Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра Компьютерных Систем и Программных Технологий

**ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №5**

Дисциплина: **Базы данных**

# Тема: Триггеры

А.М. Кириллов

А.В. Мяснов

Выполнил студент гр. № 43501/1

Преподаватель

Санкт-Петербург

2016

# Цель работы

Ознакомиться 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

# Ход 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

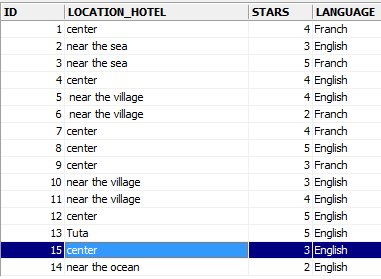
1.1 Создан триггер для автоматического заполнения поля id в таблице Hotel

|  |
| --- |
| create generator create\_id;  set generator create\_id to 15;  create trigger auto\_create\_id for hotel before insert as  begin new.id=gen\_id(create\_id, 1);  end |

Добавим новую строку в таблице без заполнения поля id

# 

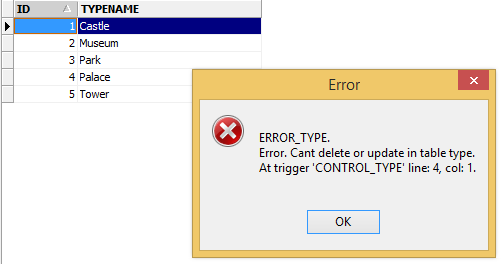
После обновления таблицы не заполненное поле id автоматически заполнилось



1.2 Создан триггер для контроля целостности

|  |
| --- |
| create exception error\_type 'Error. Cant delete or update in table type';  create trigger control\_type for type before delete or update  as begin  if(old.id in (select type\_id from place\_of\_interest))  then exception error\_type;  end |

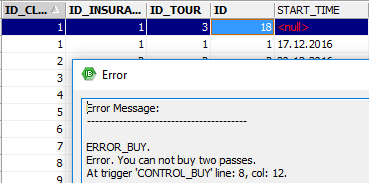
Попытаемся удалить строку в таблице Type:



2. Не давать возможность одному клиенту купить какой-либо тур два раза.

|  |
| --- |
| create exception Error\_buy  'Error. You can not buy more than two passes.';  create trigger control\_buy for trip before insert  as  declare variable tor integer;  begin  tor=-1  select id\_tour from trip  where(id\_tour=new.id\_tour)and(id\_client=new.id\_client) into: tor;  if(tor=-1) then  exception Error\_buy;  end |

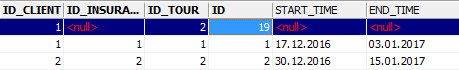
Добавили клиента с двумя турами



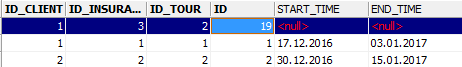
3. При создании путевки, если не создано ни одной страховки - создавать автоматически страховку.

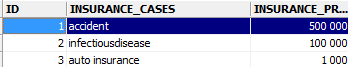
|  |
| --- |
| CREATE OR ALTER TRIGGER AUTOINSURANCE FOR TRIP  ACTIVE BEFORE INSERT POSITION 0  as  declare variable insurance\_id integer;  begin  insurance\_id = gen\_id(ID\_INSURANCE, 1);  insert into insurance values (:insurance\_id, 'auto insurance', 1000);  update insurance  set new.id\_insurance = :insurance\_id ;  end |

Добавили путевку без страховки



После нажатия команды insert





**Выводы:**

В данной работе были изучены триггеры и генераторы.

Триггер – это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных.

Это мощный инструмент для сохранения целостности и повышения безопасности базы данных.