии со схемой:

connect 'C:\SCLUB'

user 'SYSDBA' password 'masterkey';

drop database;

create database 'C:\SCLUB' user 'SYSDBA' password 'masterkey';

connect 'C:\SCLUB' user 'SYSDBA' password 'masterkey';

create table SeasonTickets(

ID\_TICKET int primary key

,ID\_TYPE int not null

,Period date

);

create table Clients(

ID\_CLIENT int primary key

,ClientName varchar(255)

,TelNumber int

,ID\_TICKET int not null

,NumOfClasses int

,ID\_DISCOUNT int

);

create table TicketTypes(

ID\_TYPE int primary key

,TypeName varchar(255)

,Price int

);

create table Accounting(

ID\_CLIENT int not null

,ID\_DISCOUNT int

,Paid int

);

create table Discounts(

ID\_DISCOUNT int primary key

,DiscountName varchar(255)

,DiscountValue int

);

create table Rooms(

ID\_ROOM int primary key

,RoomName varchar(255)

);

create table Sections(

ID\_SECTION int primary key

,SectionName varchar(255)

,ID\_ROOM int not null

);

create table Trainers(

ID\_TRAINER int primary key

,TrainerName varchar(255)

,TrainerExp int

,ID\_SECTION int not null

);

create table TicketTypesToSections(

ID\_TYPE int not null

,ID\_SECTION int not null

);

alter table SeasonTickets add constraint tickets\_to\_types foreign key (ID\_TYPE)

references TicketTypes(ID\_TYPE);

alter table Clients add constraint clients\_to\_tickets foreign key (ID\_TICKET)

references SeasonTickets(ID\_TICKET);

alter table Accounting add constraint accounting\_to\_clients foreign key (ID\_CLIENT)

references Clients(ID\_CLIENT);

alter table Accounting add constraint accounting\_to\_discounts foreign key (ID\_DISCOUNT)

references Discounts(ID\_DISCOUNT);

alter table Sections add constraint sections\_to\_rooms foreign key (ID\_ROOM)

references Rooms(ID\_ROOM);

alter table Trainers add constraint trainers\_to\_sections foreign key (ID\_SECTION)

references Sections(ID\_SECTION);

alter table TicketTypesToSections add constraint ttts\_to\_types foreign key (ID\_TYPE)

references TicketTypes(ID\_TYPE);

alter table TicketTypesToSections add constraint ttts\_to\_sections foreign key (ID\_SECTION)

references Sections(ID\_SECTION);

Был создан скрипт, заполняющий БД данными:

connect 'C:\SCLUB' user 'SYSDBA' password 'masterkey';

insert into TICKETTYPES (ID\_TYPE, TYPENAME, PRICE) values (1, 'UNLIM', 10000);

insert into TICKETTYPES (ID\_TYPE, TYPENAME, PRICE) values (2, 'FOOTBALL', 2000);

insert into TICKETTYPES (ID\_TYPE, TYPENAME, PRICE) values (3, 'BASKETBALL', 2000);

insert into TICKETTYPES (ID\_TYPE, TYPENAME, PRICE) values (4, 'POWERLIFTING', 2000);

insert into TICKETTYPES (ID\_TYPE, TYPENAME, PRICE) values (5, 'BODYBUILDING', 2000);

insert into TICKETTYPES (ID\_TYPE, TYPENAME, PRICE) values (6, 'FOOTBAL+GYM', 3500);

insert into TICKETTYPES (ID\_TYPE, TYPENAME, PRICE) values (7, 'TENNIS', 5000);

insert into SEASONTICKETS (ID\_TICKET, ID\_TYPE, PERIOD) values (1, 1, '2015-12-12');

insert into SEASONTICKETS (ID\_TICKET, ID\_TYPE, PERIOD) values (2, 1, '2015-10-05');

insert into SEASONTICKETS (ID\_TICKET, ID\_TYPE, PERIOD) values (3, 2, '2015-09-10');

insert into SEASONTICKETS (ID\_TICKET, ID\_TYPE, PERIOD) values (4, 3, '2015-09-30');

insert into SEASONTICKETS (ID\_TICKET, ID\_TYPE, PERIOD) values (5, 4, '2015-08-11');

insert into CLIENTS (ID\_CLIENT, CLIENTNAME, TELNUMBER, NUMOFCLASSES, ID\_DISCOUNT)

values(1, 'Ilya Bulatov', 666666, 5, null);

insert into CLIENTS (ID\_CLIENT, CLIENTNAME, TELNUMBER, NUMOFCLASSES, ID\_DISCOUNT)

values(2, 'Vladimir Putin', 777777, 7, 1);

insert into CLIENTS (ID\_CLIENT, CLIENTNAME, TELNUMBER, NUMOFCLASSES, ID\_DISCOUNT)

values(3, 'Artem Dzyuba', 123456, 50, 1);

insert into CLIENTS (ID\_CLIENT, CLIENTNAME, TELNUMBER, NUMOFCLASSES, ID\_DISCOUNT)

values(4, 'Megan Fox', 321321, 2, null);

insert into ROOMS (ID\_ROOM, ROOMNAME) values (1, 'FOOTBALLROOM');

insert into ROOMS (ID\_ROOM, ROOMNAME) values (2, 'TENNISCOURT');

insert into ROOMS (ID\_ROOM, ROOMNAME) values (3, 'GYM');

insert into ROOMS (ID\_ROOM, ROOMNAME) values (4, 'BASKETBALLROOM');

insert into SECTIONS (ID\_SECTION, SECTIONNAME, ID\_ROOM) values (1, 'FOOTBALL', 1);

insert into SECTIONS (ID\_SECTION, SECTIONNAME, ID\_ROOM) values (2, 'TENNIS', 2);

insert into SECTIONS (ID\_SECTION, SECTIONNAME, ID\_ROOM) values (3, 'BASKETBALL', 4);

insert into SECTIONS (ID\_SECTION, SECTIONNAME, ID\_ROOM) values (4, 'POWERLIFTING', 3);

insert into SECTIONS (ID\_SECTION, SECTIONNAME, ID\_ROOM) values (5, 'BODYBUILDING', 3);

insert into DISCOUNTS (ID\_DISCOUNT, DISCOUNTNAME, DISCOUNTVALUE)

values (1, 'Students', 10);

insert into DISCOUNTS (ID\_DISCOUNT, DISCOUNTNAME, DISCOUNTVALUE)

values (2, 'Pensioners', 10);

insert into DISCOUNTS (ID\_DISCOUNT, DISCOUNTNAME, DISCOUNTVALUE)

values (3, 'VIP', 25);

insert into ACCOUNTING (ID\_CLIENT, ID\_DISCOUNT, PAID, ID\_TICKET) values (1, 2, 10000, 1);

insert into ACCOUNTING (ID\_CLIENT, ID\_DISCOUNT, PAID, ID\_TICKET) values (2, 1, 10000, 2);

insert into ACCOUNTING (ID\_CLIENT, ID\_DISCOUNT, PAID, ID\_TICKET) values (3, 3, 10000, 3);

insert into ACCOUNTING (ID\_CLIENT, ID\_DISCOUNT, PAID, ID\_TICKET) values (4, 3, 10000, 4);

insert into TRAINERS (ID\_TRAINER, TRAINERNAME, TRAINEREXP, ID\_SECTION)

values (1, 'Leonid Slutsky', 9, 1);

insert into TRAINERS (ID\_TRAINER, TRAINERNAME, TRAINEREXP, ID\_SECTION)

values (2, 'Maria Sharapova', 10, 2);

insert into TRAINERS (ID\_TRAINER, TRAINERNAME, TRAINEREXP, ID\_SECTION)

values (3, 'Ivan Ivanov', 9, 3);

insert into TRAINERS (ID\_TRAINER, TRAINERNAME, TRAINEREXP, ID\_SECTION)

values (4, 'Maxim Ikonnikov', 20, 4);

insert into TRAINERS (ID\_TRAINER, TRAINERNAME, TRAINEREXP, ID\_SECTION)

values (5, 'Misha Skalyga', 2, 4);

insert into TRAINERS (ID\_TRAINER, TRAINERNAME, TRAINEREXP, ID\_SECTION)

values (6, 'Jay Cutler', 5, 5);

insert into TICKETTYPESTOSECTIONS (ID\_TYPE, ID\_SECTION) values (2, 1);

insert into TICKETTYPESTOSECTIONS (ID\_TYPE, ID\_SECTION) values (7, 2);

insert into TICKETTYPESTOSECTIONS (ID\_TYPE, ID\_SECTION) values (3, 3);

insert into TICKETTYPESTOSECTIONS (ID\_TYPE, ID\_SECTION) values (4, 4);

insert into TICKETTYPESTOSECTIONS (ID\_TYPE, ID\_SECTION) values (5, 5);

Затем был создан скрипт, модифицирующий базу данных согласно заданию:

*Модифицировать схему БД для удовлетворения следующим требованиям:*

1. *Реализовать учет покупок абонементов, возможность владения несколькими абонементами.*
2. *Реализовать учет посещений, дополнительных услуг, которые могут быть связаны с определенными типами абонементов или продаваться отдельно. Ввести учет продаж дополнительных услуг.*

connect 'C:\SCLUB'

user 'SYSDBA' password 'masterkey';

alter table Clients drop ID\_TICKET;

alter table SeasonTickets add ID\_CLIENT int not null;

alter table SeasonTickets add constraint tickets\_to\_clients foreign key (ID\_CLIENT)

references Clients (ID\_CLIENT);

drop table Accounting;

create table Visits(

ID\_CLIENT int not null

,VisitDate date

);

alter table Visits add constraint visits\_to\_clients foreign key (ID\_CLIENT)

references Clients (ID\_CLIENT);

create table AddServices(

ID\_SERVICE int primary key

,ServiceName varchar(255)

,ServPrice int

);

create table ServicesToTypes(

ID\_SERVICE int

,ID\_TYPE int

);

alter table ServicesToTypes add constraint stt\_to\_serv foreign key (ID\_SERVICE)

references AddServices (ID\_SERVICE);

alter table ServicesToTypes add constraint stt\_to\_types foreign key (ID\_TYPE)

references TicketTypes (ID\_TYPE);

create table AccountingServices(

ID\_CLIENT int not null

,ID\_SERVICE int not null

,BuyServDate date

);

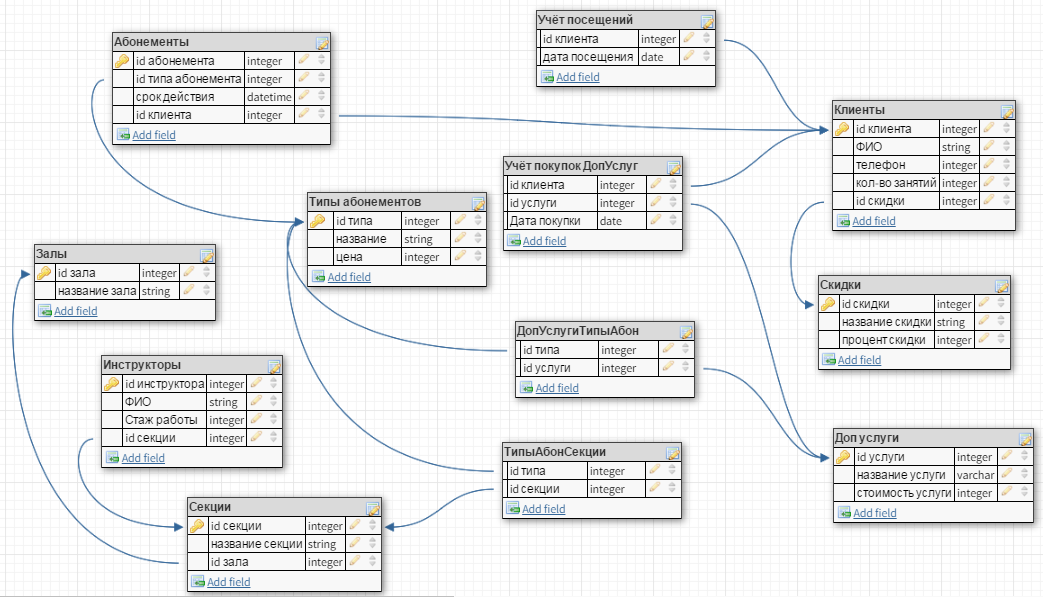
alter table AccountingServices add constraint accserv\_to\_clients foreign key (ID\_CLIENT)

references Clients(ID\_CLIENT);

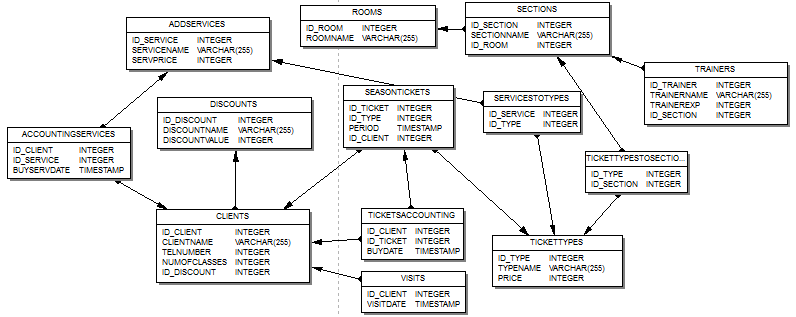
alter table AccountingServices add constraint accserv\_to\_services foreign key (ID\_SERVICE)

references AddServices(ID\_SERVICE);

Структура БД после модификации:



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



С помощью Test data generator сгенерировали данные (100 000 записей) для 3-х таблиц (Clients, SeasonTickets, Trainers).

1. **Вывод**

В результате работы было проведено знакомство и изучение языка SQL-DDL. Также были написаны скрипты для создания таблиц в БД, для заполнения их данными и для модификации БД по заданию преподавателя.

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.

Использовались такие инструменты языка DDL как <create> (создать), <alter> (изменить), <drop> (удалить). После выполнения данных команд автоматически выполняется команда <commit>, т.е. выполненные изменения уже нельзя отменить, поэтому следует быть предельно внимательным при работе с данным языком. Также можно отметить что тексты SQL-запросов, написанные на SQL DDL могут быть достаточно легко перенесены из одной СУБД в другую.

С помощью IBExpert была сгенерирована ER-диаграмма нашей БД, было произведено заполнение таблиц случайными данными.

При выполнении работы проблем не возникло.