**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №5**

**«SQL-программирование: хранимые процедуры»**

**Информационное обеспечение систем управления**

Студент гр. 43501/1 Матлаш И.Г.

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.

# 2. Программа работы

1. Изучить возможности языка PSQL

2. Создать две хранимые процедуры в соответствии с индивидуальным заданием, полученным у преподавателя.

# Реализовать хранимые процедуры:

# 2.1.Для платных объявлений, у которых истек срок публикации сменить статус.

# 2.2.Удалить объявления в определенном статусе и опубликованные ранее заданной даты.

# 3. Индивидуальное задание

## 3.1 Для платных объявлений, у которых истек срок публикации сменить статус.

Была создана дополнительная таблица is\_paid , содержащая два вариант «платное» и «бесплатное», а также добавлен внешний ключ is\_paid указывающий на эту таблицу в таблицу car(объявление).

create table is\_paid(

id int primary KEY,

name varchar(20) not null unique);

alter table car add is\_paid int references is\_paid(id);

Процедура CHECK\_PAID:

SET TERM ^ ;

CREATE OR ALTER procedure CHECK\_PAID as

declare variable CAR\_ID integer;

declare variable DD integer;

declare variable PAID decimal(10,2);

declare variable PAID\_END decimal(10,2);

begin

for select id,payed,payed\_end from car where (payed\_end < current\_date or payed\_end is null) and payed > 10 into :car\_id,:paid,:paid\_end do

begin

execute procedure math\_payed (:paid,10) RETURNING\_VALUES (:dd,:paid);

update car set payed = :paid where id=:car\_id;

if (:PAID\_END is null) then update car set payed\_end = DATEADD(:dd MONTH to CURRENT\_DATE) WHERE id=:car\_id;

else update car set payed\_end = DATEADD( MONTH, :dd, car.payed\_end) WHERE id=:car\_id;

end

for select id from car where payed\_end < current\_date or payed\_end is null into :car\_id do

begin

UPDATE car SET is\_paid=2 WHERE id=:car\_id;

UPDATE car SET payed\_end=null WHERE id=:car\_id;

end

for select id from car where payed\_end >= current\_date and is\_paid = 2 into :car\_id do

begin

UPDATE car SET is\_paid=1 WHERE id=:car\_id;

end

end

^

SET TERM ; ^

Вызываемая процедура MATH\_PAYED высчитывает и возвращает новое значение счёта в зависимости от текущей стоимости одного месяца (параметр - COST), а также возвращает количество месяцев на которое следует продлить объявление (параметр – DD).

Процедура MATH\_PAYED:

SET TERM ^ ;

CREATE OR ALTER procedure MATH\_PAYED (

PAYED decimal(10,2),

COST decimal(10,2))

returns (

DD integer,

NEW\_PAYED decimal(10,2))

as

declare variable DIV decimal(10,2);

begin

div = :payed/:cost;

dd = FLOOR(div);

new\_payed = :payed-(dd\*:cost);

end

^

SET TERM ; ^

Вызов ХП CHECK\_PAID:

select id,is\_paid,payed,payed\_end from car where id=47;

ID IS\_PAID PAYED PAYED\_END

47 1 0 01.01.2015

execute procedure check\_paid;

select id,is\_paid,payed,payed\_end from car where id=47;

ID IS\_PAID PAYED PAYED\_END

47 2 0

Текущей датой при проведении теста было 10.02.2015 соответственно объявлению был присвоен статус «бесплатное», т.к. срок публикации истёк, а на счету объявления не достаточно денег для продления.

Проведём ещё один тест:

select id,is\_paid,payed,payed\_end from car where id=46;

ID IS\_PAID PAYED PAYED\_END

46 2 30

execute procedure check\_paid;

select id,is\_paid,payed,payed\_end from car where id=46;

ID IS\_PAID PAYED PAYED\_END

46 1 0 10.05.2015

Текущей датой при проведении теста было 10.02.2015 соответственно объявление было продлено на три месяца при текущей стоимости одного, равной 10, а также получило статус «платное».

## 3.2 Удалить объявления в определенном статусе и опубликованные ранее заданной даты.

SET TERM ^ ;

CREATE OR ALTER procedure DELETE\_CAR (p int,cr date) as

declare variable car\_id int;

begin

for select id from car where create\_date<:cr and is\_paid=:p into :car\_id do

begin

delete from car where id=:car\_id;

end

end

^

SET TERM ; ^

Проверка:

select id,is\_paid,create\_date from car where id=48;

# ID IS\_PAID CREATE\_DATE

# 48 2 01.01.2000

execute procedure delete\_car(2,'01-01-2001');

1 record(s) was(were) deleted from CAR

Процедура выполнена успешно.

# 4. Вывод

Работа была посвящена изучению хранимых процедур. В результате были созданы три хранимые процедуры.

Хранимые процедуры используются для переноса части операций в саму СУБД для уменьшения интенсивности сетевого взаимодействия. На сервер выносятся операции, не связанные с пользовательских интерфейсом и реализующие общую для всех пользователей логику, в особенности выполняющие часто требуемые задачи. Также процедуры помимо стандартных возможностей SQL имеют операторы обработки ветвлений, циклов и т.д.