**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №6**

**«SQL-программирование: триггеры, вызовы процедур»**

**Информационное обеспечение систем управления**

Студент гр. 43501/1 Красильников В.В.

Преподаватель Мяснов А.В.

Санкт-Петербург

2015

# 1. Цель работы

Ознакомиться 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

# 3. Общие примеры триггеров

Триггер привязан к конкретной таблице и выполняется до или после заданного типа операции над ней. Если триггеров у одной таблицы несколько, порядок их выполнения можно задавать с помощью position.

## 3.1 Триггер для автоматического заполнения ключевого поля таблицы person.

Ранее был создан генератор идентификаторов для таблицы:

create sequence gen\_ppl\_id;

ALTER SEQUENCE gen\_ppl\_id RESTART WITH 20000;

Триггер:

set term ^ ;

create or alter trigger tr\_gen\_people\_id for PEOPLES

active before insert position 0

as

begin

if (new.ID\_PPL is null) then

begin

new.ID\_PPL = gen\_id(gen\_ppl\_id, 1);

end

end ^

set term ; ^

Выполняется проверка на присутствие ключевого поля в новой записи, если условие не выполнено, то перед добавлением новой записи в ней заполняется ключевое поле.

Пример использования:

select count(\*) from PEOPLES;

COUNT

6800

insert into PEOPLES values (null,'Zvenov','Stepan','Mihailovich','8911000202','7777777');

select count(\*) from person;

COUNT

6801

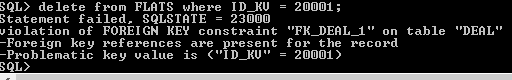


Запись создалась, ID\_PPL присвоен.

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

Контролируется удаление записей из таблицы FLATS, которое должно вызывать удаление всех сделок с данной квартирой.

Попробуем удалить квартиру до добавления триггера к таблице:



Триггер:

set term ^ ;

create or alter trigger tr\_del\_deal for FLATS

active before delete position 0

as begin

delete from DEAL where ID\_DEALFLAT = old.ID\_KV;

end ^

set term ; ^

Без триггера эту операцию невозможно совершить, так как она нарушает контроль ссылочной целостности.



В результате удаляется и квартира и связанные с ней сделки.

# 4. Индивидуальное задание

4.1 При закрытии сделки в случае превышения объема сделок по району продаваемой квартиры некоторой величины, повышать цену аналогичных квартир на некоторую величину.

Триггер повышает цену квартир в данном районе, аналогичных квартире в сделке (аналогичных по площади и кол-ву комнат), если сумма сделок за последние 30 дней превышает величину в 30млн по данному району.

Триггер:

set term ^ ;

create or alter trigger tr\_cash\_up for DEAL

active after insert position 0

as

declare variable SUM\_DEALS bigint;

declare variable DISTRICTNEWDEAL int;

declare variable TMPROOMS int;

declare variable TMPS int;

begin

for select DISTRICT from FLATS,DEAL where ID\_KV = ID\_DEALFLAT and ID\_DEAL = new.ID\_DEAL

into :DISTRICTNEWDEAL

do

begin

for select sum(DEALCASH),ROOMS,S from DEAL,FLATS where ID\_KV = ID\_DEALFLAT and DISTRICT = :DISTRICTNEWDEAL and DEALDATE between new.DEALDATE-30 and new.DEALDATE

group by ROOMS,S

into :SUM\_DEALS,:TMPROOMS,:TMPS

do

if (:SUM\_DEALS>30000000) then update FLATS set CASH = CASH\*1.05 WHERE DISTRICT = :DISTRICTNEWDEAL and ROOMS = :TMPROOMS and S between :TMPS\*0.9 and :TMPS\*1.1;

end

end ^

set term ; ^

Проверка:

Введем 6 новых квартир в одном районе. Причем квартиры с ID 20001,20002,20003,20005 – аналогичны друг другу.

insert into FLATS values (20001,3,100,'yes',9,2100,'no','yes',4200000,5,1,4000,2);

insert into FLATS values (20002,3,95,'yes',8,2100,'no','yes',3200000,5,1,4001,2);

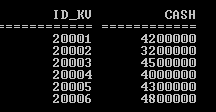
insert into FLATS values (20003,3,105,'yes',7,2100,'no','yes',4500000,5,1,4002,1);

insert into FLATS values (20004,2,89,'yes',6,2100,'no','yes',4000000,5,1,4003,2);

insert into FLATS values (20005,3,101,'yes',6,2500,'no','yes',4300000,5,1,4004,2);

insert into FLATS values (20006,4,109,'yes',6,2300,'no','yes',4800000,5,1,4005,2);

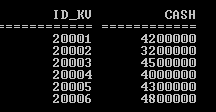
select ID\_KV,CASH from FLATS where ID\_KV >=20001 and ID\_KV<=20006;



Создадим новые сделки с данными квартирами, даты сделок попадают в период в один месяц.

insert into DEAL values (20200,301,201,4100000,'2007-04-04',401,202,20001);

select ID\_KV,CASH from FLATS where ID\_KV >=20001 and ID\_KV<=20006;

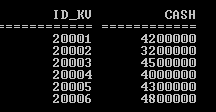


После добавления первой сделки цены аналогичных квартир остались прежними, т.к. не была превышена сумма (30млн) сделок за последний месяц относительно даты новой сделки.

insert into DEAL values (20201,301,201,3100000,'2007-04-03',401,202,20002);

select ID\_KV,CASH from FLATS where ID\_KV >=20001 and ID\_KV<=20006;

Аналогично.

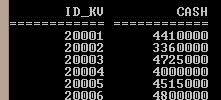


Добавим сделку на 40млн, чтобы точно превысить сумму сделок за месяц.

После добавления такой сделки цены на квартиры с ID 20001,20002,20003,20005 должны подняться. Цены на квартиры с ID 20004 и 20006 – должны остаться прежними, т.к. они не аналогичны (по кол-ву комнат и площади) квартире в сделке на 40млн.

insert into DEAL values (20202,301,201,41000000,'2007-04-04',401,202,20003);

select ID\_KV,CASH from FLATS where ID\_KV >=20001 and ID\_KV<=20006;



Триггер сработал верно.

4.2 При создании сделки, если сумма сделок за некоторый период в данном районе была меньше некоторой величины давать скидку на квартиру в заданном размере.

## Процедура считает сумму сделок по данному району за последний месяц. Входными параметрами являются дата сделки и номер района, в котором находится квартира в текущей сделке. Выходной параметр – сумма сделок по району за последний месяц с момента сделки.

Процедура DISCFLAT:

SET TERM ^ ;

CREATE OR ALTER procedure DISCFLAT (NEWDEALDATE date,NEWDISTRICTFLAT int)

returns (SUMDEAL bigint)

as

begin

for select sum(DEALCASH)

from DEAL,FLATS

where ID\_KV = ID\_DEALFLAT and DISTRICT = :NEWDISTRICTFLAT and DEALDATE between :NEWDEALDATE-30 and :NEWDEALDATE

into :SUMDEAL

do

suspend;

end

^

SET TERM ; ^

Триггер вызывает вышерасположенную процедуру. Если сумма сделок по району квартиры из новой сделки меньше 30млн, то на эту квартиру при сделке предоставляется скидка 5%.

Триггер TR\_CASH\_DOWN:

set term ^ ;

create or alter trigger tr\_cash\_down for DEAL

active before insert position 0

as

declare variable SUMDEAL bigint;

declare variable DISTRICTNEWDEAL int;

begin

select first 1 DISTRICT from FLATS,DEAL where ID\_KV = new.ID\_DEALFLAT

into DISTRICTNEWDEAL;

execute procedure DISCFLAT(new.DEALDATE,DISTRICTNEWDEAL)

RETURNING\_VALUES (SUMDEAL);

if (SUMDEAL<30000000) then

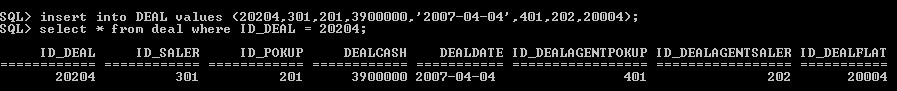
new.DEALCASH = new.DEALCASH\*0.95;

end

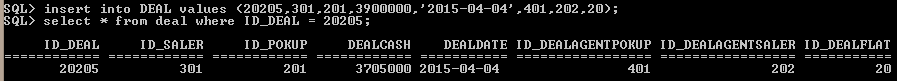
^

set term ; ^

Проверка:



Скидка не была предоставлена, т.к. сумма сделок по району данной квартиры превышает значение 30млн.



Скидка была предоставлена.

# 5. Вывод

В ходе данной работы были изучены триггеры. Триггер связан с определенной таблицей и вызывается автоматически до или после определенного действия по изменению записей в таблице. Триггеры – основной способ контроля целостности данных при удалении записи в одной из связанных таблиц. Триггер можно использовать для автоматического заполнения полей таблицы при добавлении записей или контроля проводимых операций. При этом триггеры усложняют работу БД и негативно влияют на производительность.