Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра Компьютерных систем и программных технологий

Отчет по лабораторной работе 5

**«Хранимые процедуры»**

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 возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур

1. **Программа работы**
2. Изучить возможности языка PSQL
3. Создать две хранимые процедуры в соответствии с индивидуальным заданием, полученным у преподавателя
4. Выложить скрипт с созданными сущностями в svn
5. Продемонстрировать результаты преподавателю
6. **Ход работы**

Реализуем хранимые процедуры:

1. Рассчитать выручку по типам абонементов. Нужно учитывать как продажу абонементов, так и доп. услуг.
2. Рассчитать соотношение количества продаж новым клиентам к количеству продаж существующим за заданный период.

1. Выручка будет складываться из: купленный клиентом абонемент + доп.услуги, входящие в абонемент + доп.услуги, которые клиент приобрел отдельно.

create or alter procedure HP1

returns (

TYPETICK varchar(255),

VIRYCHKA integer)

as

declare variable I integer;

declare variable CENAABON integer = 0;

declare variable CHELOVEK integer;

declare variable IDSKIDKI integer;

declare variable SKIDKA integer;

declare variable BUFVIR integer = 0;

declare variable CENA1 integer;

declare variable SERVID integer;

declare variable SERVCENA integer;

declare variable SERVCOUNT integer;

declare variable CENA2 integer;

begin

for select typeoftickets.id\_typeofticket, typeoftickets.nameofticket, typeoftickets.ticketprice

from typeoftickets into :i , :typetick , :cenaabon

do

begin

for select tickets.id\_client from tickets

where tickets.id\_typeofticket = :i into :chelovek

do

begin

select clients.id\_discount

from clients

where clients.id\_client = :chelovek

into :idskidki;

select discounts.valueofdiscount

from discounts

where discounts.id\_discount =:idskidki

into :skidka;

skidka = 100 - skidka;

cenaabon = (cenaabon \* skidka)/100;

bufvir = bufvir + cenaabon;

cena1 = bufvir;

end

for select servicestotypes.id\_service

from servicestotypes where servicestotypes.id\_typeofticket = :i into :servid

do

begin

select addservices.priceservice

from addservices

where addservices.id\_service = :servid

into :servcena;

select count(accountservices.id\_service)

from accountservices

where accountservices.id\_service = :servid

into :servcount;

cena2 = servcena \* servcount;

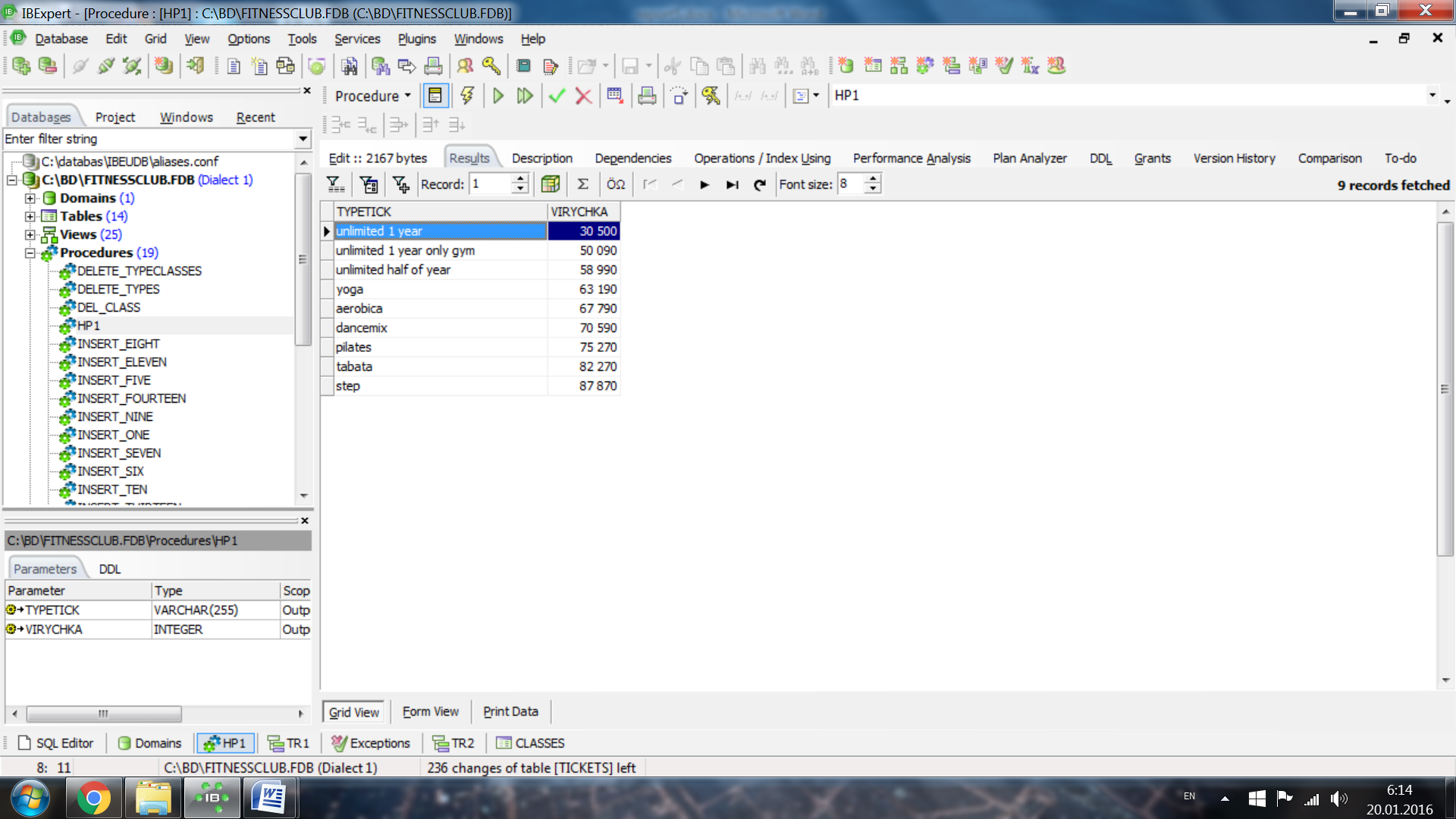
end

virychka = cena1 + cena2;

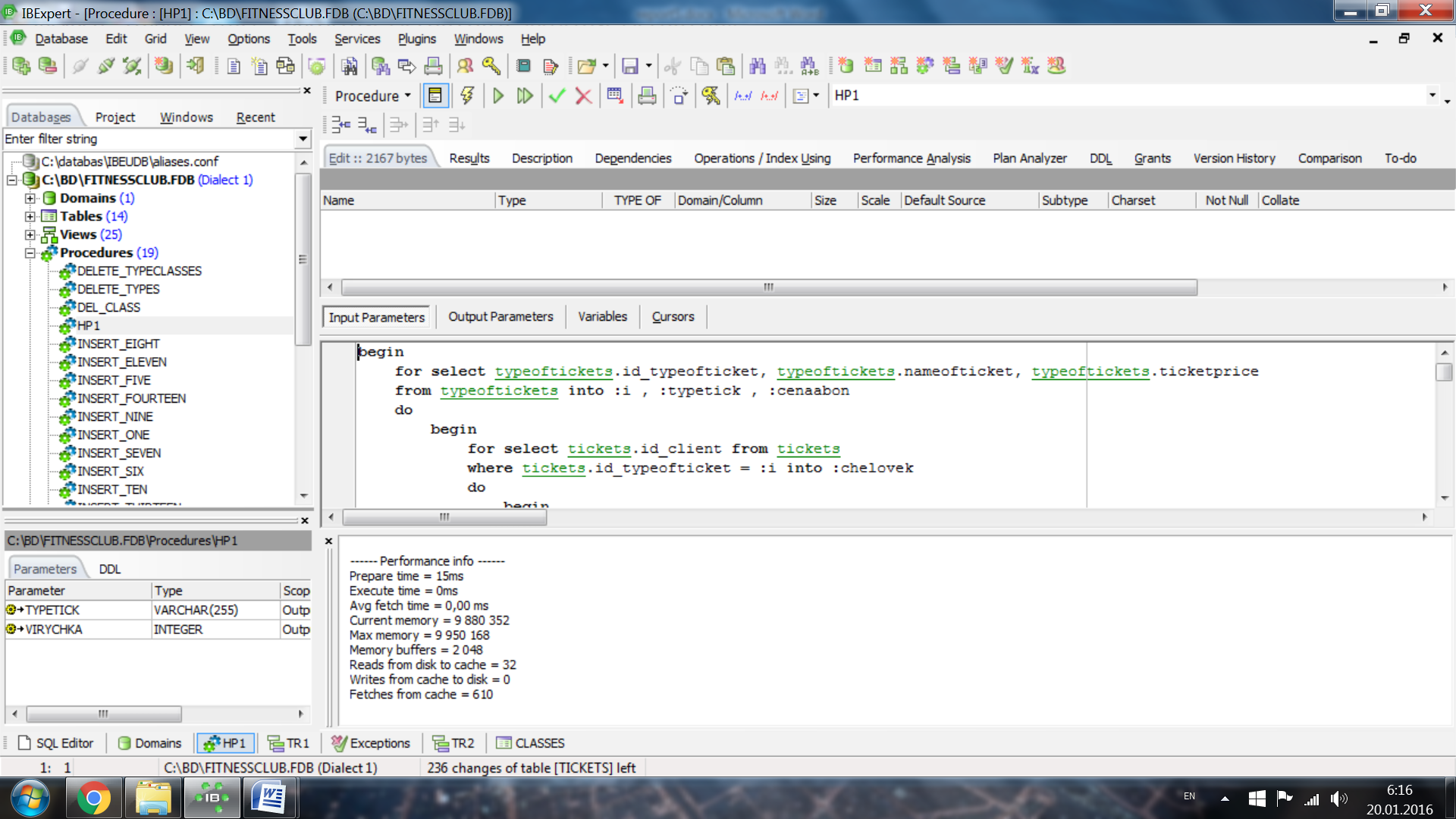
suspend;

end

Результат работы:



Время выполнения:



2.

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

create procedure sales\_ratio(min\_date date, max\_date date)

returns (new\_sales int, ratio\_of float)

as

declare variable all\_sales int default 0;

declare variable cur\_client int default 0;

declare variable cur\_count int;

declare variable cur\_old int default 0;

declare variable all\_sales\_period int;

begin

for select clients.id\_client from clients,tickets where **--те,у кого уже есть абон**

tickets.validity < :min\_date and  **-- и те,кто покупает повторно в зад пер**

tickets.id\_client = clients.id\_client group by clients.id\_client

into :cur\_client

do begin

all\_sales = all\_sales + 1;

select COUNT(tickets.id\_ticket) from tickets where

tickets.validity between :min\_date and :max\_date and

tickets.id\_client = :cur\_client

into cur\_count;

if (cur\_count!=0) then begin

cur\_old = cur\_old + 1;

end

end

select COUNT(clients.id\_client) from clients,tickets where **--новые продажи (в зад период)**

tickets.validity between :min\_date and :max\_date and

tickets.id\_client = clients.id\_client

into :all\_sales\_period;

new\_sales = all\_sales\_period - cur\_old;

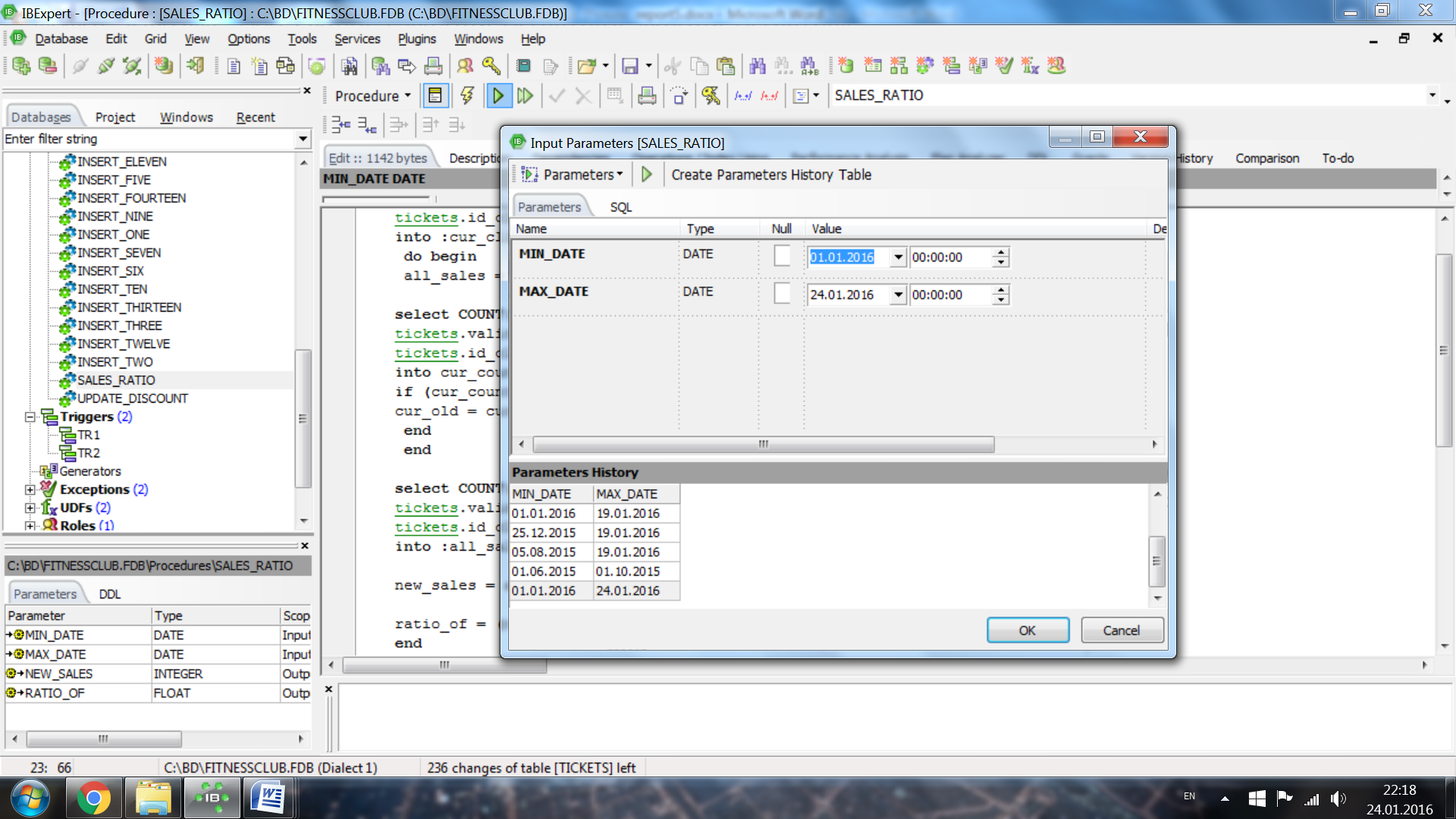
ratio\_of = (new\_sales/all\_sales)\*100;

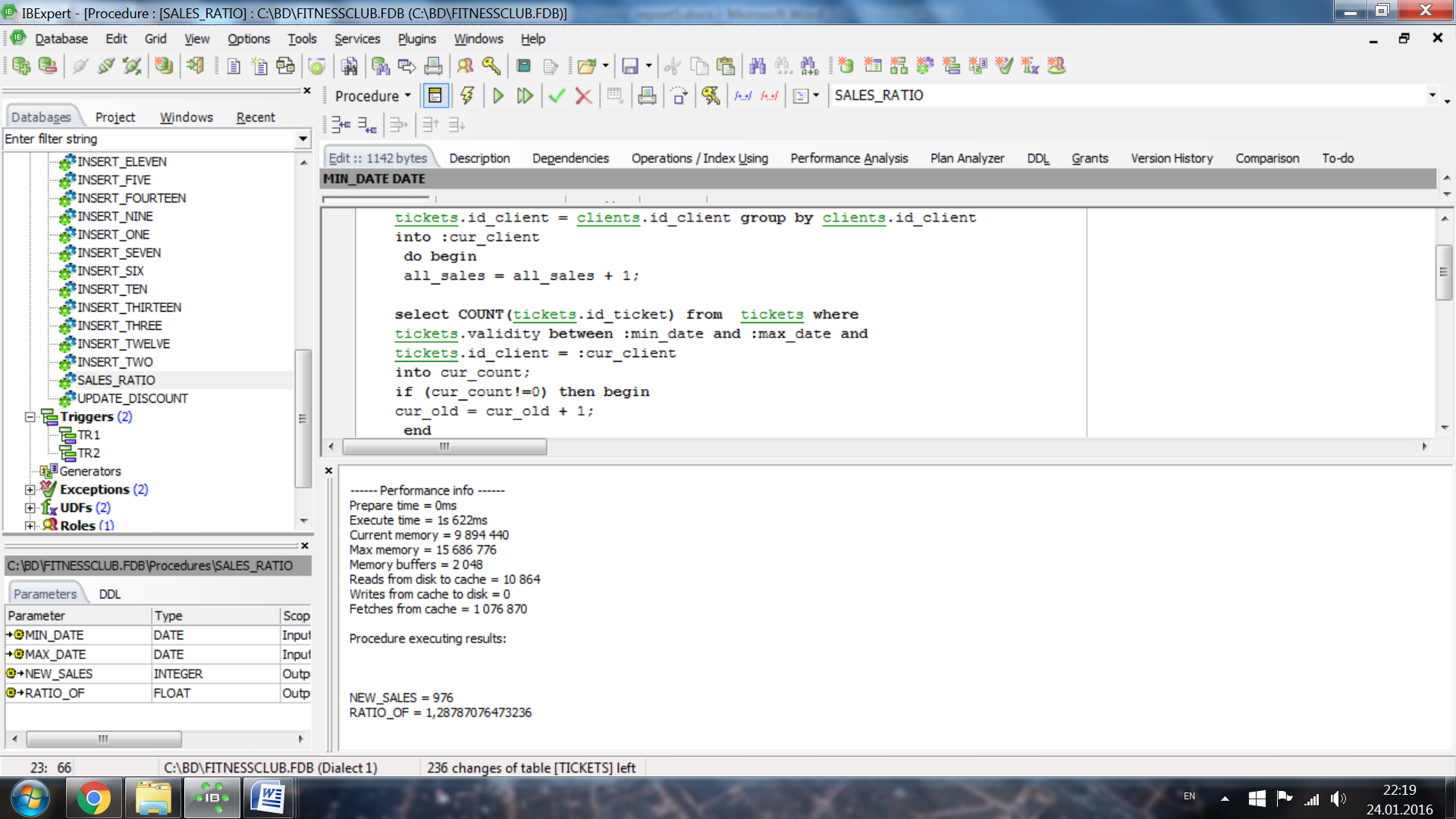
end

Результат работы:

Итак, существующие продажи клиентам это такие продажи, которые были совершены в заданный период и до него. Таким образом, нам нужно вычислить количество продаж абонементов клиентам, которые совершили повторную покупку в указанный период. И отношение будет между новыми клиентами и существующими, то есть, суммой тех, которые сделаны до заданного периода и которые приобретены снова в заданный период.

Сначала нужно задать нужный нам период времени, за которое хотим посмотреть кол-во «новых» продаж.





NEW\_SALES – количество продаж новым клиентам, RATIO\_OF – соотношение новых к существующим.

**4. Вывод**

Таким образом, в данной лабораторной работе созданы две хранимые процедуры. Преимущество хранимых процедур в том, что они выполняются на стороне сервера, т.е. не тратится время на передачу данных и т.д. Недостаток состоит в том, что хранимые процедуры зависят от типа и версии используемой СУБД , и, следовательно, перенос проекта из одной СУБД в другую достаточно сложен.