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			Нгуен	Тиен Ву
J	(подпись)		J	J
Руководитель			A.B. N	Ляснов
	(подпись)			
		"	"	2016 г

Санкт-Петербург 2016

1. Цель

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

2. 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

3. Выполнение работы

Был создан триггер для автоматического заполнения поля id_client в таблице созданной для client данного триггера, чтобы не испортить одну из таблиц БД(Для избежания ошибок, вместо получения максимального значения из ID таблицы будем использовать генератор):

```
--create generator incr;
set term ^;
create or alter trigger auto_incr for CLIENT before insert
as
begin
  new.id = id_client(incr,1);
  end^
set term ;^
```

Затем был создан триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице ADRESS

```
set term ^;
CREATE OR ALTER trigger Control for ADRESS
active before delete or update
AS
begin
   if (OLD.id in (select adress.city_id from adress)) then
      exception;
end^
```

Индивидуальное задание:

Реализовать триггеры:

- 1. При выплате по страховке проверять есть ли действующий договор на данный тип страхового случая. Если нет не добавлять.
- 2. При покупке нового полиса проверять нет ли действующего полиса аналогичного типа. Если есть не добавлять

Первый триггер:

```
set term ^;
CREATE OR ALTER trigger checkcase for insurance_case_con
active before insert position 0
as
begin
  if (current_date not between (select date_period from sign_contract
  where id_contract = sign_contract.id_contract) and (select dat_term
from sign_contract
  where sign_contract.id_contract = id_contract))
  then
  exception errcase;
endset term;^
```

Работа скрипта:

select id_contract, date_period,dat_term from sign_contract;

	ID_CONTR	DATE_PERIOD	DAT_TERM
▶	1	01.01.2014	01.01.2015
	2	03.04.2014	03.04.2015
	3	08.07.2014	08.07.2015
	10	24.07.2015	24.07.2016
	4	21.08.2014	21.08.2015
	5	03.11.2014	03.11.2015
	6	05.02.2015	05.02.2016
	8	12.04.2015	12.04.2016
	7	09.03.2015	09.03.2916
	9	15.05.2015	15.05.2016
	11	12.09.2015	12.09.2016
	12	01.11.2015	01.11.2016
	13	09.12.2015	09.12.2016
	14	28.12.2015	28.12.2016

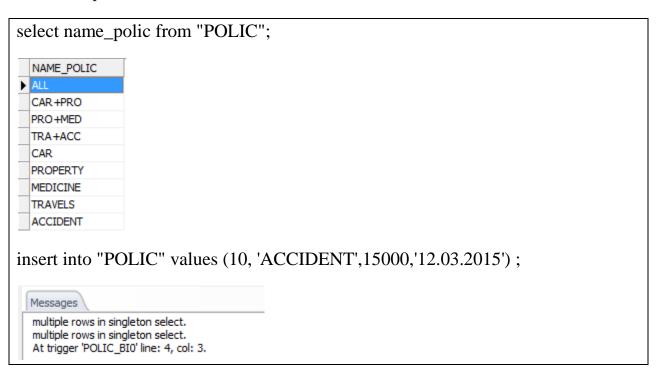
insert into insurance_case_con values (30,2,3,33);

multiple rows in singleton select. multiple rows in singleton select. At trigger 'CHECKCASE' line: 5, col: 3.

Второй триггер:

```
set term ^;
CREATE OR ALTER trigger polic_bi0 for polic
active before insert position 0
as begin
  if (new.name_polic like (select name_polic from "POLIC"
  where "POLIC".id_polic <> new.id_polic))
  then
  exception;
end
set term;^
```

Работа скрипта:



4. Вывод

В результате работы было проведено знакомство с триггерами Firebird. Было создано несколько стандартных триггеров, а так же реализованы триггеры в соответствие с индивидуальным заданием. Триггеры полезно использовать для проверки корректности вносимых в БД данных и их целостности. При данном контроле, или при добавлении записей в БД, при определенных условиях (условиях вызова триггера) выводятся диагностические сообщения (с помощью использования исключений).

С помощью триггеров можно накладывать различные ограничения на вносимые данные согласно требованиям предметной области БД.