

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

## **ОТЧЕТ**

### **о лабораторной работе № 5**

Хранимые процедуры

Базы данных

**Выполнила студентка:** гр.43501/3 Емельянова А.В.

**Преподаватель:** Мяснов А.В.

Санкт-Петербург  
2016

## 1. Цель работы.

Ознакомление с возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур.

## 2. Программа работы.

- Изучить возможности языка PSQL
- Создать две хранимые процедуры в соответствии с индивидуальным заданием, полученным у преподавателя
- Выложить скрипт с созданными сущностями в svn
- Продемонстрировать результаты преподавателю

## 3. Выполнение работы.

Создадим две хранимые процедуры в соответствии с заданием преподавателя:

1. Пользователям, которые не проявляли активности в заданном промежутке времени закрыть доступ на комментарии для выбранного пользователя.
2. Десяти наиболее активным пользователям за выбранный промежуток времени добавить бесплатно выбранную услугу.

3.1. Пользователям, которые не проявляли активности в заданном промежутке времени закрыть доступ на комментарии для выбранного пользователя:

```
CREATE OR ALTER PROCEDURE INDIV_TASK1 (  
    startdate date not null,  
    enddate date not null,  
    d1 date not null,  
    d2 date not null,  
    countt integer)  
  
as  
  
declare variable userid_to_update integer;  
declare variable enaid integer;  
  
begin  
  
SELECT first 1 eb.ena_id FROM enable_blog eb  
WHERE eb.ena_to_comment=1 AND eb.ena_to_watch=0 AND eb.ena_to_write = 0  
  
into :enaid;  
  
FOR  
  
select blog.owner
```

```

from blog

where not exists (select post_owner from blog_post where
blog_post.date_create > :d1 AND blog_post.date_create < :d2 and blog.blog_id
= blog_post.blog_id)

group by blog.owner

into :userid_to_update

DO

begin

UPDATE privacy_blog set privacy_blog.ena_blog = :ena_id
WHERE privacy_blog.user_id =: userid_to_update;

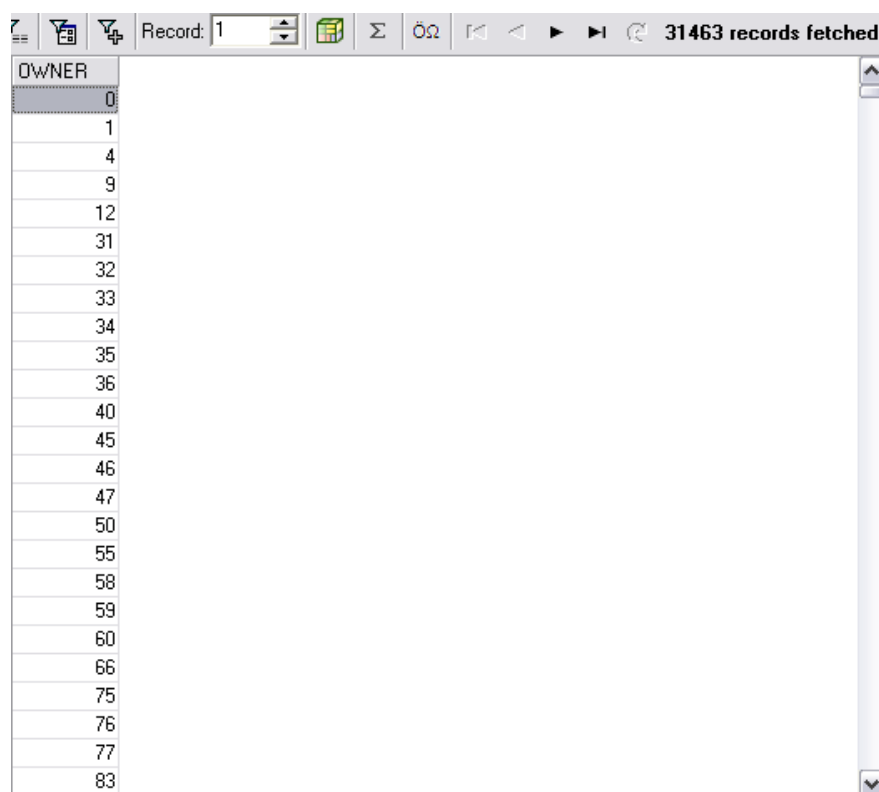
end

suspend;

end

```

Для проверки работоспособности процедуры сначала выполним запрос, чтобы вывести пользователей, которые не писали постов за определенный промежуток времени (с 29.05.2011 до 29.05.2016):



OWNER
0
1
4
9
12
31
32
33
34
35
36
40
45
46
47
50
55
58
59
60
66
75
76
77
83

Для вышеуказанных пользователей после выполнения процедуры должно будет установиться значение ENA\_BLOG = 3 в таблице PRIVACY\_BLOG.

Результат:

PRIV_ID	ENA_BLOG	USER_ID	BLOG_ID
0	3	98 959	0
1	3	9 878	0
2	6	6 943	0
3	5	87 095	0
4	5	92 688	0
5	7	23 674	0
6	8	49 950	0
7	8	8 775	0
8	3	53 155	0
9	8	90 156	0
10	6	86 968	0
11	8	13 041	0
12	3	50 154	0
13	8	37 847	0
14	4	42 536	0
15	1	74 461	0
16	3	29 883	0
17	1	25 093	0
18	3	80 881	0
19	1	42 335	0
20	7	89 193	0
21	7	64 966	0
22	3	88 226	0
23	3	22 686	0
24	2	42 906	0
25	1	83 479	0
26	3	17 690	0
27	4	84 955	0
28	7	79 506	0
29	2	25 153	0
30	1	54 604	0

Процедура была выполнена корректно.

3.2. Десяти наиболее активным пользователям за выбранный промежуток времени добавить бесплатно выбранную услугу:

```
CREATE OR ALTER PROCEDURE INDIV_TASK2 (  
    d1 date not null,  
    d2 date not null,  
    serv integer)  
as  
declare variable us_id integer;  
declare variable countt integer;  
begin  
    for  
        select first 10 count(bp.post_id), u.user_id
```

```

from userr u, blog b, blog_post bp
where b.owner=u.user_id and b.blog_id=bp.blog_id
and bp.date_create between :d1 and :d2
group by u.user_id
order by count(bp.post_id) desc
into :countt, :us_id
do
begin
update userr set paid_service =:serv
where userr.user_id =:us_id;
end
suspend;
end

```

Результат выполнения:

Сначала выполним SELECT из процедуры, чтобы получить 10 самых активных пользователей с количеством постов с 29.05.2014 по 29.05.2015:

COUNT	USER_ID
13	40 112
12	91 805
11	47 057
11	38 817
11	33 764
11	1 352
11	3 222
11	65 931
11	74 393
10	44 906

Затем выполним саму процедуру, задав в качестве входных параметров вышеуказанные даты и номер сервиса – 87.

Для проверки корректности выполнения процедуры напомним следующий запрос:

```
select * from userr where userr.paid_service=87
```

Результат выполнения:

USER_ID	USERNAME	PASSWORD	MAIL	COUNTRY	GENDER	BIRTH_DATE	STATUS	PAID_SERVICE
1 352	xRZh^QZsW^C0qx_ca_cUKNZfUL	i2F	rNPKn3Nmxj	Iceland	male	29.09.1990	online	87
3 222	x0dFTpfcHRdznk	`tly	hH	Czech Republic	male	07.09.2006	offline	87
33 764	FTM\g^Nxn[EKUmlhx	PVa	8	Djibouti	male	21.09.1975	offline	87
38 817	hVpBQv^_qNkucyHHQIIaS[faKk	bM10;	5qFL<z2K32h_X2j	Equatorial Guinea	male	10.09.2006	offline	87
40 112	hSlz^RXQuhZ]fCRossgoRp_^\	g:z	xUjb	Iraq	male	27.04.1979	online	87
44 906	LMC^ubMoxj\rmMaOBicvRUB\[ubxLkM	>FjA\TS	Ug3dK	India	male	08.04.1990	offline	87
47 057	G^PVmBHemMdynjRldJ	a^=aAt@^	<4@ValXgB7^Y2K	Taiwan	female	10.09.1964	online	87
65 931	lyLPfMIDkobiQy	m7:NQk	gDqG_=xMA[byf	Bosnia Herzegovina	female	24.03.2011	online	87
74 393	jvcgayrd	4Je05xz?	G:	Monaco	male	09.04.2001	offline	87
91 805	DidrAC[_rpQ]cuNLntv	ycE@	K45Y?	Libya	female	10.04.1962	offline	87

Как видно из результата, всем наиболее активным пользователям была присвоена заданная услуга.

#### **4. Выводы.**

Хранимые процедуры – это объекты базы данных, представляющие собой набор SQL-инструкций, который компилируется один раз и хранится на сервере. Они используются для того, чтобы предоставить пользователю возможность выполнять сложные SQL-запросы, используя только краткий вызов процедуры. За счет того, что ХП выполняются на стороне сервера, их использование повышает производительность.

Хранимые процедуры повышают безопасность, так как можно предоставлять пользователям доступ только к ним, а не непосредственно к таблицам базы данных. Удобно, что в хранимые процедуры можно передавать различные параметры, что позволяет осуществлять с их помощью операции, в которых требуется наложить на используемые данные различные условия. Наличие локальных переменных в ХП существенно упрощает составление большинства сложных SQL-запросов. Кроме того, в хранимых процедурах можно вызывать другие хранимые процедуры, как функции в любом высокоуровневом языке, что, скорее всего, будет удобно для человека, который их составляет.