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 С.В. Смирнов

(подпись)

Руководитель А.В. Мяснов

(подпись)

“ ” 2016 г.

Санкт-Петербург

2016

# Цели работы

Познакомить студентов с возможностями реализации более сложной обработки данных на стороне сервера с помощью 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

# Ход работы

Триггер для автоматического заполнения поля PartID в таблице Parts:

CREATE generator increment^

CREATE OR ALTER TRIGGER auto\_gen FOR Parts BEFORE INSERT

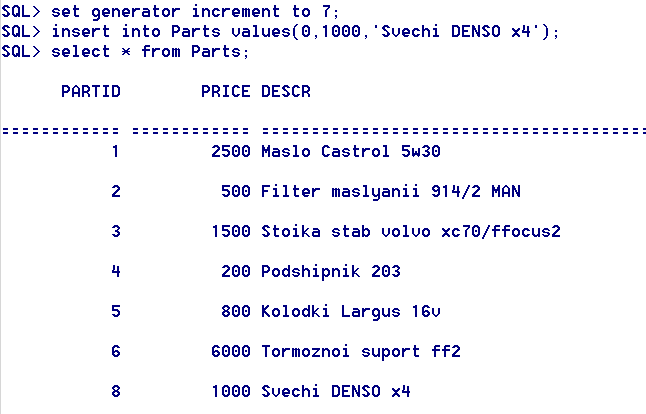
AS

BEGIN

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

END^

Результат тестирования:



Триггер для контроля целостности данных в подчиненной таблице works при удалении/изменении записей в главной таблице ID\_Works.

Код:

CREATE EXCEPTION ERROR\_STAGE 'ERROR\_1: CANNOT DELETE STAGE TYPE'^

CREATE OR ALTER TRIGGER CHECK\_TRIGGER FOR ID\_Works BEFORE DELETE OR UPDATE

AS

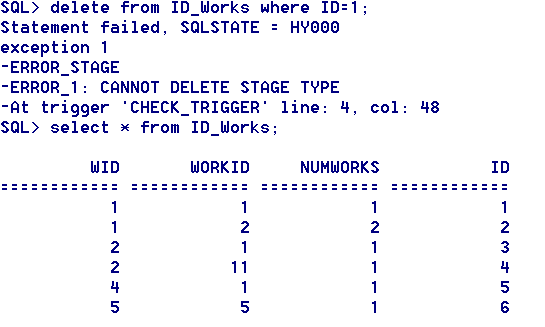
BEGIN

IF (OLD.WorkID IN (SELECT WorkID FROM Works)) THEN

EXCEPTION ERROR\_STAGE;

END^

Результат выполнения(попытка удалить список работ с ID=1):



**Индивидуальные задания:**

1. При добавлении новой запчасти в заказ вызывать процедуру 2 из [#236](http://tiger.ftk.spbstu.ru/trac/edu-db-2016/ticket/236).

(По заданному набору запчастей формировать рекомендуемый набор работ по имеющимся данным).

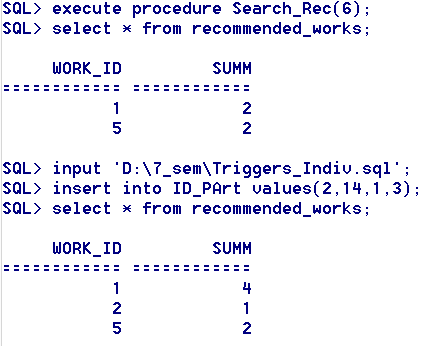
Код :

CREATE OR ALTER TRIGGER RECOMEND\_TRIGGER FOR ID\_Part AFTER INSERT

AS BEGIN

execute procedure Search\_Rec(new.PID);

END^



1. Пересчитывать сумму заказа при добавлении/изменении/удалении работ и запчастей из заказа.

CREATE OR ALTER PROCEDURE Update\_Price( WPID INT)

--returns (nums int)

AS

DECLARE VARIABLE nums int;

DECLARE VARIABLE sum1 int;

DECLARE VARIABLE sum2 int;

BEGIN

select sum(Works.Price\*Id\_Works.numWorks )

from Id\_Works natural join Works

where Id\_Works.ID in(

select Id from Id\_works where Id\_works.WID= :WPID)

into :sum1;

select sum(Parts.Price\*Id\_PArt.NumParts )

from ID\_Part natural join Parts

where ID\_Part.ID in (

select Id from Id\_part where Id\_part.PID= :WPID)

into :sum2;

nums= :sum1 + :sum2;

update Ser\_Info set Ser\_Info.Price=:nums where Ser\_Info.WPID= :WPID;

END

^

CREATE OR ALTER TRIGGER Update\_TRIGGER1 FOR ID\_Works AFTER INSERT or UPDATE or DELETE

AS

begin

if(DELETING) then

begin

execute procedure Update\_Price(old.WID);

end

if(UPDATING or INSERTING)then

begin

execute procedure Update\_Price(new.WID);

end

end ^

CREATE OR ALTER TRIGGER Update\_TRIGGER2 FOR ID\_Part AFTER INSERT or UPDATE or DELETE

AS

begin

if(DELETING) then

begin

execute procedure Update\_Price(old.PID);

end

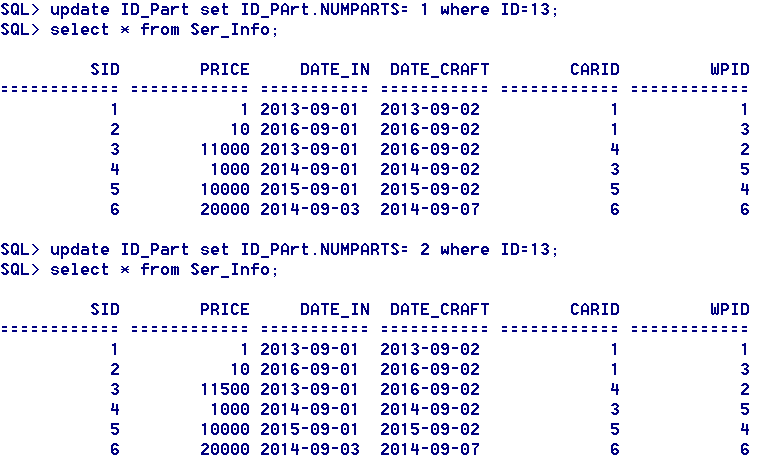
if(UPDATING or INSERTING)then

begin

execute procedure Update\_Price(new.PID);

end

end ^



**Выводы:**

В лабораторной работе мы с помощью триггеров выдавали 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.