

Санкт-Петербургский Политехнический Университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

## Отчет по лабораторной работе

Дисциплина: «Базы данных»

Тема: «SQL-программирование: Хранимые процедуры»

**Работу выполнила:**

Мартюшева Надежда

Группа: 43501/3

**Преподаватель:**

Мяснов Александр Владимирович

Санкт-Петербург  
2017

# 1 Цели работы

Познакомиться 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.
3. Выложить скрипт с созданными сущностями в svn.
4. Продемонстрировать результаты преподавателю.

## 3 Индивидуальное задание

1. Вывести 10 танцоров, которые имели наилучшую динамику результатов за заданный период.
2. Вывести 5 клубов, участники которых чаще всего становились судьями после этого не участвовали в соревнованиях, как танцоры.

## 4 Ход работы

Вывести 10 танцоров, которые имели наилучшую динамику результатов за заданный период.

```
1 set term ^;
2 create or alter procedure results_dinam (
3     num_year int,
4     date_from date
5 )
6 returns (
7     ASH_id int,
8     dancer_firstname varchar(20),
9     dancer_secondname varchar(20),
10    club_name varchar(20),
11    club_city varchar(20),
12    dinam float )
13 as
14 declare variable cur_dinam float;
15 declare variable last_dinam float not null;
16 declare variable date_st date;
17 declare variable date_end date;
18 declare variable cnt_year int;
19 declare variable cnt int;
20 declare variable cur_f cursor for(
21 select
22     d.ASH_id,
23     d.firstname,
24     d.secondname,
25     c.name,
26     c.city
27 from dancer d
28 join dancer_club dc on d.dancer_id = dc.dancer_id
29 join club c on dc.club_id = c.club_id
30 );
31
32 begin
33 open cur_f;
34 while (1 = 1) do
35 begin
36     fetch cur_f
37     into :ASH_id, :dancer_firstname, :dancer_secondname, :club_name, :club_city;
38     if (row_count = 0) then leave;
39
40     dinam = 0.;
41     last_dinam = 0.;
42     cnt_year = :num_year;
43     cnt = 0;
```

```

44 while (cnt_year>0) do
45 begin
46   date_st = cast(:date_from as date) - :cnt_year*365;
47   date_end = cast(:date_from as date) - :cnt_year*365 +365;
48
49   select
50     avg(cast(r.plase as float)/r.num_of_part) as dinam
51   from competition_result r
52   join competition_history h on r.comp_h_id = h.comp_h_id
53   where (r.leader_id = :ASH_id or r.partner_id = :ASH_id) and
54     (h.date_start between :date_st and :date_end)
55   into :cur_dinam;
56
57   if (:cur_dinam is null)
58   then cur_dinam = last_dinam;
59
60   if( :num_year = 1 or cnt = 1 and :last_dinam < 0.000001 )
61   then dinam = :dinam +:cur_dinam;
62   else
63   begin
64     if ( cnt > 0)
65     then dinam = :dinam + :last_dinam - :cur_dinam;
66   end
67
68   last_dinam = :cur_dinam;
69
70   cnt_year = :cnt_year - 1;
71   cnt = cnt + 1;
72 end
73
74 suspend;
75 end
76 close cur_f;
77 end^
78
79 set term ;^
80
81 select first 10 * from results_dinam(4, '25.12.2016') order by dinam desc;

```

Листинг 1: Хранимая процедура №1

Вывести 5 клубов, участники которых чаще всего становились судьями после этого не участвовали в соревнованиях, как танцоры.

```

1 set term ^;
2 create or alter procedure club_judge_dancer
3 returns (
4   club_name varchar(20),
5   club_city varchar(20),
6   num_not_dancer_judge integer,
7   num_dancer integer,
8   frequency float )
9 as
10 declare variable club_id int not null;
11 declare variable valid_judge_date date not null;
12 declare variable last_comp_date date not null;
13 declare variable cur_f cursor for(
14 select
15   dc.club_id,
16   count(*) as num_judge
17 from judge j
18   join dancer d on j.ASH_id = d.ASH_id
19   join dancer_club dc on d.dancer_id = dc.dancer_id
20 group by dc.club_id);
21 declare variable cur_s cursor for(
22 select
23   j.date_valid as valid_judge,
24   max(h.date_start) as last_comp
25 from judge j
26   join competition_result r
27   on j.ASH_id=r.leader_id or j.ASH_id=r.partner_id
28   join competition_history h on r.comp_h_id = h.comp_h_id
29   join dancer d on j.ASH_id = d.ASH_id
30   join dancer_club dc on d.dancer_id = dc.dancer_id
31 where dc.club_id = :club_id
32 group by j.ASH_id, valid_judge);
33

```

```

34| begin
35| open cur_f;
36| while (1 = 1) do
37| begin
38|   fetch cur_f
39|   into :club_id, :num_not_dancer_judge;
40|   if (row_count = 0) then leave;
41|
42|   open cur_s;
43|   while (1 = 1) do
44|   begin
45|     fetch cur_s
46|     into :valid_judge_date, :last_comp_date;
47|     if (row_count = 0) then leave;
48|
49|     if (valid_judge_date < last_comp_date)
50|     then num_not_dancer_judge = num_not_dancer_judge - 1;
51|   end
52|   close cur_s;
53|
54|   select c.name, c.city, count(*)
55|   from dancer_club dc
56|   join club c on dc.club_id = c.club_id
57|   where dc.club_id = :club_id
58|   group by c.name, c.city
59|   into :club_name, :club_city, :num_dancer;
60|
61|   if (num_dancer = 0) then
62|     frequency = NULL;
63|   else frequency = cast(:num_not_dancer_judge as float) / :num_dancer;
64|   suspend;
65| end
66| close cur_f;
67| end^
68|
69| set term ;^
70|
71| select first 5 * from club_judge_dancer order by frequency desc, num_not_dancer_judge desc;

```

Листинг 2: Хранимая процедура №2

## 5 Вывод

В ходе работы я ознакомилась с возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур.

Хранимая процедура является программой, хранящейся в области метаданных базы данных и выполняющейся на стороне сервера. К хранимой процедуре могут обращаться хранимые процедуры (в том числе и сама к себе), триггеры и клиентские программы.

Выполнение функций на сервере снижает нагрузку на канал связи, поскольку передается только окончательный результат, при его наличии. К тому же, на сервере процедуры хранятся уже в скомпилированном виде, поэтому на их выполнение тратится меньше времени.

Еще одним достоинством хранимых процедур является то, что приложения, работающие с одной и той же базой данных, могут использовать одну и ту же хранимую процедуру, тем самым уменьшив размер кода приложения и устранив дублирование кода. Также процедуры позволяют организовать интерфейс доступа к данным и в случае изменений на серверной стороне, позволяют избежать необходимости изменения и перекомпиляции клиентского приложения.