**Федеральное агентство по образованию**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №6**

# «Триггеры, вызовы процедур»

**Студент гр.43501/1: Русаков Р.В.**

**Преподаватель: Мяснов А.В.**

**Санкт-Петербург**

**2014**

1. **Цель работы**

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

1. **Проведение работы**

Создать триггер для автоматического заполнения ключевого поля:

CREATE SEQUENCE GEN;

ALTER SEQUENCE GEN RESTART WITH 0;

Триггер для заполнения в таблице orders:

SET TERM // ;

CREATE OR ALTER TRIGGER GEN FOR orders

ACTIVE BEFORE INSERT POSITION 0

AS

begin

IF (new.tickets\_id IS NULL) THEN

NEW.order\_id = GEN\_ID(GEN, 1);

end

^

SET TERM // ^

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

Триггер, удаляющий строки orders\_entre , входящие в отмененный заказ:

SET TERM ^ ;

CREATE OR ALTER TRIGGER DELETE\_TICKETS\_IN\_ORDER FOR ORDERS

ACTIVE BEFORE DELETE POSITION 0

AS

begin

/\* Trigger text \*/

delete from tickets where tickets. order\_id=old.order\_id;

end

^

SET TERM ; ^

1. **Триггер для :При попытке удалять заказ не давать этого сделать, если его статус более заданного**

CREATE EXCEPTION wrong\_state\_del 'Заказ нельзя удалить - он ещё активен' ;

SET TERM ^ ;

CREATE OR ALTER TRIGGER lab5\_1 FOR ORDERS

ACTIVE BEFORE DELETE POSITION 0

as

declare variable stat integer;

begin

stat=old.status;

if(stat!=0) then exception wrong\_state\_del;

end

^

SET TERM ; ^

1. **При создании более, чем на заданную сумму, выдавать клиенту подарок**

CREATE OR ALTER TRIGGER lab5\_2 FOR orders\_entre

ACTIVE BEFORE INSERT position 0

as

declare variable cost FLOAT;

declare variable d\_cost FLOAT;

declare variable gift\_id INT;

begin

if(new.book\_price!=0) then

begin

execute procedure lab5(new.order\_id) returning\_values :cost;

d\_cost=new.book\_price\*new.book\_cnt;

if(((:cost+:d\_cost)>300) and (cost <700)) then

begin

select first 1 book\_id from top100 order by top100\_id into :gift\_id;

insert into orders\_entre values(new.order\_id, :gift\_id,1,0 );

end

end

end

Хранимая процедура – считает сумму заказа

SET TERM \\ ;

CREATE PROCEDURE lab5 (new\_order\_id FLOAT)

returns (cost FLOAT)

AS

declare variable cnt INT;

declare variable price FLOAT;

declare variable t\_cnt INT;

declare variable sm FLOAT ;

declare variable t\_price FLOAT ;

BEGIN

sm=0;

price=0;

for select book\_cnt, book\_price from orders\_entre where order\_id=:new\_order\_id

into :t\_cnt,:t\_price

do

begin

price=:t\_cnt\*:t\_price;

sm=:sm + :price;

end

cost=:price;

END

\\

SET TERM ; \\

1. Вывод

В ходе работы были изучены основы работы с триггерами. Триггеры являются мощным средством контроля целостности, позволяя контролировать удаление и изменение данных. Так же с их помощью можно фиксировать изменения данных и немедленно реагировать на них, изменяя зависящие от них данные, автоматически поддерживая актуальность всей БД.