

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

(подпись) Дао Суан Хоа

Руководитель

(подпись) А.В. Мяснов

“__” _____ 2015 г.

Санкт-Петербург

2015

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_sport в таблице Sport

```
create trigger id_increment for sport before insert
as
begin
    select max(sport.id_sport) from sport into NEW.id_sport;
    NEW.id_sport = NEW.id_sport + 1;
end;
```

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

```
create exception error 'Error deleting ticket type';
```

```
CREATE OR ALTER TRIGGER CHECK_CEL FOR SEASONTICKET_TYPE
active before update or delete position 0
as
begin
    if (OLD.id_seasonticket_type in (select
season_ticket.id_seasonticket_type
from season_ticket)) then
        exception error;
end
```

Индивидуальное задание:

1. При продаже абонеента проверять нет ли действующего абонеента такого же типа для данного клиента. Если есть - не регистрировать продажу.

1. При продаже абонеента проверять нет ли действующего абонеента такого же типа для данного клиента. Если есть - не регистрировать продажу.

```
create exception error_buy111khk 'Cannot buy type that already in deal';
create trigger checbuy before insert on season_ticket
as
begin
    if(new.id_seasonticket_type not in (select season_ticket.id_seasonticket_type from
season_ticket
    where season_ticket.id_client = new.id_client)) then
        exit;
    else exception error_buy;
end
```

Добавлении 3 абонеента в таблице season_ticket

```
insert into season_ticket values (20,4,9);
insert into season_ticket values (21,3,6);
insert into season_ticket values (22,5,7);
```

ID_SEASON_T...	ID_SEASONTICKET_TYPE	ID_CLIENT
22	5	7
21	3	6
20	4	9

При добавлении тир абонеента с ошибки

```
connect 'D:/xuanhoa.fdb' user 'SYSDBA' password 'masterkey';
insert into season_ticket values (19,8,5);
```

Line	Message
2	ERROR_BUY. Cannot buy any more . At trigger 'CHECBUYT' line: 7, col: 5.

2. При продаже абонеента на определенный вид спорта проверять нет ли среди действующих абонементов данного клиента возможности ходить на данный вид спорта. Если есть - не добавлять.

```
drop trigger checkghhd ;
```

```
create trigger checkghhd for sport_tickettype before insert
```

```
as
```

```
declare variable spo int;
```

```
declare variable tic int;
```

```
begin
```

```
select sport_tickettype.id_sport , sport_tickettype.id_tickettype
```

```
from sport_tickettype where sport_tickettype.id_tickettype = new.id_tickettype
```

```
into spo,tic;
```

```
if(new.id_sport in (spo ) and tic <4 ) then exception error_spo;
```

```
else exit;
```

```
end
```

Добавлении абонеента с sport_tickettype.id_tickettype > 3 в таблице sport_tickettype

```
connect 'D:/xuanhoa.fdb' user 'SYSDBA' password 'masterkey';
insert into sport_tickettype values (21,11,11);
```

При добавлении тир абонеента с ошибки

```
connect 'D:/xuanhoa.fdb' user 'SYSDBA' password 'masterkey';
insert into sport_tickettype values (21,1,1);
```

Line	Message
2	multiple rows in singleton select. multiple rows in singleton select. At trigger 'CHECKGHHD' line: 7, col: 9.

4. Вывод

В данной работе были созданы триггеры. Триггеры полезно использовать для проверки корректности вносимых в БД данных и их целостности. С помощью триггеров можно выдавать сообщения (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.