Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра Компьютерных систем и программных технологий

Отчет по лабораторной работе 6

**«Триггеры»**

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 и триггеров.

1. **Программа работы**
2. Создать два триггера: один триггер для автоматического заполнения ключевого поля, второй триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице
3. Создать триггер в соответствии с индивидуальным заданием, полученным у преподавателя
4. Создать триггер в соответствии с индивидуальным заданием, вызывающий хранимую процедуру
5. Выложить скрипт с созданными сущностями в svn
6. Продемонстрировать результаты преподавателю
7. **Ход работы**

Создадим два триггера: первый – автоматически заполняет ключевое поле id\_trainer в таблице TRAINERS, второй – контролирует целостность данных в подчиненной таблице CLASSES при удалении или изменении записей в таблице TRAINERS.

Первый:

SET TERM ^ ;

CREATE OR ALTER TRIGGER FILLING\_TRAINERS FOR TRAINERS

ACTIVE BEFORE INSERT POSITION 0

as

begin

if ((new.id\_trainer is null)) then

new.id\_trainer = gen\_id(my\_gen,1);

end

^

SET TERM ; ^

Второй:

SET TERM ^ ;

CREATE OR ALTER TRIGGER CONTROL\_TRAINERS FOR TRAINERS

ACTIVE BEFORE UPDATE OR DELETE POSITION 0

as

begin

if (OLD.id\_trainer in (select classes.id\_trainer from classes)) then

exception error\_;

end

^

SET TERM ; ^

Реализуем триггеры в соответствии с индивидуальными заданиями:

1. При покупке клиентом абонемента проверять наличие у него действующего абонемента. Если есть - не добавлять.
2. При регистрации посещения проверять наличие действующего абонемента у клиента. Если нет - не регистрировать посещение.

1.

CREATE OR ALTER TRIGGER TR1 FOR TICKETS

ACTIVE BEFORE INSERT POSITION 0

AS

declare variable bufdatstart date;

declare variable bufdatend date;

declare variable i date;

begin

i = new.validity;

for select tickets.validity,tickets.finish\_date

from tickets

where tickets.id\_client = new.id\_client into :bufdatstart,:bufdatend

do

begin

if(bufdatstart<= i and i<= bufdatend)

then

begin

exception deisttick;

end

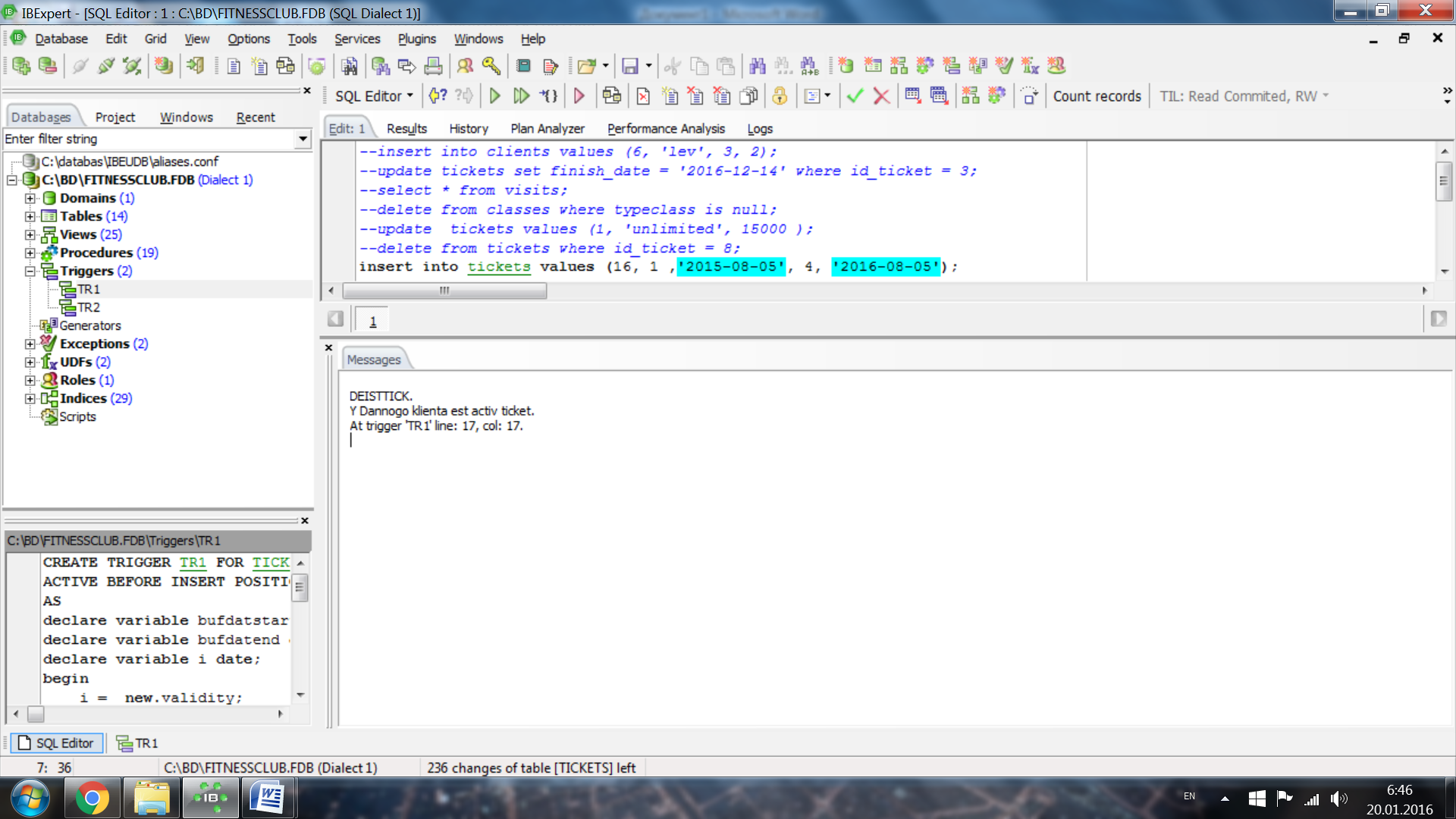
end

end

Триггер проверяет дату покупки клиентом абонемента – если она находится между датами приобретения, сделанного ранее, и даты конца срока действия имеющегося абонемента – то выдается исключение.

Далее пример проверки правильной работы триггера:

Пробуем купить от лица клиента с ID=4 абонемент, но он у него уже существует.



2.

CREATE OR ALTER TRIGGER TR2 FOR VISITS

ACTIVE BEFORE INSERT POSITION 0

AS

declare variable bufdatstart date;

declare variable bufdatend date;

begin

for select tickets.validity,tickets.finish\_date

from tickets

where tickets.id\_client = new.id\_client into :bufdatstart,:bufdatend

do

begin

if((new.dateofvisit<bufdatstart) or (new.dateofvisit > bufdatend))

then

begin

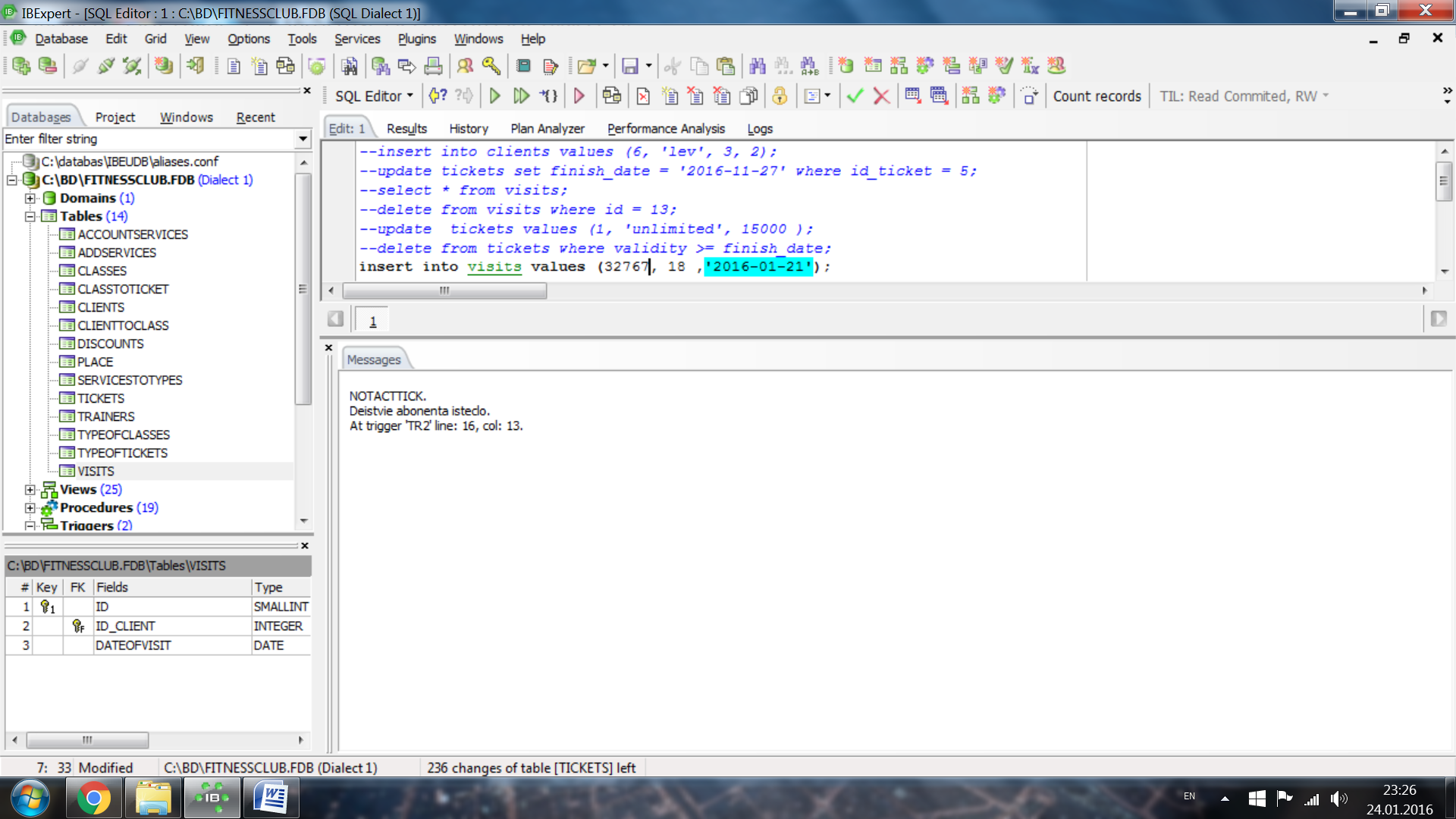
exception notacttick;

end

end

end

Проверяем введенную дату посещения и клиента, собирающегося пойти на занятие. Если эта дата меньше, чем дата покупки абонемента, то, следовательно, абонемента у него в этот момент нет. А значит, не фиксируем посещение. И также, если абонемент у клиента истек, а он пришел регистрироваться на занятие, то также срабатывает триггер и запись не производится. Пример ниже как раз демонстрирует запись клиента, у которого уже истек абонемент!



1. **Вывод**

Таким образом, в данной лабораторной работе созданы триггеры. Триггеры полезно и уместно использовать для того, чтобы проверить корректность вносимых в БД данных. С помощью триггеров можно выдавать сообщения (exceptions) о том, что необходимо выполнить некоторые действия при изменении таблиц, о том, что неверно введены данные. Также триггеры используются для оповещения об изменении данных в таблицах. С помощью триггеров можно накладывать ограничения на вносимые данные согласно требованиям.