Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт о лабораторной работе №6**

**Дисциплина**: Базы данных

**Тема**: Триггеры

Выполнил студент гр. 43501/1 М.А. Иконников

(подпись)

Руководитель А.В. Мяснов

(подпись)

“ ” 2015 г.

Санкт-Петербург

2015

1. **Цель**

Познакомить студентов 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. **Выполнение работы**

Был создан триггер для автоматического заполнения поля id\_trainer в таблице Trainers

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

--create generator increment;

--set generator increment to TRAINERS.id\_trainer;

drop trigger auto\_fill;

create trigger auto\_fill for TRAINERS before insert

as

begin

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

end;

Затем был создан триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице TicketTypes

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

create exception errror 'Error deleting ticket type';

drop trigger check\_cel;

create trigger check\_cel for tickettypes before delete or update

as

begin

if (OLD.id\_type in (select seasontickets.id\_type from seasontickets)) then

exception errror;

end;

Индивидуальное задание:

Реализовать триггеры:

1. При приобретении абонемента рассчитывать его стоимость с учетомскидки клиента.
2. При превышении суммарных приходов от клиента за некоторый период некоторой величины - единоразово предоставлять наиболее часто заказываемую услугу.

Первый триггер:

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

drop trigger real\_price;

create trigger real\_price for seasontickets before insert

as

declare variable discid int;

declare variable discval int;

declare variable fprice int;

declare variable rprice float;

begin

select clients.id\_discount from clients

where new.id\_client = clients.id\_client

into :discid;

if (discid = null) then exit;

select discounts.discountvalue from discounts, clients where

clients.id\_discount = discounts.id\_discount and

new.id\_client = clients.id\_client

into :discval;

select TICKETTYPES.price from tickettypes where

tickettypes.id\_type = new.id\_type

into :fprice;

rprice = fprice - (fprice\*discval)/100;

new.paid = rprice;

end;

При добавлении абонемента с ID\_CLIENT = 4, у которого ID\_DISCOUNT = 3, что означает у этого клиента скидку 25%. Тип абонемента 1, т.е. тот, который стоит 10000 без скидки. Результат:



Видно, что клиент заплатил 7500 вместо 10000.

Второй триггер:

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

drop trigger free\_addserv;

create trigger free\_addserv for visits after insert

as

declare variable n\_visits int;

declare variable t\_visits int;

begin

n\_visits = 1;

select COUNT(visits.id\_client) from visits

where visits.id\_client = new.id\_client and

visits.visitdate between '2015-11-01' and '2015-11-21'

into :t\_visits;

if (t\_visits = n\_visits) then

execute procedure add\_free\_serv(new.id\_client,new.visitdate);

end;

Данный тригер вызывает следующую процедуру:

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

drop procedure add\_free\_serv;

create procedure add\_free\_serv(clientid int, bdate date)

as

declare variable popular\_id int;

declare variable counterr int;

begin

select first 1 addservices.id\_service,

count(accountingservices.id\_service) as Counter from addservices, accountingservices

where addservices.id\_service = accountingservices.id\_service

group by addservices.id\_service order by Counter desc into :popular\_id, :counterr;

insert into ACCOUNTINGSERVICES values (:clientid, :popular\_id, :bdate);

end;

1. **Вывод**

В данной работе были созданы триггеры. Триггеры полезно использовать для проверки корректности вносимых в БД данных и их целостности. С помощью триггеров можно выдавать сообщения (предупреждения) о том, что необходимо выполнить некоторые действия при изменении таблиц.