

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчёт о лабораторной работе №5

Дисциплина: Базы данных

Тема: Хранимые процедуры

Выполнил студент гр. 43501/1

А.А. Нагорнов

Руководитель

А.В. Мяснов

Санкт -Петербург

2016

1. Цели работы

Познакомить студентов 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. Язык PSQL

Procedural SQL (PSQL) — процедурное расширение языка SQL. Это подмножество языка используется для написания хранимых процедур, триггеров и PSQL блоков. Это расширение содержит все основные конструкции классических языков программирования. Кроме того, в него входят немного модифицированные DML операторы (SELECT, INSERT, UPDATE, DELETE и др.).

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. Если хранимая процедура вызывает саму себя, то такая хранимая процедура называется рекурсивной.

Хранимые процедуры имеют следующие преимущества:

1. Модульность. 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.
2. Упрощение поддержки приложений. При изменении хранимой процедуры, изменения отражаются сразу во всех приложениях, без необходимости их перекомпиляции.
3. Увеличение производительности. Поскольку хранимые процедуры выполняются на стороне сервера, а не клиента, то это уменьшает сетевой трафик, что повышает производительность

4. Ход работы

4.1. Для каждой страны вывести наиболее талантливую автора, группу, исполнителя с количеством написанных композиций.

Получение ID человека, исполнившего наибольшее кол-во треков для заданной страны

```
SET TERM ^ ;
create or alter procedure get_performer(countryName varchar(50))
returns(r_ID int)
as
declare variable currentCount int = 0; --кол. треков исполн. текущим чел.
declare variable currentID int = 0; --ID текущего человека
declare variable maxCount int = 0; --максимум исполненных треков
begin
r_ID = NULL;
for select people.id from people where people.country = :countryName into
:currentID --перебор всех исполнителей заданной страны
do
begin
select count(*) from performers where performers.id_ispolnit = :currentID into
:currentCount; --получение кол-ва исполнен. треков для текущего исполнителя
if(:currentCount > :maxcount) --если кол. треков исполнен. текущим
человеком больше максимума
then
begin
maxCount = currentCount; --максимум равен кол-ву исполнен. текущим
человеком треков
r_ID = currentID; --ID человека исполн. наибольшее количество
треков равно ID текущего человека
suspend;
end
end
end^
```

Результат:

```
SQL> select * from get_performer('Russia');

R_ID
=====
1

SQL> select * from get_performer('USSR');

R_ID
=====
2

SQL> select * from get_performer('Latvia');

R_ID
=====
9
```

Получение ID автора наибольшего кол-ва треков для заданной страны

```
create or alter procedure get_author(countryName varchar(50))
returns(r_ID int)
as
declare variable currentCount int = 0; --кол. треков написанных тек. человек.
declare variable currentID int = 0; --ID текущего человека
declare variable maxCount int = 0; --максимум написанных треков
begin
r_ID = NULL;
--перебор всех людей заданной страны
for select people.id from people where people.country = :countryName into
:currentID
do
begin
select count(*) from authors where authors.artist_id = :currentID into
:currentCount; --получение кол-ва написанных треков для текущего автора
if(:currentCount > :maxCount) --если количество треков написанных
текущим человеком больше максимума
then
begin
maxCount = currentCount; --максимум равен кол-ву треков написанных
текущим человеком
r_ID = currentID; --ID человека написавшего наибольшее количество
треков равно ID текущего человека
suspend;
end
end
end^
```

Результат:

```
SQL> select * from get_author('Russia');
```

```
      R_ID
=====
      1
```

```
SQL> select * from get_author('Lativa');
```

```
SQL> select * from get_author('USSR');
```

```
      R_ID
=====
      2
```

Получение ID группы исполнившей наибольшее кол-во треков для заданной страны

```
create or alter procedure get_group(countryName varchar(50))
returns(r_ID int)
as
declare variable currentCount int = 0; --количество треков исполненных
группой
declare variable currentID int = 0; --ID текущей группы
declare variable maxCount int = 0; --максимум исполненных треков
begin
r_ID = NULL;
for select groups.id from groups where groups.country = :countryName into
:currentID --перебор всех групп заданной страны
do
begin
select count(*) from performers where performers.id_group = :currentID into
:currentCount; --получение кол-ва исполненных треков для текущего группы
if(:currentCount > :maxcount) --если количество треков исполненных
текущей группой больше максимума
then
begin
maxCount = currentCount; --максимум равен кол-ву исполненных
текущей группой треков
r_ID = currentID; --ID группы исполнившей наибольшее
количество треков равно ID текущей группы
suspend;
end
end
end^
```

Результат:

```
--
SQL> INSERT INTO tracks VALUES (35, 'Book', 'Russian');
SQL> INSERT INTO tracks VALUES (36, 'about', 'Russian');
SQL> INSERT INTO tracks VALUES (37, 'love', 'Russian');
SQL> INSERT INTO tracks VALUES (38, 'Boo abou lov', 'Russian');
SQL>
SQL> INSERT INTO performers VALUES (33, 35, null, 3,2004);
SQL> INSERT INTO performers VALUES (34, 36, null, 3,2005);
SQL> INSERT INTO performers VALUES (35, 37, null, 4,2013);
SQL> INSERT INTO performers VALUES (36, 38, null, 3,2007);
SQL>
SQL> select * from get_group('Russia');

=====
R_ID
=====
3
```

Получение информации по странам об авторах, группах и исполнителях

```
create or alter procedure get_country_statistic
returns(r_RowName varchar(50), r_Name varchar(50), r_Count varchar(50))
as
declare variable currentCountry varchar(50) = NULL;
declare variable currentID int = NULL;
begin
for select * from country into :currentCountry --перебор всех стран
do
begin
currentID = (select * from get_performer(:currentCountry));
if(:currentID is not null) then
begin
r_RowName = 'Country:' || :currentCountry;
r_Name = 'Performer:' || (select people.name from people where people.id =
:currentID);
r_Count = 'Track count:' || (select count(*) from performers where
performers.id_ispolnit = :currentID);
suspend;
end
currentID = (select * from get_author(:currentCountry));
if(:currentID is not null) then
begin
r_RowName = 'Country:' || :currentCountry;
r_Name = 'Author:' || (select people.name from people where people.id =
:currentID);
r_Count = 'Track count:' || (select count(*) from authors where authors.artist_id =
:currentID);
suspend;
end
currentID = (select * from get_group(:currentCountry));
if(:currentID is not null) then
begin
r_RowName = 'Country:' || :currentCountry;
r_Name = 'Group:' || (select groups.name from groups where groups.id =
:currentID);
r_Count = 'Track count:' || (select count(*) from performers where
performers.id_group = :currentID);
suspend;
end
end
end
end^
```

Результат:

R_ROWNAME	R_NAME	R_COUNT
Country:Russia	Performer:Shnurov	Track count:4
Country:Russia	Author:Shnurov	Track count:5
Country:Russia	Group:Black Star Mafia	Track count:3
Country:Latvia	Performer:Laima Vajkuļe	Track count:2
Country:USSR	Performer:Tsoi	Track count:3
Country:USSR	Author:Tsoi	Track count:5

SQL> █

4.2. Для каждого человека вывести динамику активности по годам. Активность вычислять произвольно с учетом авторства и исполнения композиций.

Получение активности человека по годам

```
create or alter procedure get_people_statistic
returns(r_People varchar(50), r_Year int, r_Activity int)
as
    declare variable currentPeopleID int    = NULL;           --ID текущего человека
    declare variable currentYear int        = NULL;           --текущий год
    declare variable currentActivity int     = NULL;           --активность тек. человека
begin
    for select people.id from people into :currentPeopleID --перебор всех людей
    do
        begin
            --перебор всех годов в которых были исполнены или написаны треки
            for select *
            from (select distinct authors.yeard as dateYear from authors
            union
            select distinct performers.yeard as dateYear from performers) into :currentYear
            do
                begin
                    --получение кол-ва исполненных текущим человеком треков в текущем году
                    currentActivity = (select count(*) from performers where performers.id_ispolnit =
:currentPeopleID and performers.yeard = :currentYear);
                    --плюс кол-во написанных текущим человеком треков в текущем году
                    currentActivity = currentActivity + (select count(*) from authors where
authors.artist_id = :currentPeopleID and authors.yeard = :currentYear);
                    if(:currentActivity != 0) --если активность не равна нулю
                    then
                        begin
                            --получение статистики активности текущего человека в текущем году
                            r_People = (select people.name from people where people.id = :currentPeopleID);
                            r_Year = currentYear;
                            r_Activity = currentActivity;
                            suspend;
                        end
                    end
                end
            end
        end
    end^

SET TERM ; ^
```

Результат:

```
SQL> select * from get_people_statistic;
```

R_PEOPLE	R_YEAR	R_ACTIVITY
Shnurov	1993	2
Shnurov	2000	1
Shnurov	2002	2
Shnurov	2005	1
Shnurov	2011	1
Shnurov	2012	1
Shnurov	2016	1
Tsoi	1872	1
Tsoi	1979	1
Tsoi	1980	2
Tsoi	1993	2
Tsoi	2009	1
Tsoi	2016	1
Scriptonit	2012	1
Nikolaj Baskov	1978	1
Nikolaj Baskov	2001	1
Nikolaj Baskov	2009	1
Bjanka	1999	1
Bjanka	2011	1
Bjanka	2012	1
<hr/>		
R_PEOPLE	R_YEAR	R_ACTIVITY
Laima Vajkule	2006	1
Laima Vajkule	2008	1
Filip Kirkorov	2013	1
Filip Kirkorov	2014	1
Alla Pugachova	2001	1
Lolita	1997	1
Lolita	2008	1
Lolita	2009	1
Timati	1980	1
Timati	1998	1
Timati	2005	1
Timati	2007	1
Timati	2008	1
Timati	2010	1
Agutin	2004	1
Agutin	2005	1
Irina Allegrava	2007	1
Alisa	2013	1
Butusov	1991	1
Butusov	1994	1
<hr/>		
R_PEOPLE	R_YEAR	R_ACTIVITY
Butusov	1998	1
Butusov	2006	1
Butusov	2007	1
Butusov	2008	1
Visockij	1965	1
Visockij	1970	1
Visockij	1978	1
Visockij	2003	1
Visockij	2016	2

5. Выводы

В ходе выполнения данной лабораторной работы были получены навыки написания хранимых процедур. Сложные запросы или их последовательность удобнее один раз оформить в хранимую процедуру, а дальше просто обращаться к ней, указывая необходимые параметры. Это значительно сокращает код и делает работу с запросами более логичной. Следует отметить, что использование хранимых процедур усложняет поддержку и развитие проекта в целом за счет переноса части логики в СУБД, что усложняет архитектуру приложения. Языки, на которых пишут хранимые процедуры, зачастую уступают по возможностям языкам высокого уровня.