

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Институт компьютерных наук и технологий
Кафедра Компьютерных Систем и Программных Технологий**

**Отчет по дисциплине
«Базы данных»**

SQL-программирование: триггеры, вызовы процедур.

Работу выполнил студент группы №: 43501/1

Работу принял преподаватель: _____

Карандашов С.М.

Мяснов А.В.

**Санкт-Петербург
2016 г.**

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

Ход работы:

1. Создание триггера для автоматического заполнения ключевого поля

Был создан генератор GEN_OPTIONS_ID:

```
CREATE SEQUENCE GEN_OPTIONS_ID
```

Далее был создан триггер, использующий данный генератор:

```
CREATE OR ALTER trigger options_bi for razrab
active before insert position 0
as
declare variable tmp DECIMAL (18 ,0);
begin
  if (new.id is null) then
    new.id = gen_id(gen_options_id, 1);
  else
    begin
      tmp= gen_id (gen_options_id, 0);
      if(tmp <new. pers ) then
        tmp= gen_id (gen_options_id, new. pers -tmp);
      end
    end
end
```

2. Создание триггера для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице.

Триггер:

```
create or alter trigger user_delete_update for genres
before delete or update
as
begin
    if( old. genre_types in
        (select genre from game_genre))
        then exception ex_modify_genre;
end
```

При попытке удалить строчку в genres, ключ которой содержится в подчиненных таблицах, произойдет ошибка и будет выведено исключение.

3. Исключить дублирование связей разработчиков со студиями. При попытке внесения дубли выбрасывать исключение.

Триггер:

```
CREATE OR ALTER TRIGGER DUPLICATE_SVYAZ FOR RAZRAB
ACTIVE BEFORE UPDATE OR DELETE POSITION 0
as
begin
    if( new. studio in
        (select studio from razrab))
        then exception ex_svyaz;
end
```

4. При добавлении данных о призовых участникам турнира проверять, что сумма призовых с учетом добавляемых данных не превышает призовой фонд турнира. Если превышает - добавлять разницу между призовым фондом и суммой выданных призов.

Вспомогательная процедура, которая проверяет состояние баланса:

```
create or alter procedure raznost(
cont int,
new_nagrad int)
as
declare variable sum_nagrad integer;
declare variable fond integer;
begin
    select contests.prize_fond from contests into :fond;
    select sum(winners.prize) from winners where winners.contest=:cont into :sum_nagrad;
    if (:sum_nagrad+:new_nagrad>:fond) then
```

```
update contests set prize_fond = :sum_nagrad+:new_nagrad
where id=:cont;
end
```

Триггер:

```
CREATE OR ALTER TRIGGER RAZN FOR WINNERS
ACTIVE AFTER UPDATE POSITION 0
AS
begin
    execute procedure raznost(new.contest, new.prize);
end
```

Вывод:

В результате работы было проведено знакомство с триггерами. Триггер можно считать автоматической процедурой, срабатывающей на стороне сервера в результате некоторого события. Триггер может срабатывать до наступления события или после его наступления.

Главным преимуществом триггеров является контроль целостности базы данных любой сложности. Так же упрощается приложение, так как часть логики уже выполняется на сервере.

Однако, у триггеров есть и недостатки. Например, большое количество триггеров сильно уменьшает производительность системы.