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 А.Е. Балясников

(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 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

CREATE generator increment^

CREATE OR ALTER TRIGGER auto\_gen FOR action\_types BEFORE INSERT

AS

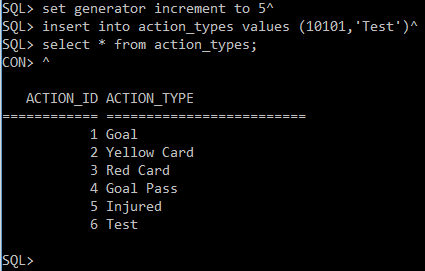
BEGIN

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

END^

Установим генератор на 5, так как в таблице уже есть 5 записей.

Добавим запись в таблицу и посмотрим на результат:



Затем был создан триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице stage\_type:

CREATE EXCEPTION ERROR\_STAGE 'ERROR: CANNOT DELETE STAGE TYPE'^

CREATE OR ALTER TRIGGER check\_stage FOR stage\_type BEFORE DELETE OR UPDATE

AS

BEGIN

IF (OLD.stage\_id IN (SELECT club\_cup.stage FROM club\_cup)) THEN

EXCEPTION ERROR\_STAGE;

END^

**Индивидуальное задание:**

Реализовать триггеры:

1. Вызов процедуры обновления турнирной таблицы после очередного состоявшегося матча.

2. Проверки консистентности данных в действиях игрока за матч: если он был удален, то после этого не может совершать действий, если забил гол, то не может быть автором голевой передачи для этого гола.

* 1. Обновление турнирной таблицы после очередного состоявшегося матча.

CREATE OR ALTER TRIGGER update\_league\_table FOR game AFTER INSERT

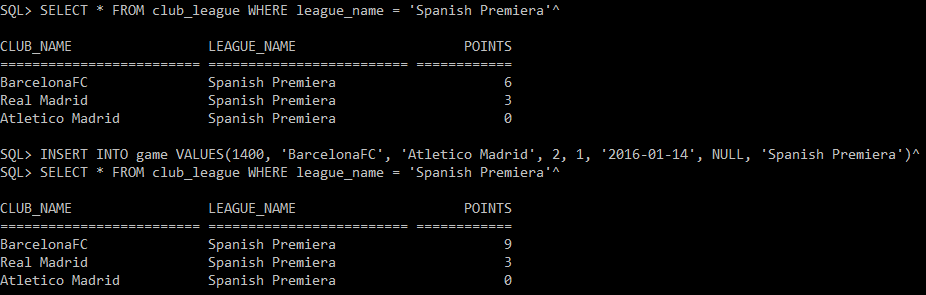
AS

BEGIN

EXECUTE PROCEDURE update\_league(NEW.game\_date, NEW.game\_date, NEW.league\_name);

END^

Результаты работы:



При добавлении записи в таблицу игр game, триггер срабатывает и автоматически обновляет таблицу результатов. В триггере просто вызывается хранимая процедура, которая и производит обновление таблицы.

* 1. Если игрок был удалён, то он больше не может совершать никаких действий.

Перед внесением записи в таблицу action будем проверять, есть ли уже этот игрок в этой таблице и получил ли он уже красную карточку. Если есть, то необходимо выдать EXCEPTION.

CREATE EXCEPTION RED\_CARD 'ERROR: THIS PLAYER CANNOT DO ANY ACTION, HE TOOK A RED CARD'^

CREATE OR ALTER TRIGGER player\_out FOR action BEFORE INSERT

AS

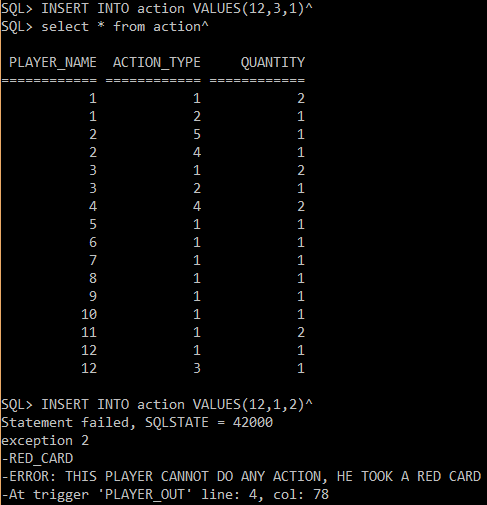
BEGIN

IF (NEW.player\_name IN (SELECT player\_name FROM action WHERE action\_type=3)) THEN

EXCEPTION RED\_CARD;

END^

Проверка:



* 1. Если игрок забил гол, то он не может быть автором голевой передачи для него.

Если игрок в данной игре забил гол, то, соответственно, он не может быть автором голевой передачи для него. Action\_type голевой передачи равен 4.

Соответственно, если есть запись о забитом голе игрока и она – единственная, то он не может быть автором голевой передачи для себя самого.

CREATE EXCEPTION GOAL\_AND\_PASS 'ERROR: ONE PLAYER CANNOT BE AUTHOR AND ASSISTENT FOR ONE GOAL'^

CREATE OR ALTER TRIGGER player\_goal\_assist FOR action BEFORE INSERT

AS

BEGIN

IF (NEW.player\_name IN (SELECT player\_name FROM action WHERE action\_type=1) AND NEW.action\_type=4) THEN

EXCEPTION GOAL\_AND\_PASS;

END^

Проверка:



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.