

Санкт-Петербургский государственный политехнический университет  
Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина:** Базы данных

**Тема:** Триггеры, вызовы процедур

Выполнил студент гр. 43501/3

\_\_\_\_\_  
(подпись) В.Е. Бушин

Руководитель

\_\_\_\_\_  
(подпись) А.В. Мяснов

“ \_ ” \_\_\_\_\_ 2015 г.

Санкт-Петербург

2015

## 1. Цель работы

Познакомить студентов с возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур и триггеров.

## 2. Программа работы

- 1) 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
- 2) Создать триггер в соответствии с индивидуальным заданием, полученным у преподавателя
- 3) Создать триггер в соответствии с индивидуальным заданием, вызывающий хранимую процедуру
- 4) Выложить скрипт с созданными сущностями в svn
- 5) Продемонстрировать результаты преподавателю

## 3. Ход работы

1)

- Триггер для автоматического заполнения ключевого поля id таблицы Player:

```
create generator autoinc;  
set generator autoinc TO Player.id;  
  
create trigger auto_trigger for Player before insert  
as begin  
    new.id=gen_id(autoinc, 1);  
end;
```

- Триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице:

```
create trigger data_control for club before delete or update  
as begin  
    if(old.id in (select Player.clubid from Player)) then  
        exception data_error;  
    end;
```

- 2) В соответствии с индивидуальным заданием создан триггер, который обновляет турнирную таблицу после завершения всех матчей тура:

```
create trigger champ_trig for Championship before update  
as  
declare variable cnt int;  
begin  
    select count(Match_Champ.tour) from Match_champ where  
    Match_champ.tour=new.played into :cnt;  
    if(cnt=10) then  
        execute procedure update_champ(new.played);  
    end;
```

- 3) В соответствии с индивидуальным заданием создан триггер, который делает невозможным регистрацию действий игрока в матче, если после получения четырех желтых карточек в одном турнире он не пропустил ни одной игры в этом турнире:

```
create exception lot_of_cards 'This player has more than 4 yellow cards';

create trigger yellowcard_trig for Act before insert
as
declare variable cards int;
begin
    select count(Act.acttype) from Act, Player where Act.acttype="yellow card"
        and Player.id=Act.playerid and Player.id=new.playerid into :cards;
    if(cards>4) then
        exception lot_of_cards;
    end;
```

#### **4. Вывод**

Во время работы изучены триггеры и генераторы. Триггер – это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных. Триггеры позволяют контролировать и изменять операции, проводимые над таблицами БД. Например, триггеры позволяют обеспечивать целостность данных.

Генератор – это специальный объект базы данных, который генерирует уникальные последовательные числа. Одним из применений генераторов является их использование в триггерах автоинкрементации ключей. В таких триггерах необходимо использовать генераторы, потому что они обеспечивают уникальность генерируемых значений даже при параллельной обработке нескольких запросов.