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 Калугина М.О.

Руководитель Мяснов А.В.

“ ” 2016 г.

Санкт -Петербург

2016

1. **Цели работы**

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

# Программа работы

* 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
* Создать триггер в соответствии с индивидуальным заданием, полученным у преподавателя
* Создать триггер в соответствии с индивидуальным заданием, вызывающий хранимую процедуру

1. **Ход работы**

Триггер является программой, которая хранится в области метаданных базы данных и выполняется на стороне сервера. Напрямую обращение к триггеру невозможно. Он вызывается автоматически при наступлении одного или нескольких событий, относящихся к одной конкретной таблице (к представлению), или при наступлении одного из событий базы данных.

Триггер, вызываемый при наступлении события таблицы, связан с одной таблицей или представлением, с одним или более событиями для этой таблицы или представления (INSERT, UPDATE, DELETE) и ровно с одной фазой такого события (BEFORE или AFTER).

Триггер выполняется в той транзакции, в контексте которой выполнялась программа, вызвавшая соответствующее событие. Исключением являются триггеры, реагирующие на события базы данных. Для некоторых из них запускается транзакция по умолчанию.

1. **Задание**
   1. **Триггер для автоматического заполнения ключевого поля**

create generator increment^

create or alter trigger gener for Clients

before insert

as

begin

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

end^

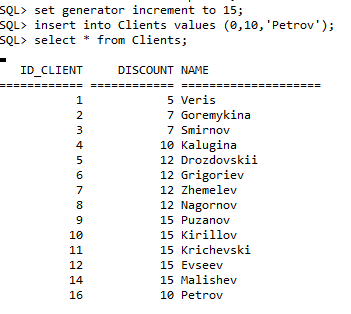


Рис.1. Результат выполнения триггера

1. **Триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице**

**create** **exception** error\_stage1 'Error: cannot delete this'**^**

**create** **or** **alter** **trigger** Er\_trig **for** Clients

**before** **delete** **or** **update**

**as**

**begin**

**If** **(old.**ID\_client **in** **(select** ID\_client **from** Orders**))** **then**

**exception** error\_stage1**;**

**end^**

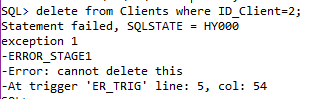


Рис.2. Результат выполнения триггера

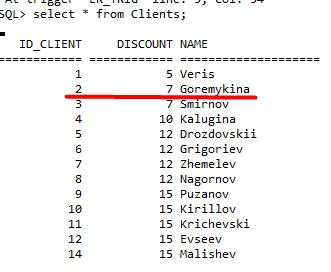


Рис.3. Результат выполнения триггера

## 5. Индивидуальное задание

1. **Проверка на дубли для противопоказаний и побочных действий. При добавлении дубля - выбрасывать исключение.**

**create** **exception** error\_stage2 'Error: this Constraindication already exists'**^**

**create** **exception** error\_stage3 'Error: this Side\_effect already exists'**^**

**create** **or** **alter** **trigger** Err\_doubleC **for** Constraindications

**before** **insert** **or** **update**

**as**

**begin**

**If** **(new.**Name **in** **(select** Name **from** Constraindications**))** **then**

**exception** error\_stage2**;**

**end^**

**create** **or** **alter** **trigger** Err\_doubleS **for** Side\_effects

**before** **insert** **or** **update**

**as**

**begin**

**If** **(new.**Name **in** **(select** Name **from** Side\_effects**))** **then**

**exception** error\_stage3**;**

**end^**

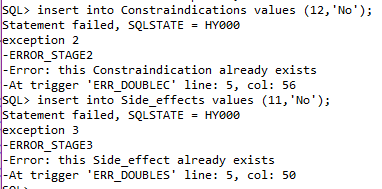


Рис.4. Результат выполнения триггера

1. **При удалении из заказа последней позиции удалять заказ.**

**create** **or** **alter** **procedure** Check\_order

**as**

**begin**

**delete** **from** Orders **where** ID\_order **not** **in**

**(select** ID\_ord **from** Ord\_Med**);**

**end**

**^**

**create** **or** **alter** **trigger** Del\_order **for** Ord\_med

**after** **delete**

**as**

**begin**

**execute** **procedure** Check\_order**;**

**end**

**^**

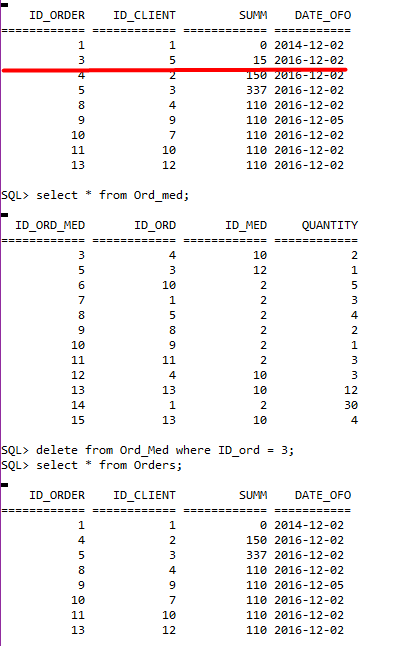


Рис.4. Результат выполнения триггера

1. **Вывод**

С помощью триггеров можно легко выдавать предупреждения для пользователя или вызывать процедуры, совершать действия при модификации определенных данных. С помощью триггеров можно накладывать ограничения на вносимые данные согласно требованиям предметной области БД.