Санкт-Петербургский политехнический университет

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

ОТЧЕТ

По лабораторным работам

по дисциплине "Базы данных"

Выполнил: студент гр. 43501/3 Бабарицкий П.А.

Проверил: Мяснов А.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/подпись преподавателя, дата/

Санкт-Петербург

2016

1. **SQL-программирование: Триггеры, вызовы процедур**
   1. **Цель работы**

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

* 1. **Программа работы**
     + 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
     + Создать триггер в соответствии с индивидуальным заданием, полученным у преподавателя
     + Создать триггер в соответствии с индивидуальным заданием, вызывающий хранимую процедуру
     + Выложить скрипт с созданными сущностями в svn
  2. **Выполнение работы**

Создание триггеров:

Триггер для автоматического заполнения ключевого поля

CREATE SEQUENCE my\_generator

CREATE OR ALTER trigger tr1\_in for brends

active before insert position 0

AS

begin

if (new.id\_brand is null) then

NEW.id\_brand = GEN\_ID(my\_generator, 1);

end;

CREATE PROCEDURE get\_my\_generator\_val

RETURNS (new\_value\_ INTEGER)

AS

BEGIN

new\_value\_ = GEN\_ID(my\_generator, 1);

END

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

CREATE TRIGGER TR\_3 FOR BRENDS

BEFORE UPDATE OR DELETE

AS

begin

delete from modeles t

where t.id\_brand = old.id\_brand;

end

Триггер, для автоматического заполнения поля стоимости заказа

CREATE OR ALTER trigger deals\_aiu0 for deals

active before insert position 0

AS

declare variable am integer;

begin

am = (select sum((complete.price\_compl+COALESCE(tab.sum\_opt,0))\*(100-COALESCE(discounts.discount\_size,0))/100)

from cars left join complete

on cars.id\_compl = complete.id\_compl

left join (select options\_cars.id\_car, sum(options.price\_option) sum\_opt

from options\_cars join options

on options\_cars.id\_options = options.id\_options

where options\_cars.id\_car = new.id\_car

group by options\_cars.id\_car) tab

on cars.id\_car = tab.id\_car

left join discounts on discounts.id\_discount = new.id\_discount

where cars.id\_car = new.id\_car);

new.amount = am;

end;

**Триггер, согласно индивидуальному заданию:**

При превышении суммы заказа заданной величины добавлять к автомобилю заданную опцию.

CREATE OR ALTER trigger tr\_sum\_deal for deals

active before insert position 0

AS

declare disc integer;

declare sum\_deal integer;

declare optcar integer;

begin

if (new.id\_discount is not null) then

disc = (select COALESCE(discounts.discount\_size,0)

from discounts

where discounts.id\_discount = new.id\_discount);

else

disc = 0;

sum\_deal = (select sum(complete.price\_compl+COALESCE(sum\_opt.s,0))

from cars left join complete ON cars.id\_compl = complete.id\_compl

left join (select options\_cars.id\_car, sum(options.price\_option) as s

from options\_cars

join options on options\_cars.id\_options = options.id\_options

group by options\_cars.id\_car) as sum\_opt

on cars.id\_car = sum\_opt.id\_car

where cars.id\_car = new.id\_car

group by cars.id\_car);

sum\_deal = sum\_deal\*(100-disc)/100;

if (sum\_deal > 2400000) then

begin

optcar = (select 1 from options\_cars where options\_cars.id\_car = new.id\_car and options\_cars.id\_options = 3);

if (optcar is null) then

insert into options\_cars values (3, new.id\_car);

end

end;

При создании обращения по гарантии уменьшать сумму сделок по таким же (как и комплектация автомобиля, с которым обращаются по гарантии) комплектациям за последний месяц на 5%.

CREATE OR ALTER procedure PROCEDURE\_UPDATE\_AMOUNT\_DEALS (

NEW\_AM integer not null,

ID\_DEAL\_ integer not null)

as

begin

update deals

set deals.amount = :new\_am \* 0.95

where deals.id\_deal = :id\_deal\_;

end;

CREATE OR ALTER trigger request\_biu0 for request

active before insert position 0

AS

declare variable id\_compl\_ integer;

declare variable id\_deal\_ integer;

declare variable deal\_am\_ integer;

declare variable deal\_mar\_ integer;

declare variable sum\_am integer;

begin

id\_compl\_ = (select cars.id\_compl

from deals left join cars ON deals.id\_car = Cars.id\_car

where deals.id\_deal = new.id\_deal);

for select deals.id\_deal,

sum(deals.amount) as am,

sum(((complete.price\_compl+COALESCE(sum\_opt.s,0))\*(100-discounts.discount\_size)/100)) as marg

from deals left join cars ON deals.id\_car = Cars.id\_car

left join complete ON cars.id\_compl = complete.id\_compl

left join discounts ON deals.id\_discount = discounts.id\_discount

left join (select options\_cars.id\_car, sum(options.price\_option) as s

from options\_cars

join options on options\_cars.id\_options = options.id\_options

group by options\_cars.id\_car) as sum\_opt

on cars.id\_car = sum\_opt.id\_car

WHERE complete.id\_compl = :id\_compl\_

and deals.date\_deal >= DATEADD(day, -30, new.date\_requ)

group by deals.id\_deal

into :id\_deal\_, :deal\_am\_, :deal\_mar\_

as cursor cur\_deal\_am

do

begin

if (:deal\_am\_ is null or :deal\_am\_ = 0) then

sum\_am = :deal\_mar\_;

else

sum\_am = :deal\_am\_;

execute procedure procedure\_update\_amount\_deals(:sum\_am, :id\_deal\_);

end

end;

* 1. **Выводы**

В работе были исследованы триггеры. Триггеры предназначены для выполнения простых операций по поддержанию целостности базы. Например, они могут генерировать индексы для таблицы, что является самым безопасным способом получения индекса. Также они позволяют каскадно удалять или изменять поля, зависящие от других таблиц.

Триггеры активизируются автоматически в результате определенных действий пользователя.