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 Чинь Тхань Нам

(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 film\_id в таблице film

connect 'D:/library\_films.fdb' user 'SYSDBA' password 'masterkey';

drop trigger auto\_fill;

--create generator increment;

--set generator increment to film.film\_id;

create trigger auto\_fill for film before insert

as

begin

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

end;

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

connect 'D:/library\_films.fdb' user 'SYSDBA' password 'masterkey';

create exception errror 'Error deleting film type';

drop trigger check\_cel;

create trigger check\_cel for film before delete or update

as

begin

if (OLD.film\_id in (select request.film\_id from request)) then

exception errror;

end;

Индивидуальное задание:

Реализовать триггеры:

1. При создании заявки. Если зритель уже смотрел данный фильм - показывать его бесплатно.
2. Если зритель в текущем месяце посмотрел более заданного количества фильмов - дарить ему бесплатный просмотр наиболее популярного фильма прошлого месяца.

connect 'D:/library\_films.fdb' user 'SYSDBA' password 'masterkey';

drop trigger sale\_viewer;

create trigger sale\_viewer for request before insert

as

declare variable v\_id int default 0;

declare variable fprice float;

declare variable rprice float;

begin

select first 1 request.viewer\_id from request , viewer

where new.viewer\_id = request.viewer\_id and

new.film\_id = request.film\_id

into :v\_id;

if (v\_id=0) then begin

rprice = new.price;

end

else begin

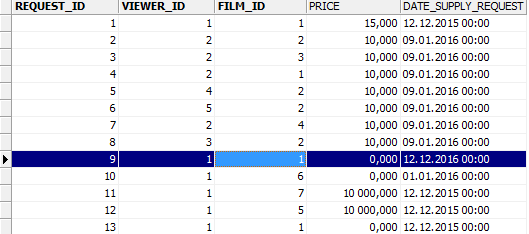
rprice = 0;

end

new.price = rprice;

end;

При добавлении заявки новой с REQUEST\_ID = 9, у которого VIEWER\_ID =1, FILM\_ID=1, то цена аналогично = 0 – бесплатно потому что человек это фильм смотрел . Результат:



insert into request values (9,1,6,10000,'2016-12-12')

Видно, что цена автоматически заменялась на 0.

Второй триггер:

connect 'D:/library\_films.fdb' user 'SYSDBA' password 'masterkey';

drop trigger free\_view;

create trigger free\_view for request after insert

as

declare variable tmonth int;

declare variable tkol int default 0;

declare variable kol int;

begin

kol = 2;

select first 1 extract(month from request.date\_supply\_request) from request where

new.date\_supply\_request = request.date\_supply\_request

into :tmonth;

select first 1 count (request.film\_id) as Counterr from request where

new.viewer\_id = request.viewer\_id and

(extract (month from request.date\_supply\_request)) = :tmonth

into :tkol;

if (tkol > kol) then

begin

execute procedure add\_free\_film(new.request\_id,new.viewer\_id,new.date\_supply\_request);

end

end;

Данный тригер вызывает следующую процедуру:

connect 'D:/library\_films.fdb' user 'SYSDBA' password 'masterkey';

drop procedure add\_free\_film;

create procedure add\_free\_film(requestid int, viewerid int, datesup date)

as

declare variable pop\_id int;

declare variable req\_id int;

declare variable new\_request int;

declare variable new\_date date;

begin

new\_date = :datesup;

select first 1 request.film\_id, count (request.request\_id) as Counter from request where

(extract(month from request.date\_supply\_request)) = (extract(month from :datesup)) -1

group by request.film\_id order by Counter desc

into :pop\_id, :req\_id;

new\_request = :requestid +1;

insert into gift\_film values (:new\_request,:viewerid,:pop\_id,0,:new\_date);

end;

Создание новой таблицы для подарков на бесплатные фильмы:

connect 'd:/library\_films.fdb'

user 'SYSDBA' password 'masterkey';

drop table gift\_film;

CREATE TABLE gift\_film (

id\_request int primary key,

viewer\_id int,

film\_id int,

Prices float,

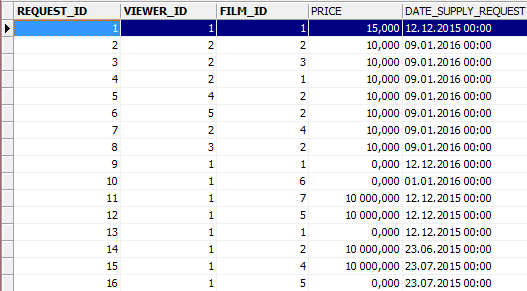
Date\_supply date

);

ALTER TABLE gift\_film ADD CONSTRAINT gift\_film\_fk0 FOREIGN KEY (Viewer\_id) REFERENCES Viewer(Viewer\_id);

ALTER TABLE gift\_film ADD CONSTRAINT gift\_film\_fk1 FOREIGN KEY (Film\_id) REFERENCES Film(Film\_id);

Таблица заявок:



Добавить в request:

insert into request values (17,1,6,10000,'2015-07-23')

В июне фильм с id = 2 самый популярный. И зритель смотрел больше 2 раза – подарить ему фильм этот.

В таблице gift\_film:



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.

Также триггеры удобно использовать для оповещения об изменении данных в таблицах.

С помощью триггеров можно накладывать ограничения на вносимые данные согласно требованиям предметной области БД.