

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

Отчет по лабораторной работе №5
по дисциплине «Базы данных»
SQL-программирование: Хранимые процедуры

Выполнил

студент гр. 43501/3

Е.А. Никитин

Преподаватель

А.В. Мяснов

«__»_____2015г.

Санкт-Петербург

2015

Цели работы

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

Программа работы

1. Изучить возможности языка PSQL
2. Создать хранимые процедуры в соответствии с **индивидуальным заданием**, полученным у преподавателя
3. Выложить скрипт с созданными сущностями в svn
4. Продемонстрировать результаты преподавателю

Ход работы:

1) Хранимая процедура — объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере. Хранимые процедуры очень похожи на обыкновенные процедуры языков высокого уровня, у них могут быть входные и выходные параметры и локальные переменные, в них могут производиться числовые вычисления и операции над символьными данными, результаты которых могут присваиваться переменным и параметрам. В хранимых процедурах могут выполняться стандартные операции с базами данных (как DDL, так и DML). Кроме того, в хранимых процедурах возможны циклы и ветвления, то есть в них могут использоваться инструкции управления процессом исполнения.

Хранимые процедуры похожи на определяемые пользователем функции (UDF). Основное различие заключается в том, что пользовательские функции можно использовать как и любое другое выражение в SQL запросе, в то время как хранимые процедуры должны быть вызваны с помощью функции CALL или EXECUTE.

2) Была создана хранимая процедура по индивидуальному заданию.

```
create procedure sr_zap(first_date date, second_date date)
```

```
returns( percent int)
```

```
as
```

```
declare variable rout_cnt int;
```

```
declare variable all_p int;
```

```
begin
```

```

select count(*) from shedule

where shedule.r_date between :first_date and :second_date

into :rout_cnt;


select SUM(p)/:rout_cnt from

(select T1.name1 as name1, T1.all_places as all_places,
T2.name as name2, T2.tick_cnt as tick_cnt, T2.tick_cnt*100/T1.all_places as p
from

(select shedule.route_id as name1, count(place.place_id) as all_places
from place, shedule
where shedule.train_id=place.train_id
group by name1 order by all_places desc)T1,

(select shedule.route_id as name, count(ticket.t_num) as tick_cnt
from shedule, ticket, place where
shedule.train_id=place.train_id and place.place_id=ticket.place_id
/*and ticket.r_date between '03.12.2015' and '31.12.2015' */
and ticket.r_date between :first_date and :second_date
group by name order by tick_cnt desc)T2
where T1.name1 = T2.name and T1.all_places > T2.tick_cnt order by p desc)
into :percent;

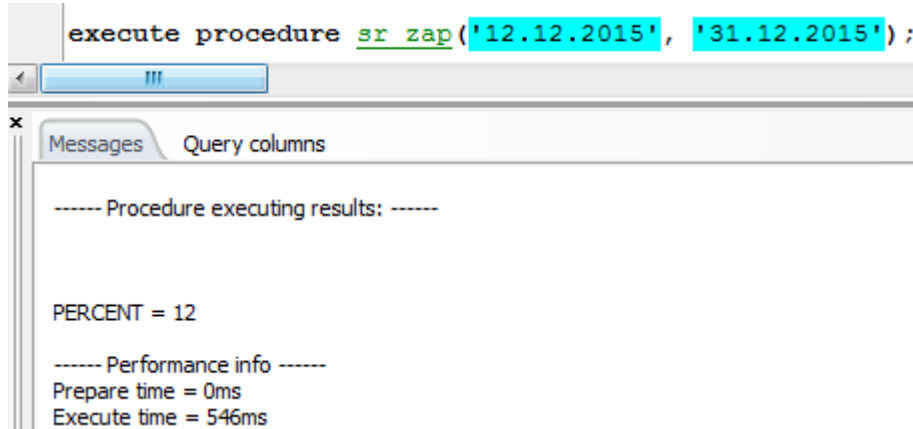
end;

```

В процедуре вычисляется, сколько маршрутов задействовано за данный период, вычисляется, сколько всего мест есть на данный маршрут и сколько было использовано – вычисляется процент заполненности для каждого маршрута. Далее проценты складываются и делятся на количество маршрутов данного периода.

Результат:

```
execute procedure sr_zap('12.12.2015', '31.12.2015');
```



----- Procedure executing results: -----

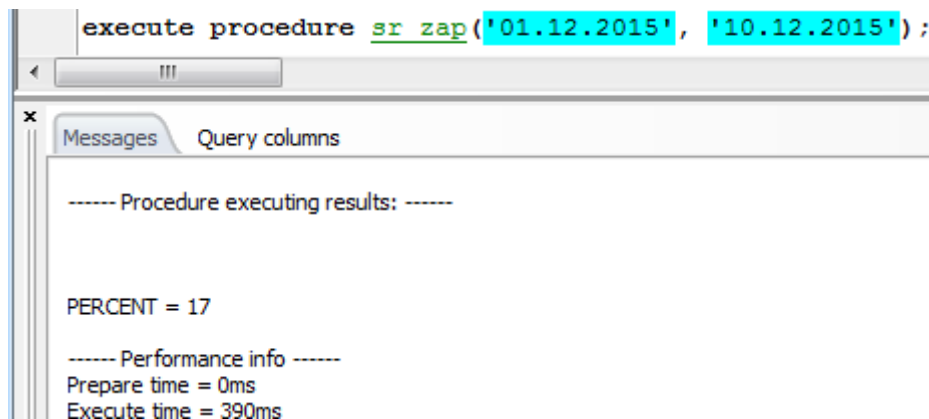
PERCENT = 12

----- Performance info -----
Prepare time = 0ms
Execute time = 546ms

Низкий процент вызван большим количеством созданных маршрутов по отношению к проданным билетам.

Результат с другими данными на входе:

```
execute procedure sr_zap('01.12.2015', '10.12.2015');
```



----- Procedure executing results: -----

PERCENT = 17

----- Performance info -----
Prepare time = 0ms
Execute time = 390ms

Вывод.

В ходе лабораторной работы были изучены хранимые процедуры в SQL.

В языке SQL хранимые процедуры позволяют сохранить какую-либо последовательность действий, чтобы в дальнейшем не писать их снова. Так же, ХП позволяют придать гибкость запросам с помощью параметров. ХП могут быть использованы в других запросах, ХП или триггерах – такое разбиение на «модули» позволит уменьшить количество кода для повышения читаемости.

Хранимые процедуры хранятся на сервере, поэтому выполняются быстрее (они уже откомпилированы). Также каждому пользователю доступен только его набор хранимых процедур – позволяет разграничить доступ. В любом случае легче выполнить уже готовую процедуру, чем отправлять серверу большой запрос.

Также упрощается разработка ПО, где программисты не знают или знают плохо SQL.