

Федеральное государственное автономное образовательное учреждение  
высшего образования «Санкт-Петербургский политехнический университет  
Петра Великого»  
Институт компьютерных наук и технологий  
Кафедра Компьютерных систем и программных технологий

**Отчёт по лабораторной №6**

Дисциплина: Базы данных

Тема: Триггеры

Выполнил студент гр. 43501/4

\_\_\_\_\_  
(подпись) В.С. Филиппов

Руководитель

\_\_\_\_\_  
(подпись) А.В. Мяснов

“ ” \_\_\_\_\_ 2015 г.

Санкт -Петербург

2015

## Цель работы

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

## 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

## Выполнение работы

```
/*1 автоинкремент см. лаб 1*/

/*1 триггер на изменение или удаление*/
CREATE EXCEPTION ex_no_modify 'This LEAGUE in table SEASONS';
CREATE TRIGGER tr_modif_league_id FOR leagues
BEFORE delete or update
AS
BEGIN
    if(old.league_id
        in (select seasons.league_id from seasons))
    then exception ex_no_modify;
END;
/*
вывод триггера:

delete from leagues where league_id = 7;
EX_NO_MODIFY.
This LEAGUE in table SEASONS.
At trigger 'TR_MODIF_LEAGUE_ID' line: 8, col: 5.

*/
```

## Индивидуальные задания

```
/*
1
При получении второй желтой карточки давать красную.
*/
CREATE TRIGGER tr_two_yellow_in_red FOR events
BEFORE insert
AS
BEGIN
    if (new."EVENT" = 'YELLOW')
    then
        if ('YELLOW' in
            (select events."EVENT" from events
             where events.player_id = new.player_id
              and events.match_id = new.match_id))
        then
```

```

        execute procedure ins_event(new.player_id, new.match_id,
new.event_minute, 'RED', 0);
END;

```

226	153	27 GOAL	0
258	392	9 RED	0
25	546	14 YELLOW	0
82	289	88 REPLACEMENT	0
495	687	62 RED	0
217	438	40 REPLACEMENT	0

Добавим желтую, тому у кого уже есть:

```

/*execute procedure ins_event(25,546,22,'YELLOW',0);*/

```

Результат:

```

/*select * from events where player_id = 25 and events.match_id = 546;*/
/*

```

ВЫВОД:

PLAYER_ID	MATCH_ID	EVENT_MINUTE	EVENT	AG_IO
25	546	22	RED	0
25	546	22	YELLOW	0
25	546	14	YELLOW	0

```

*/

```

Видно, что при добавлении события «жёлтая карточка», если она уже 2ая у игрока в этом матче, то генерируется красная карточка.

```

/*

```

```

2

```

```

При изменении данных в таблице ГОЛОВ изменять данные в записи о матче.
*/

```

```

create or alter procedure add_goal_to_matches (pid integer, mid integer, ag
CHAR(1))
as
begin

```

```

        if ((select count(*) from current_club_players_list, matches
            where matches.match_id = :mid
            and current_club_players_list.club_id = matches.first_club_id
            and current_club_players_list.player_id = :pid) > 0)
        then
            if (ag = 0) then
                update matches set first_club_goals = first_club_goals + 1
where match_id = :mid;
            else
                update matches set second_club_goals = second_club_goals + 1
where match_id = :mid;

        if ((select count(*) from current_club_players_list, matches
            where matches.match_id = :mid
            and current_club_players_list.club_id = matches.second_club_id
            and current_club_players_list.player_id = :pid) > 0)
        then
            if (ag = 1) then
                update matches set first_club_goals = first_club_goals + 1
where match_id = :mid;
            else
                update matches set second_club_goals = second_club_goals + 1
where match_id = :mid;
        end;

```

```
CREATE or alter TRIGGER tr_add_goal_to_matches FOR events
BEFORE insert
AS
BEGIN
    if (new."EVENT" = 'GOAL')
    then
        execute procedure add_goal_to_matches(new.player_id, new.match_id,
new.ag_io);
END;
```

Выберем матч со счётом 0:2:

```
--execute procedure ins_event(1, 1, 90, 'GOAL', 0);
/*гол Дзюбы, гол 2ой команды*/
```

MATCH_...	MATCH_DATE	SEASON_ID	FIRST_CLUB_ID	SECOND_CLUB_ID	FIRST_CLUB_PERCENT	FIRST_CLUB_SHORTS	SECOND_CLUB_SHORTS	FIRST_CLUB_GOALS	SECOND_CLUB_GOALS
1	26.09.2015	1	2	1	<null>	<null>	<null>	0	2
2	30.11.2015	1	3	1	<null>	<null>	<null>	1	1
3	30.11.2015	1	2	3	<null>	<null>	<null>	1	0
5	14.09.2015	4	409	77	9	213	618	12	38
6	27.02.2014	6	251	469	86	364	561	14	31
7	08.03.2014	7	161	436	19	45	198	1	27

Добавим гол, счёт станет 0:3:

```
execute procedure ins_event(1, 1, 90, 'GOAL', 0);
/*гол Дзюбы, гол 2ой команды*/
```

MATCH_...	MATCH_DATE	SEASON_ID	FIRST_CLUB_ID	SECOND_CLUB_ID	FIRST_CLUB_PERCENT	FIRST_CLUB_SHORTS	SECOND_CLUB_SHORTS	FIRST_CLUB_GOALS	SECOND_CLUB_GOALS
1	26.09.2015	1	2	1	<null>	<null>	<null>	0	3
2	30.11.2015	1	3	1	<null>	<null>	<null>	1	1
3	30.11.2015	1	2	3	<null>	<null>	<null>	1	0

Добавим автогол, счёт станет 1:3:

```
execute procedure ins_event(1, 1, 90, 'GOAL', 1);
/*гол Дзюбы, в свои ворота*/
/*
```

MATCH_...	MATCH_DATE	SEASON_ID	FIRST_CLUB_ID	SECOND_CLUB_ID	FIRST_CLUB_PERCENT	FIRST_CLUB_SHORTS	SECOND_CLUB_SHORTS	FIRST_CLUB_GOALS	SECOND_CLUB_GOALS
1	26.09.2015	1	2	1	<null>	<null>	<null>	1	3
2	30.11.2015	1	3	1	<null>	<null>	<null>	1	1
3	30.11.2015	1	2	3	<null>	<null>	<null>	1	0
5	14.09.2015	4	409	77	9	213	618	12	38

Видно, что при добавлении события «гол», счёт матча обновляется.

## Вывод

Триггеры позволяют автоматизировать процесс заполнения базы данных. Если в одной из таблиц происходит операция, то триггер способен создать запись в другой таблице, изменить запись, или не позволить эту операцию.

Триггеры асинхронно реагируют на изменения в системе. Они позволяют контролировать корректность вводимых данных непосредственно в момент их добавления. Это исключает периодические проверки корректности данных в БД.