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 Д. С. Раскин

(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 созданной для теста данного триггера, чтобы не испортить одну из таблиц БД(Для избежания ошибок, вместо получения максимального значения из ID таблицы будем использовать генератор):

--create generator incr;

set term ^;

create or alter trigger auto\_incr for TEST before insert

as

begin

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

end^

set term ;^

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

set term ^;

CREATE OR ALTER trigger Control for developers

active before delete or update

AS

begin

if (OLD.id in (select games.dev\_id from games)) then

exception;

end^

set term ;^

Индивидуальное задание:

Реализовать триггеры:

* 1. Ввести проверку при создании нового экземпляра турнира: если уже есть такой же турнир в таком же году - не добавлять.
  2. Не включать в турниры более трех игр одинакового жанра от одного разработчика.

Первый триггер:

set term ^;

CREATE OR ALTER trigger AlreadyHave for tournaments

create or alter exception already ’Tournament already has been held’^

active before insert or update

AS

begin

IF(NEW.tour\_id in (select tour\_id from tournaments where tour\_year = NEW.tour\_year))

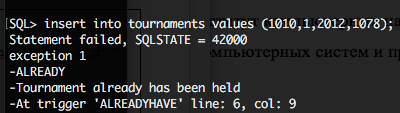
then

exception already;

end^

set term ;^

Пример, в 2012 году уже был турнир принадлежащей группе TOUR\_ID=1, при попытке добавить еще один такой же турнир получим:



Турнир не добавился.

Второй триггер:

set term ^;

create or alter exception toomuch 'Too much games from one developer and one genre'^

CREATE OR ALTER trigger TooMuchG for tournaments

active before insert

AS

begin

IF(EXISTS(select games.dev\_id, count(games.dev\_id) as games\_in\_genre

from games, tournaments where games.id = tournaments.game\_id

group by dev\_id, genre\_id having count(games.dev\_id) > 2))

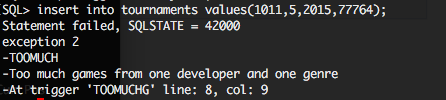
then

exception toomuch;

end^

set term ;^

Данный триггер срабатывает, если мы пытаемся добавить в турнир принадлежащей группе турниров игру определенного жанра от определенного разработчика, но в турнире уже есть, хотя бы 2 игры одного жанра от одного разработчика, и не дает нам этого сделать.



1. **Вывод**

В результате работы было проведено знакомство с триггерами Firebird. Было создано несколько стандартных триггеров, а так же реализованы триггеры в соответствие с индивидуальным заданием. Триггеры полезно использовать для проверки корректности вносимых в БД данных и их целостности. При данном контроле, или при добавлении записей в БД, при определенных условиях(условиях вызова триггера) выводятся диагностические сообщения(с помощью использования исключений).

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