Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра Компьютерных систем и программных технологий

Отчет по лабораторной работе 4

**«Язык SQL - DML»**

по дисциплине «Базы данных»

Работу выполнил:

студент гр. 43501/3

Родина В.В.

Руководитель:

Мяснов А.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2016 г

Санкт-Петербург

2016

1. **Цель работы**

Познакомиться с основами проектирования схемы БД, языком создания запросов управления данными БД SQL-DML.

1. **Программа работы**
2. Изучите SQL-DML
3. Выполните все запросы из списка стандартных запросов. Продемонстрируйте результаты преподавателю.
4. Получите у преподавателя и реализуйте SQL-запросы в соответствии с индивидуальным заданием. Продемонстрируйте результаты преподавателю.
5. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE - в виде ХП. Выложите скрипт в SVN.
6. **Ход работы**

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

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

1. Выборка всех данных из каждой таблицы

create view AccountServicesSel as select \* from accountservices;

create view AddServicesSel as select \* from addservices;

create view ClassesSel as select \* from classes;

create view ClassToTicketSel as select \* from classtoticket;

create view ClientsSel as select \* from clients;

create view ClientToclassSel as select \* from clienttoclass;

create view DiscountsSel as select \* from discounts;

create view PlaceSel as select \* from place;

create view ServicesToTypesSel as select \* from servicestotypes;

create view TicketsSel as select \* from tickets;

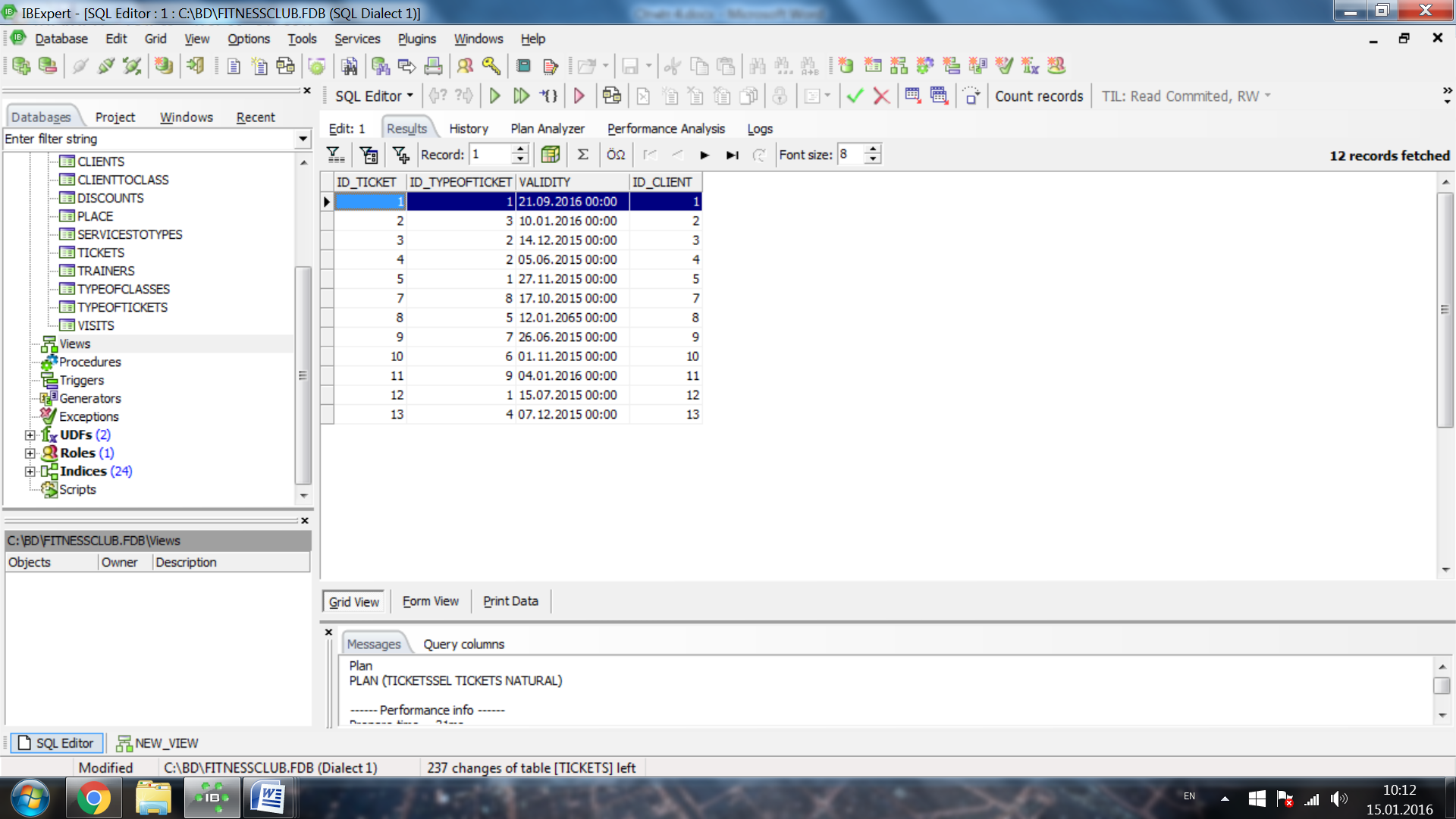
create view TrainersSel as select \* from trainers;

create view TypeOfClassesSel as select \* from typeofclasses;

create view TypeOfTicketsSel as select \* from typeoftickets;

create view VisitsSel as select \* from visits;

Выборка данных из таблицы TICKETS:



1. Выборка данных из одной таблицы при нескольких условиях, с использованием логических операций LIKE, BETWEEN, IN

Были созданы следующие выборки:

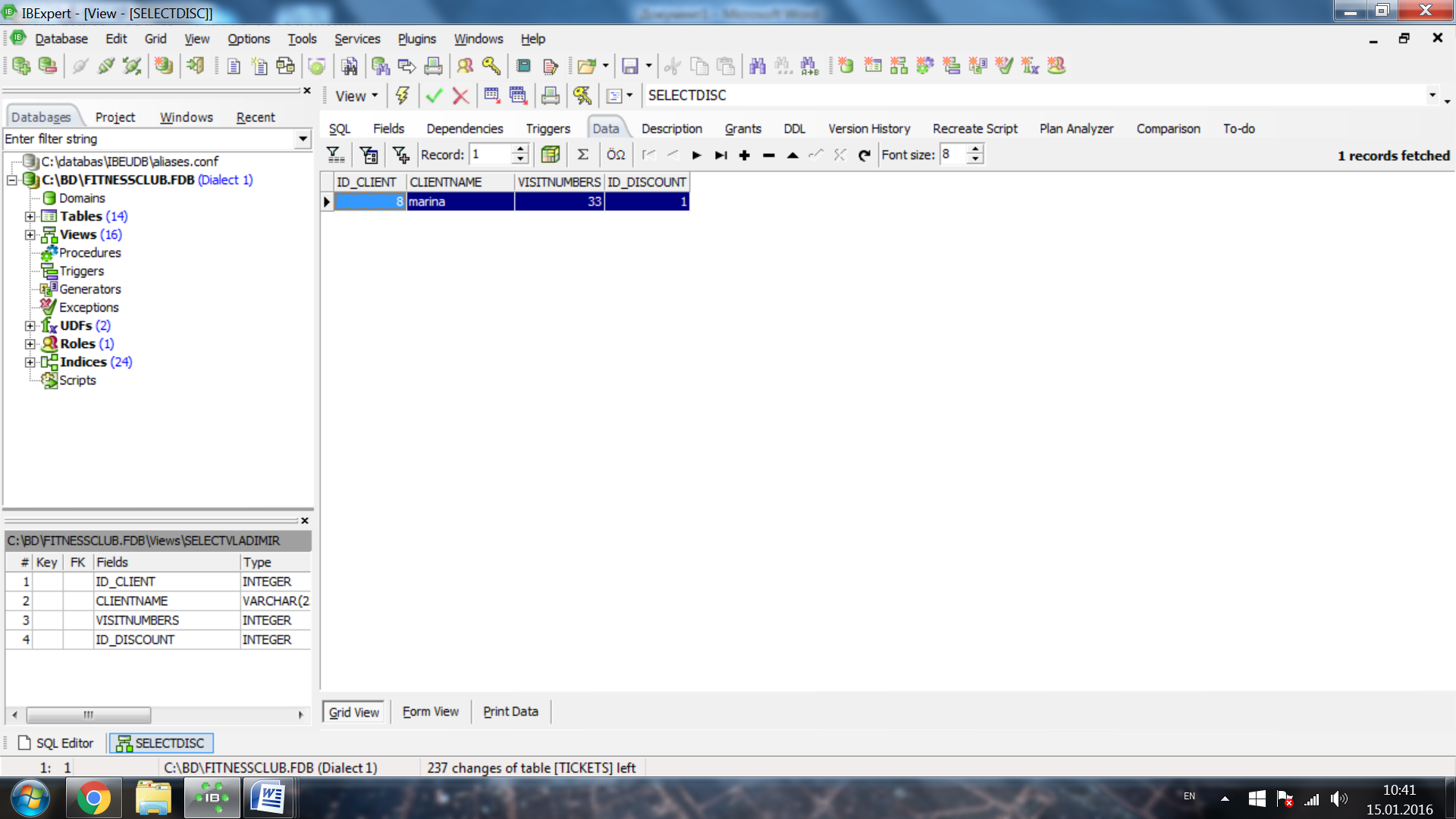
1. Из таблицы CLIENTS показать клиента, у которого 33 посещения клуба.
2. Из таблицы CLASSES показать занятия, которые проводятся в тренажерном зале и малом зале (ID=1, ID=2)
3. Из таблицы TRAINERS показать тех тренеров, чей стаж работы находится в интервале от 1 до 3х лет

create view selectdisc from CLIENTS where VISITNUMBERS like 33;

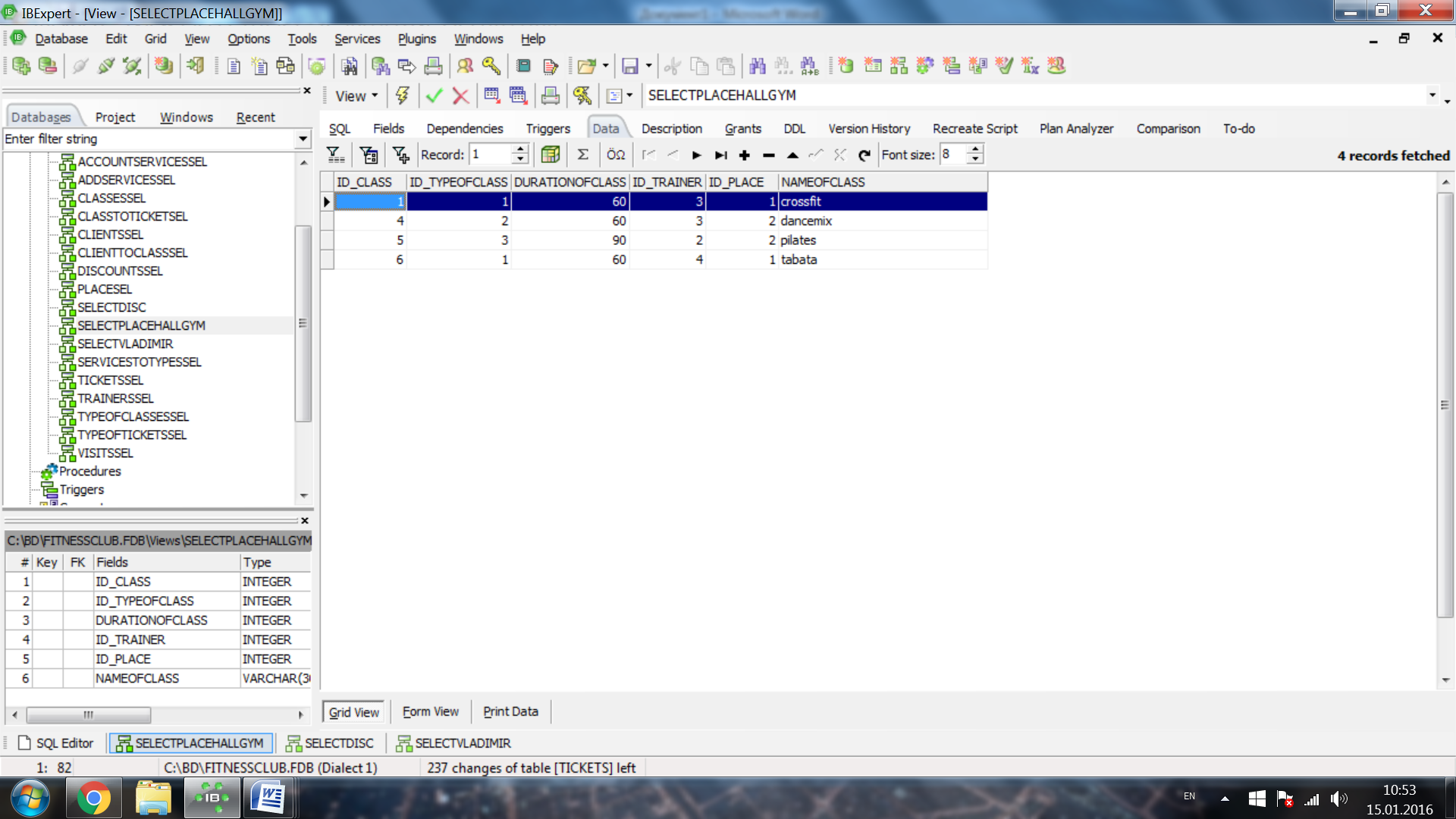
create view selectplacehallgym as select \* from CLASSES where ID\_PLACE in (1,2);

create view selectTrainers as select \* from TRAINERS where WORKINGTIMEOFTRAINER between 1 and 3;

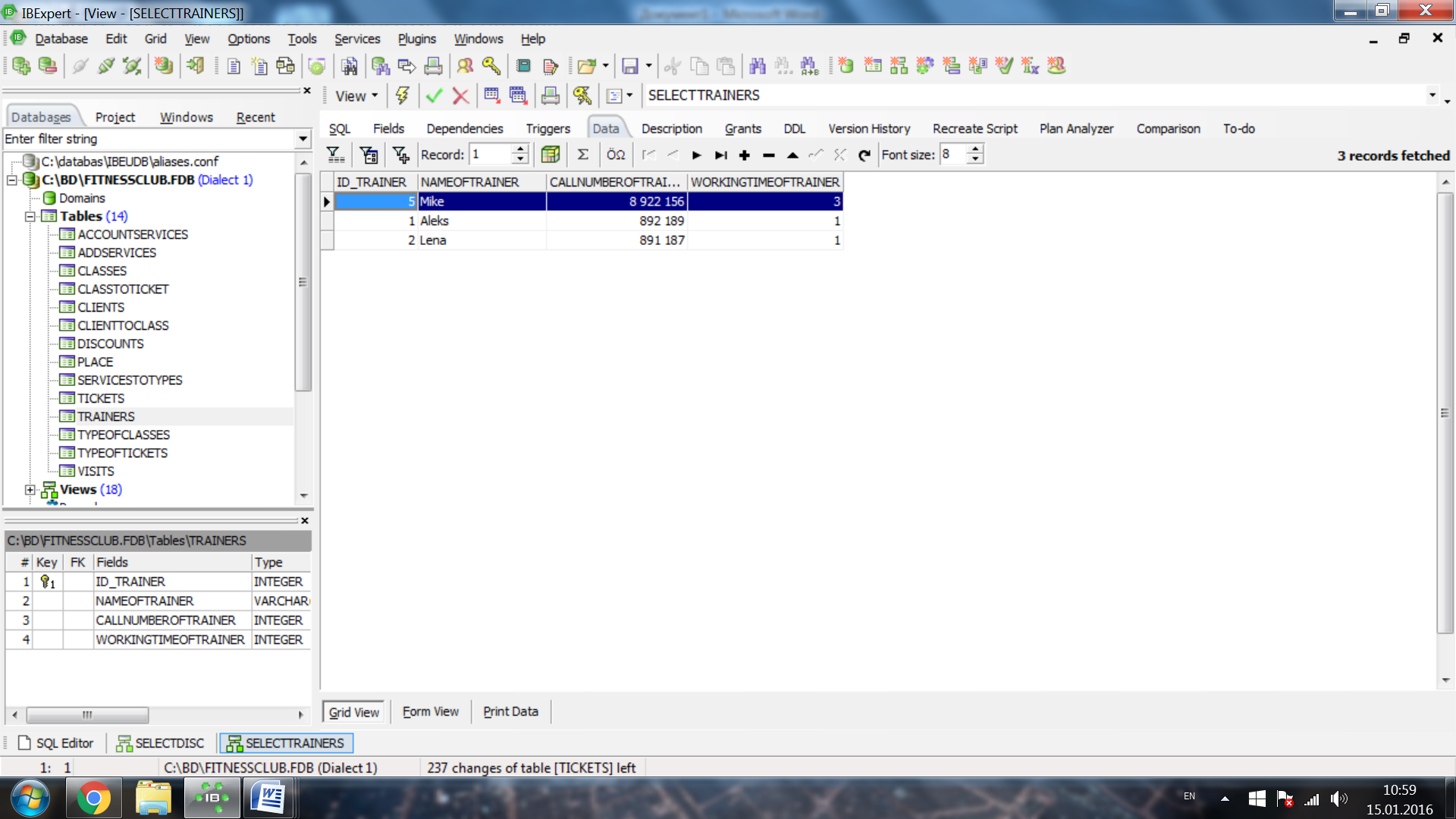
Пример выборки I:



Пример выборки II:



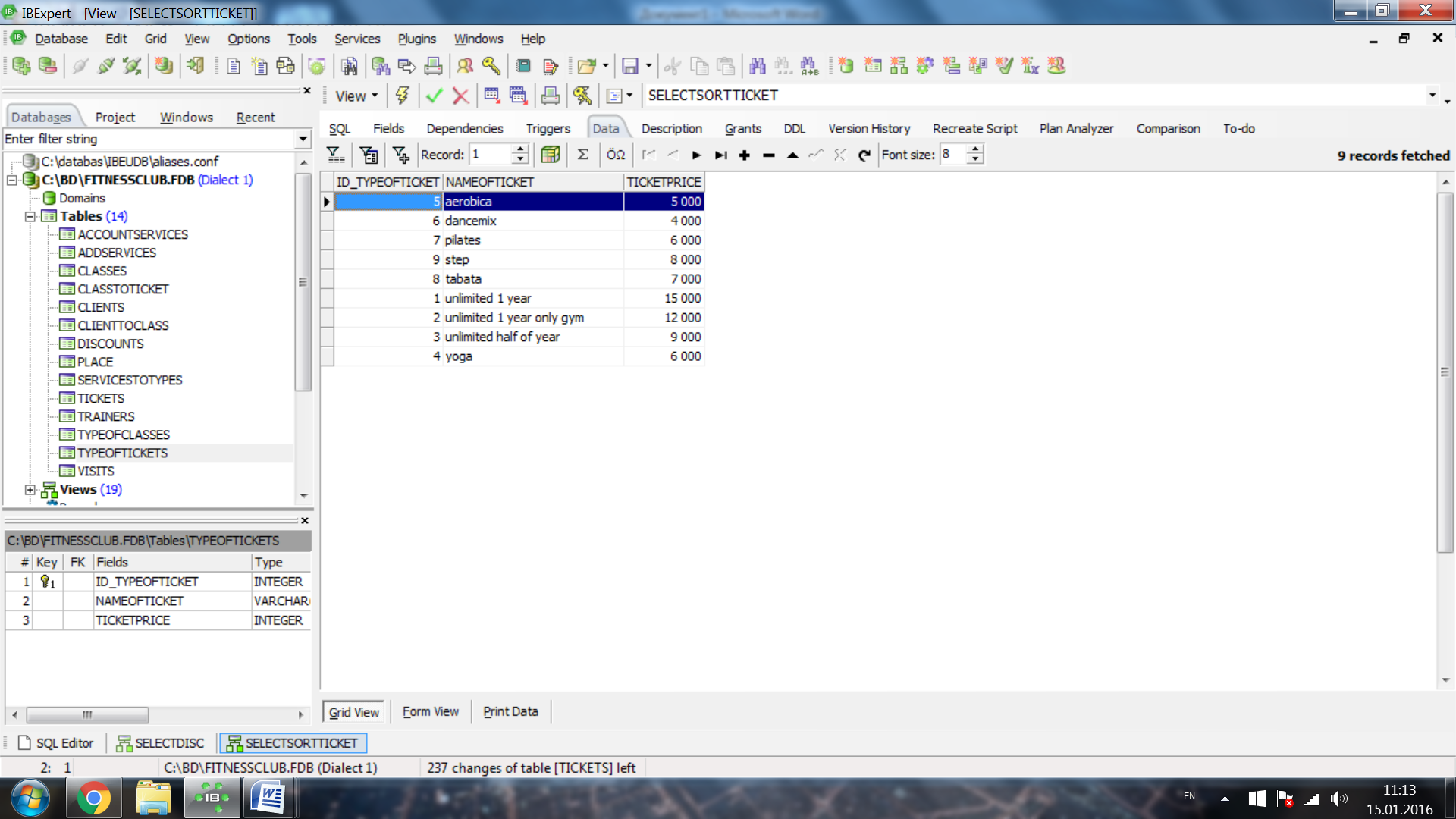
Пример выборки III:



1. Выборка всех данных с сортировкой по нескольким полям

Типы абонементов, отсортированные по имени и цене:

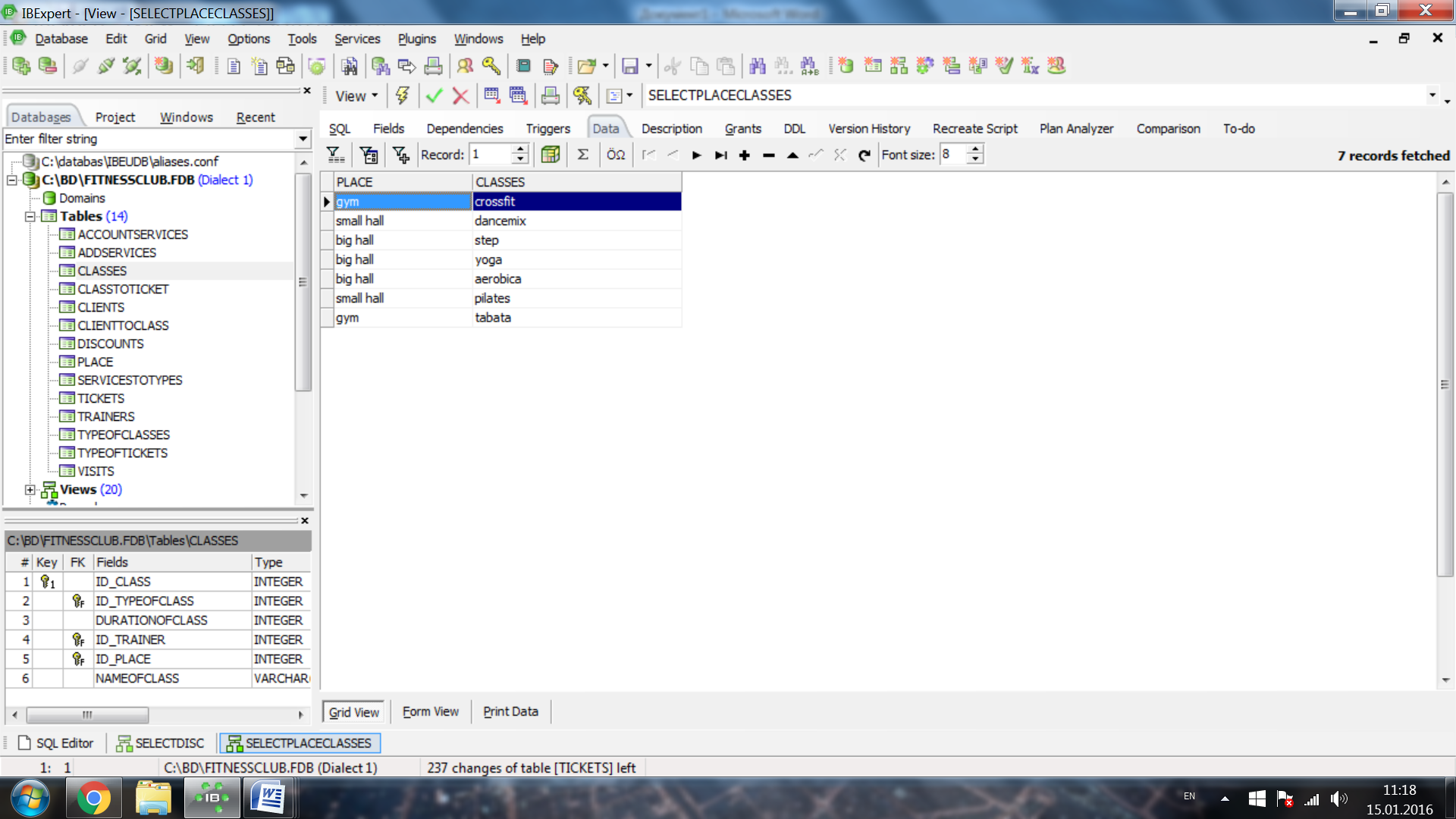
create view selectSortTicket as select \* from typeoftickets order by NAMEOFTICKET asc, TICKETPRICE asc;



1. Выборка данных из связанных таблиц

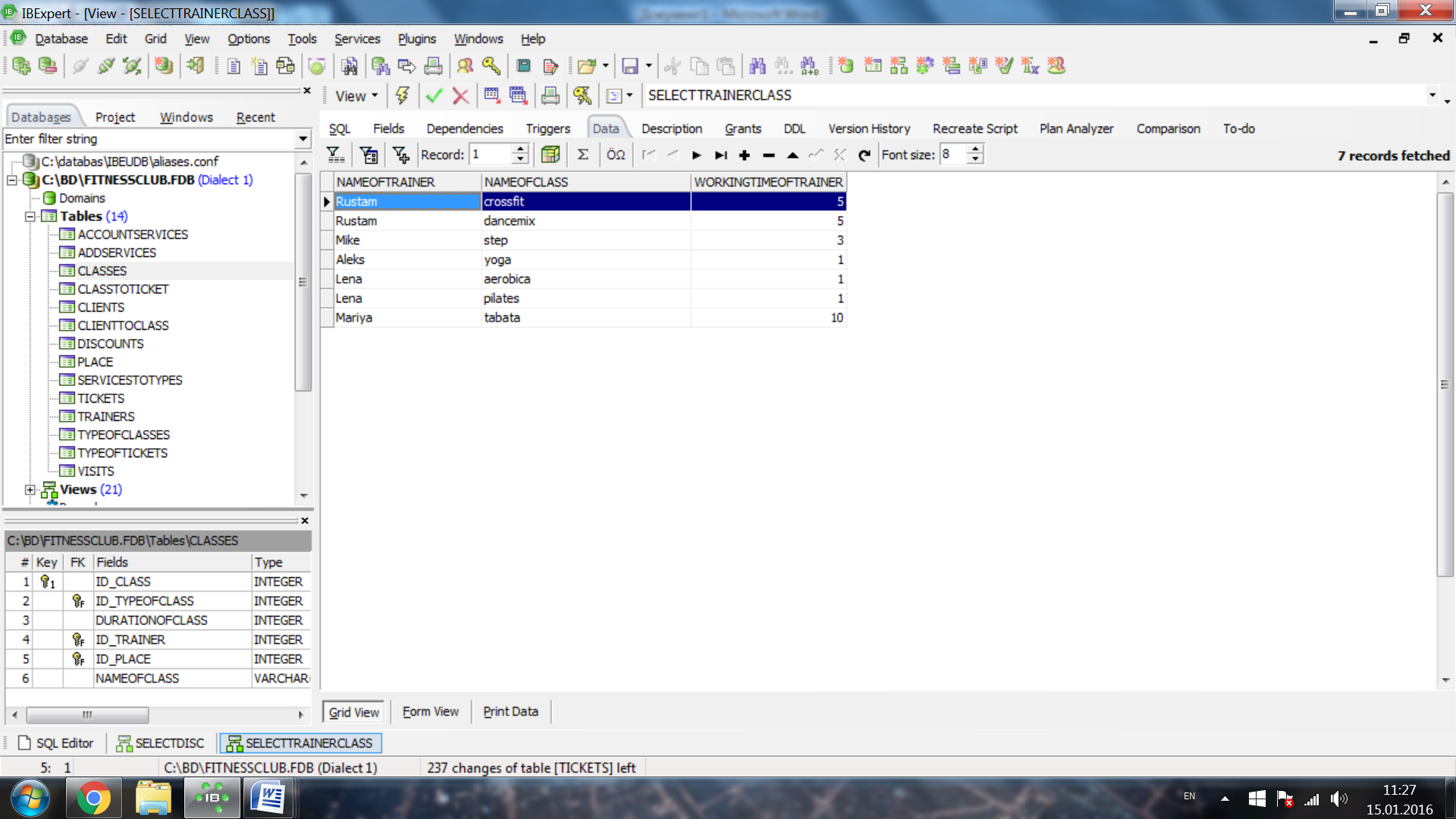
Выборка, отображающая место проведения и занятие:

create view selectPlaceClasses as select place.nameofplace as place, classes.nameofclass as classes from place, classes where classes.id\_place = place.id\_place;



Выборка, отображающая занятие, тренера, который его проводит и его стаж:

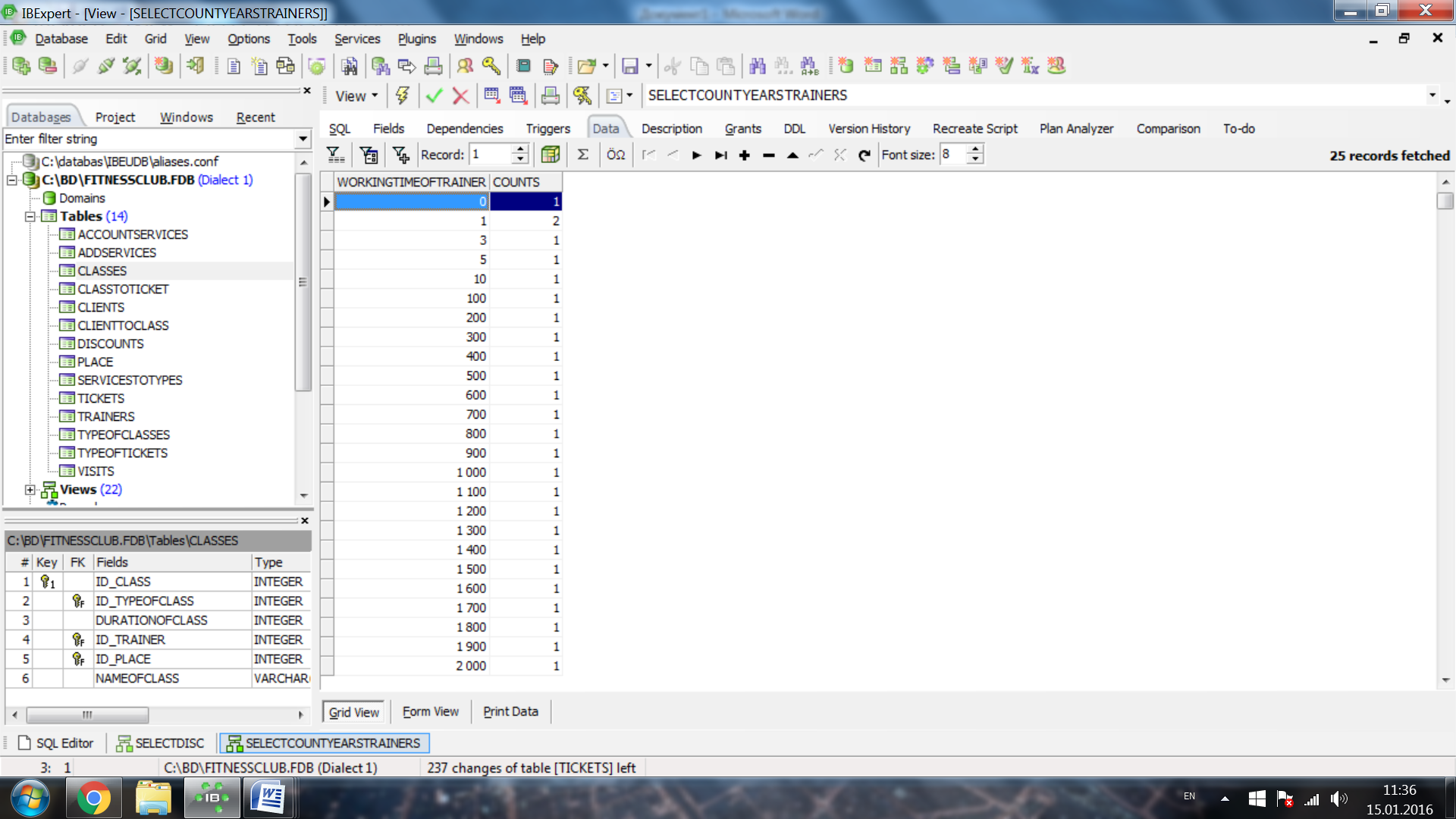
create view selectTrainerClass as select TRAINERS.nameoftrainer as nameoftrainer, classes.nameofclass as nameofclass, TRAINERS.workingtimeoftrainer as workingtimeoftrainer from TRAINERS, classes where classes.id\_trainer = trainers.id\_trainer;



1. Запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки

Определение кол-ва тренеров с различным стажем работы:

create view selectCountYearsTrainers as select trainers.workingtimeoftrainer as workingtimeoftrainer, COUNT(trainers.workingtimeoftrainer) as Counts from trainers group by trainers.workingtimeoftrainer;

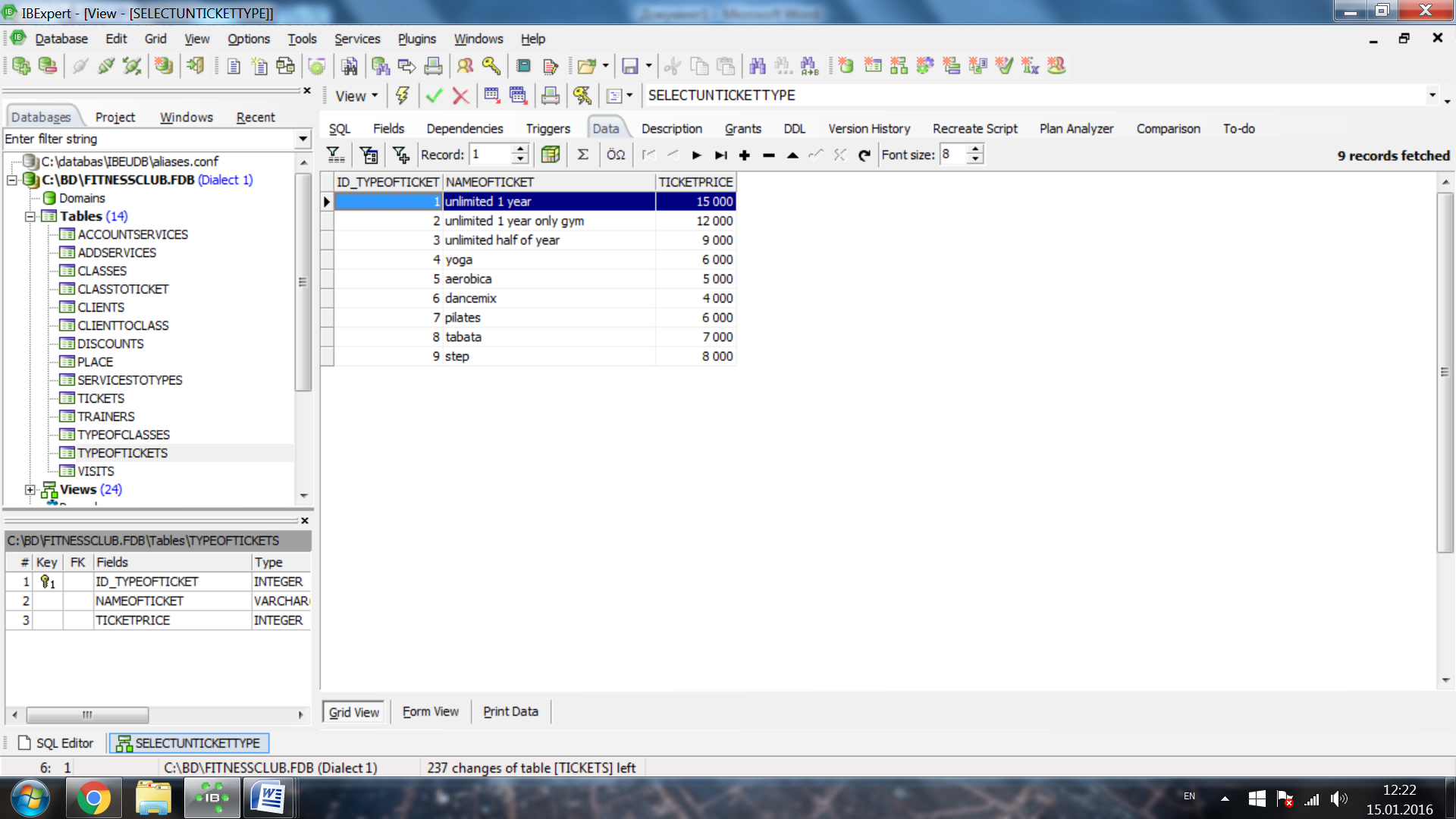


1. Использование вложенного запроса

Отображение только тех типов абонементов, которые куплены:

create view selectUnTicketType as select \* from typeoftickets where id\_typeofticket

in (select id\_typeofticket from tickets);



1. Использование оператора INSERT для добавления по одной записи в каждую таблицу:

connect 'C:\BD\FITNESSCLUB.FDB' user 'SYSDBA' password 'masterkey';

create procedure insert\_one ( i int, d int)

as begin

insert into accountservices (id\_client, id\_service)

values (:i, :d);

end;

create procedure insert\_two ( i int, n varchar(255), d int)

as begin

insert into addservices (id\_service, nameservice, priceservice)

values (:i, :n, :d);

end;

create procedure insert\_three ( i int, t int, k int, s int, p int, n varchar(30))

as begin

insert into classes (ID\_CLASS, ID\_TYPEOFCLASS, durationOfClass, ID\_TRAINER, ID\_PLACE, nameofclass)

values (:i, :t, :k, :s, :p, :n);

end;

create procedure insert\_four ( i int, t int)

as begin

insert into classtoticket (ID\_TYPEOFTICKET, ID\_CLASS)

values (:i, :t);

end;

create procedure insert\_five ( i int, n varchar(255), v int, s int)

as begin

insert into Clients (ID\_CLIENT, clientName, visitNumbers, ID\_DISCOUNT)

values (:i, :n, :v, :s);

end;

create procedure insert\_six ( i int, k int)

as begin

insert into clienttoclass (ID\_CLASS, ID\_CLIENT)

values (:i, :k);

end;

create procedure insert\_seven ( i int, n varchar (255), p date)

as begin

insert into discounts (ID\_DISCOUNT, nameOfDiscount, valueOfDiscount)

values (:i, :n, :p);

end;

create procedure insert\_eight ( i int, n varchar(255))

as begin

insert into Place (ID\_PLACE, nameofPlace)

values (:i, :n);

end;

create procedure insert\_nine ( i int, p int)

as begin

insert into servicestotypes (id\_service, id\_typeofticket)

values (:i, :p);

end;

create procedure insert\_ten( i int, s int, l date, k int)

as begin

insert into tickets (ID\_TICKET, ID\_TYPEOFTICKET, validity, ID\_CLIENT)

values (:i, :s, :l, :k);

end;

create procedure insert\_eleven ( i int, n varchar(255), e int, s int)

as begin

insert into Trainers (ID\_TRAINER, nameOfTrainer, callNumberOfTrainer, workingTimeOfTrainer)

values (:i, :n, :e, :s);

end;

create procedure insert\_twelve ( i int, n varchar(255))

as begin

insert into typeofclasses (ID\_TYPEOFCLASS, typeclass)

values (:i, :n);

end;

create procedure insert\_thirteen ( i int, n varchar(255), v int)

as begin

insert into typeoftickets (ID\_TYPEOFTICKET, nameOfTicket, ticketprice)

values (:i, :n, :v);

end;

create procedure insert\_fourteen ( i int, v date)

as begin

insert into visits (id\_client, dateofvisit)

values (:i, :v);

end;

Процедуры созданы. Добавили записи. И проверили, что записи добавлены в таблицу.

1. С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию

Изменить или добавить скидку для клиентов, у которых больше 40 посещений.

connect 'C:\BD\FITNESSCLUB.FDB' user 'SYSDBA' password 'masterkey';

create procedure update\_discount (discoun int)

as

begin

update clients set ID\_DISCOUNT = :discoun where clients.visitnumbers > 40;

end;

1. С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики

Процедура удаляет занятие, длящееся меньше всех по времени:

connect 'C:\BD\FITNESSCLUB.FDB' user 'SYSDBA' password 'masterkey';

create procedure del\_class (i int)

as

begin

delete from classes where ID\_CLASS = :i and

durationofclass = (select MIN(durationofclass) from classes where ID\_CLASS = :i);

end;

1. С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

Удалили типы абонементов, которые никем не приобретены:

connect 'C:\BD\FITNESSCLUB.FDB' user 'SYSDBA' password 'masterkey';

create procedure delete\_types as

begin

delete from typeoftickets where ID\_TYPEOFTICKET not in (select ID\_TYPEOFTICKET from tickets);

end;

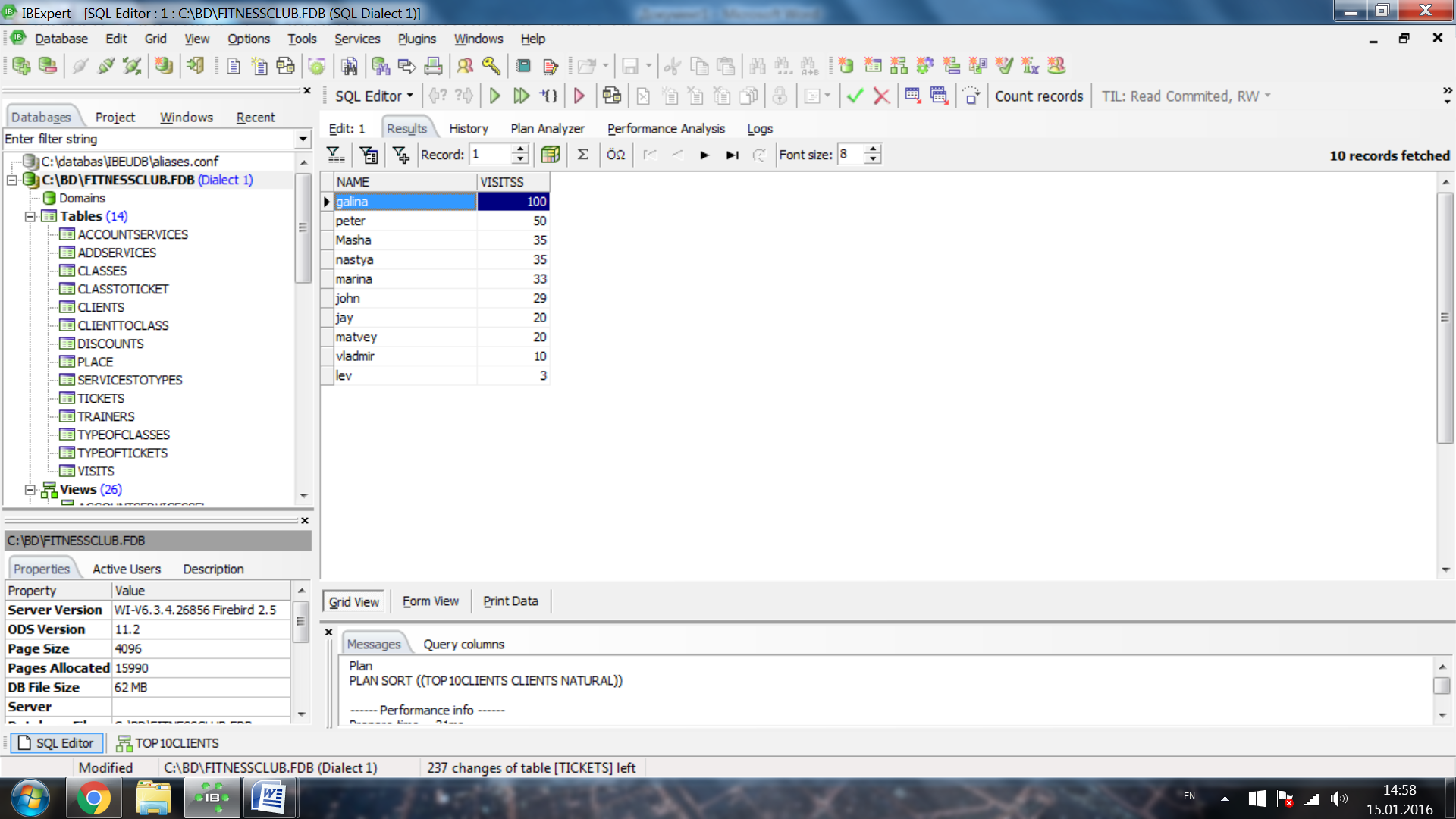
**Индивидуальные задания**

1. За заданный период времени вывести 10 клиентов, которые чаще всего посещали клуб.
2. За заданный период вывести 5 наиболее популярных типов по покупкам абонементов.
3. Удалить неиспользуемые виды занятий

1. create view top10clients as

select first 10 clients.clientname as name, clients.visitnumbers as visitss from clients

order by clients.visitnumbers desc;

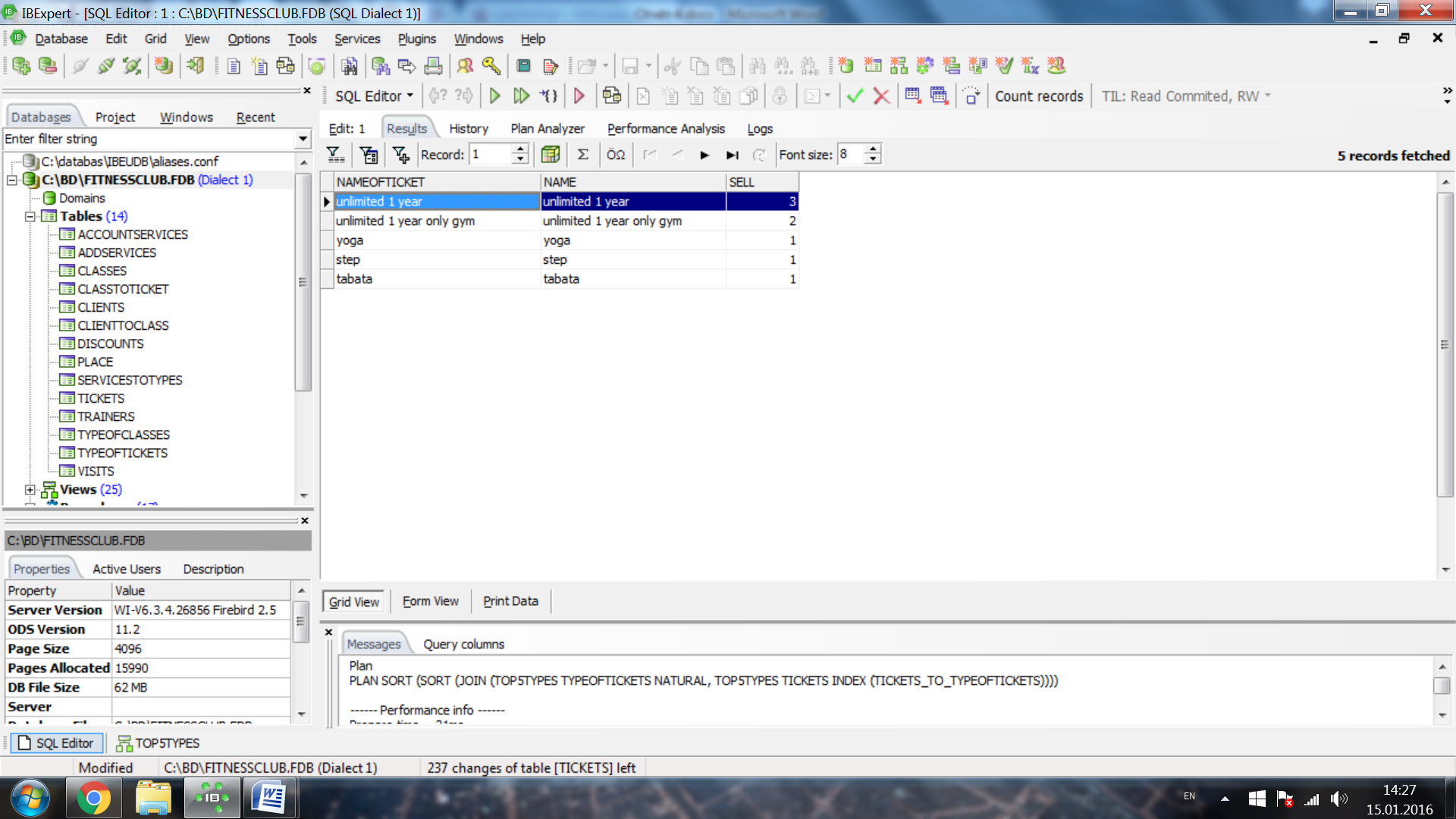


2. create view top5types as select first 5 typeoftickets.nameofticket, typeoftickets.nameofticket as Name,

COUNT(tickets.id\_typeofticket) as Sell from typeoftickets, tickets

where tickets.id\_typeofticket = typeoftickets.id\_typeofticket group by

typeoftickets.nameofticket order by Sell desc;



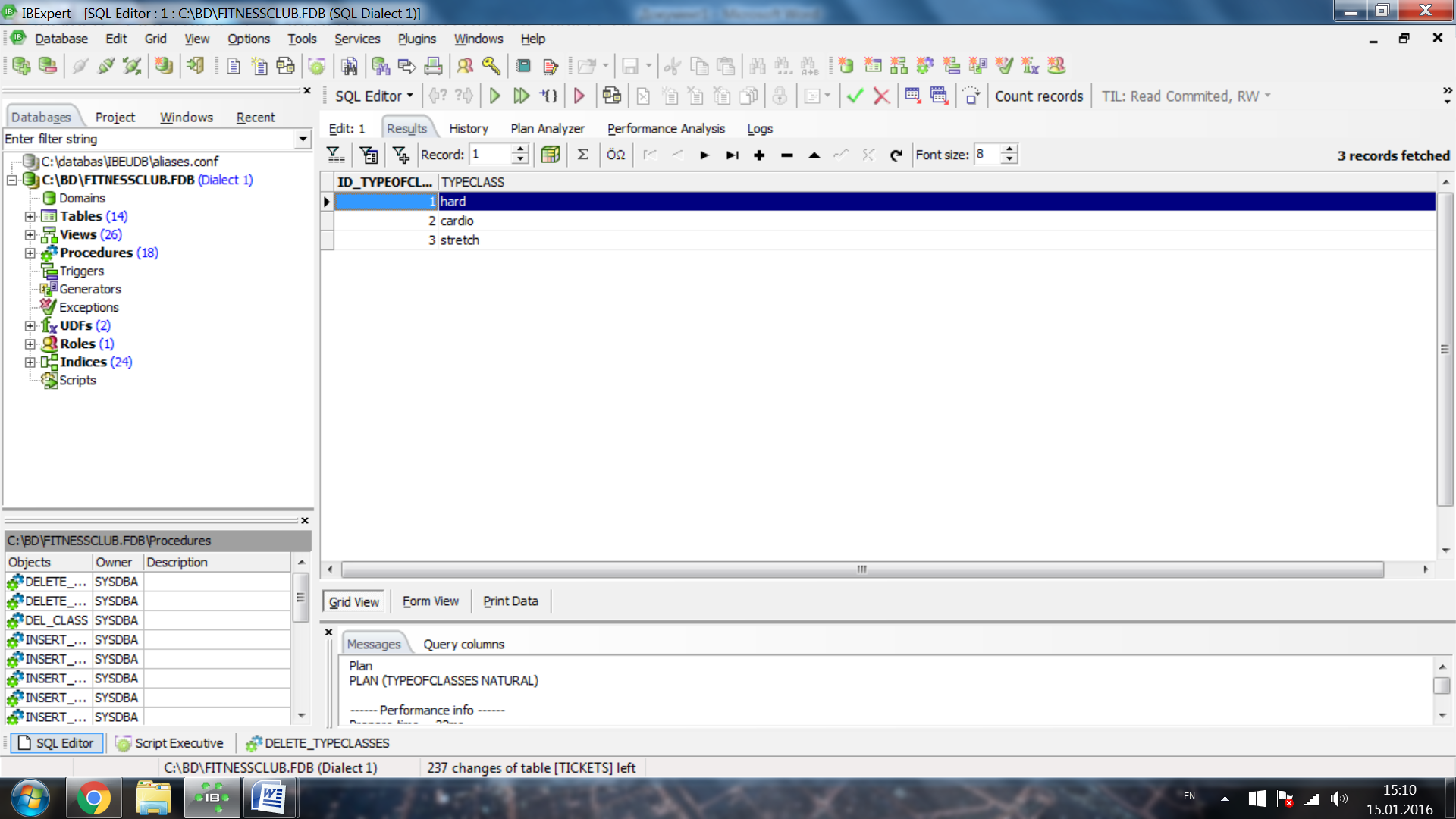
3. connect 'C:\BD\FITNESSCLUB.FDB' user 'SYSDBA' password 'masterkey';

create procedure delete\_typeclasses as

begin

delete from typeofclasses where ID\_TYPEOFCLASS not in (select ID\_TYPEOFCLASS from classes);

end;



1. **Выводы**

Таким образом, изучены основы языка управления данными SQL-DML. Выполнены стандартные запросы извлечения данных, запросы в соответствии с индивидуальным заданием. Также познакомились с представлениями и хранимыми процедурами, с помощью которых можно добавлять данные в БД.

Использовались такие команды языка DML: insert (добавить), update (обновить), delete (удалить), select (выборка данных). Язык удобен для написания запросов разной сложности. При обращении к нескольким таблицам в запросе нужно выбирать подходящую связь между таблицами для более быстрого выполнения SQL-запроса.