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

Выполнила студентка гр. 43501/1 Я.Н. Олефир

(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 route\_id в таблице Routes

connect 'D:\bd\avia.fdb' user 'SYSDBA' password 'masterkey';

--create generator increment;

--set generator increment to Routes.route\_id;

drop trigger auto\_fill;

create trigger auto\_fill for Routes before insert

as

begin

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

end;

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

connect 'D:\bd\avia.fdb' user 'SYSDBA' password 'masterkey';

create exception errror 'Error deleting aircraft type';

drop trigger check\_man;

create trigger check\_man for Airplanes before delete or update

as

begin

if (OLD.aircraft\_type in (select manufacturers.aircraft\_type from manufacturers)) then

exception errror;

end;

Индивидуальное задание:

Реализовать триггеры:

1. При возврате билета проверять является ли он возвратным. Если нет - не оформлять возврат.
2. При покупке билетов начислять бонусные мили.

Первый триггер:

connect ‘D:\bd\new\avia.fdb’ user ‘SYSDBA’ password ‘masterkey’;

drop trigger OK\_RETURN;

CREATE OK\_RETURN FOR RETURNTICK BEFORE INSERT

as

declare variable ttype int;

declare variable tprice float;

declare variable fprice5 float;

begin

tprice = new.price;

select tickets.ticket\_type\_id from tickets

where tickets.ticket\_id=new.ticket\_id

into :ttype;

fprice5=tprice\*0.5;

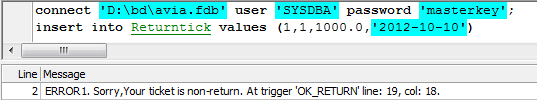
if (ttype=1) then exception error1;

else if (ttype=2) then new.price=:tprice;

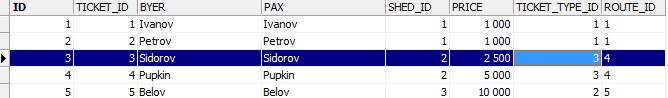
else if (ttype=3) then new.price= cast (:fprice5 as int);

end

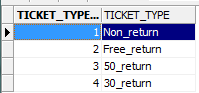
При добавлении возврата билета с типом «Невозвратный»



В таблице Ticket под номером Ticket\_id=3 находится билет с Ticket\_type\_id=3



То есть к возврату доступно только 50% стоимости билета.



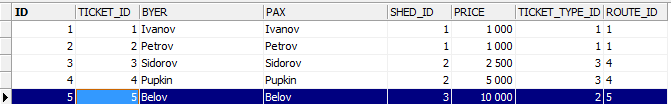
При добавлении такого типа билета в таблицу возвратов за



Результат работы триггера, в таблице Returntick стоимость возврата составила 50% первоначальной:



В таблице Ticket под номером Ticket\_id=5 находится билет с Ticket\_type\_id=2



Билет типа «Free return», то есть стоимость должна остаться без изменений.



Второй триггер:

connect ‘D:\bd\new\avia.fdb’ user ‘SYSDBA’ password ‘masterkey’;

drop trigger add\_miles;

CREATE OR ALTER TRIGGER ADD\_MILES FOR TICKETS

ACTIVE AFTER INSERT POSITION 0

as

declare variable tmiles int;

declare variable tstatus int;

declare variable tpaxid int;

begin

select priority.miles, priority.status\_id, priority.id from priority

where

priority.pax = new.pax

into :tmiles, :tstatus, :tpaxid;

if (tstatus=1) then tmiles=tmiles+1000;

if (tstatus=2) then tmiles=tmiles+2000;

if (tstatus=3) then tmiles=tmiles+3000;

execute procedure add\_miles (tmiles, tpaxid);

end

Данный тригер вызывает следующую процедуру:

connect ‘D:\bd\new\avia.fdb’ user ‘SYSDBA’ password ‘masterkey’;

drop procedure add\_miles;

create procedure ADD\_MILES (QMILES integer, QPAXID integer)

as

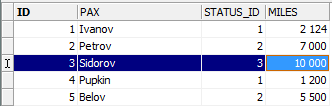
begin

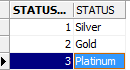
update priority set Miles = :qmiles

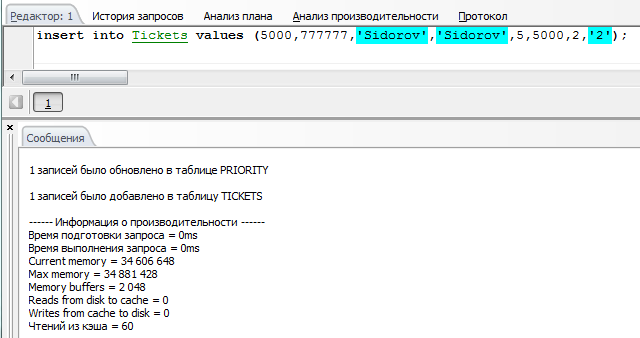
where :qpaxid=priority.id;

end

Пассажир Sidorov имеет Status\_id =3 (что соответствует Platinum статусу), при покупке билета его количество его миль должно увеличиться на 3000.



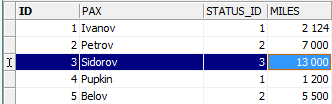




В таблице Tickets появилась запись о покупке нового билета.



И количество миль у пассажира Sidorov увеличилось на 3000.

****

1. **Вывод**

При выполнении данной лабораторной работы были изучены триггеры. Триггер – это хранимая процедура, которую пользователь не вызывает непосредственно, она вызывается при модификации данных: добавлением, удалением или изменением. Триггеры позволяют контролировать и изменять операции, производимые над таблицами базы данных. Они позволяют обеспечивать логическую целостность данных в соответствии с предметной областью. Триггеры могут вызывать до наступления события (ключевое слово before) или после него (after). С помощью триггеров можно генерировать сообщения о том, что необходимо выполнить некоторые действия при изменении таблиц. Были созданы триггеры, соответствующие индивидуальному заданию. При выполнении работы сложностей не возникло.