

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

## Базы данных

Отчет по лабораторной работе №5  
SQL-программирование: Хранимые процедуры

**Работу выполнила:**

Темнова А.С.

Группа: 43501/3

**Преподаватель:**

Мяснов А.В.

Санкт-Петербург  
2016

# 1 Цель 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 Ход выполнения работы

### 3.1 Для заданного поставщика и года вычислить динамику поставок поквартально в сравнении с аналогичными периодами за 3 предыдущих года.

```
1 CREATE PROCEDURE DELIVERY_DINAMIC (supplier_id int, choose_year int)
2 RETURNS (kvartal int, sum_year int, dinamic1yearago float, dinamic2yearago float,
3 ↪ dinamic3yearago float)
4 AS
5 declare VARIABLE sum1yearago integer;
6 declare VARIABLE sum2yearago integer;
7 declare VARIABLE sum3yearago integer;
8 begin
9     for
10
11     select
12     distinct trunc( (extract(month from delivery.date_delivery)-1)/3+1 ) as kvartal ,
13
14     (select
15     NULLIF(sum(d.number),0) as sum_delivery
16     from delivery d
17     where d.supplier_id = :supplier_id
18     and extract(year from d.date_delivery) = :choose_year
19     and trunc((extract(month from d.date_delivery)-1)/3+1) = trunc( (extract(month from
20 ↪ delivery.date_delivery)-1)/3+1 ) ) as sum_year,
21
22     (select
23     NULLIF(sum(d1.number),0) as sum_delivery
24     from delivery d1
25     where d1.supplier_id = :supplier_id
26     and extract(year from d1.date_delivery) = :choose_year - 1
27     and trunc((extract(month from d1.date_delivery)-1)/3+1) = trunc( (extract(month from
28 ↪ delivery.date_delivery)-1)/3+1 ) ) as sum_year_1,
29
30     (select
31     NULLIF(sum(d2.number),0) as sum_delivery
32     from delivery d2
33     where d2.supplier_id = :supplier_id
34     and extract(year from d2.date_delivery) = :choose_year - 2
35     and trunc((extract(month from d2.date_delivery)-1)/3+1) = trunc( (extract(month from
36 ↪ delivery.date_delivery)-1)/3+1 ) ) as sum_year_2,
37
38     (select
39     NULLIF(sum(d3.number),0) as sum_delivery
40     from delivery d3
41     where d3.supplier_id = :supplier_id
42     and extract(year from d3.date_delivery) = :choose_year - 3
43     and trunc((extract(month from d3.date_delivery)-1)/3+1) = trunc( (extract(month from
44 ↪ delivery.date_delivery)-1)/3+1 ) ) as sum_year_3
45
46     from delivery
47     order by kvartal
48
49     into :kvartal, :sum_year, :sum1yearago, :sum2yearago, :sum3yearago
```

```

46
47 do
48 begin
49 IF (:sum1yearago > 0) THEN
50 begin
51     dinamic1yearago = ( :sum_year - :sum1yearago ) / :sum1yearago ) * 100;
52 end
53 ELSE
54 begin
55     dinamic1yearago = 100;
56 END
57
58 IF (:sum2yearago > 0) THEN
59 begin
60     dinamic2yearago = ( :sum_year - :sum2yearago ) / :sum2yearago ) * 100;
61 end
62 ELSE
63 begin
64     dinamic2yearago = 100;
65 END
66
67 IF (:sum3yearago > 0) THEN
68 begin
69     dinamic3yearago = ( :sum_year - :sum3yearago ) / :sum3yearago ) * 100;
70 end
71 ELSE
72 begin
73     dinamic3yearago = 100;
74 END
75
76 SUSPEND;
77 END
78
79 END
80

```

Листинг 1: indiv 1

Пояснения: в процедуру передаются поставщик и год. Далее вытаскиваются из базы:

1. Кварталы (по которым есть данные, если в 3 квартале никогда в жизни никаких поставок не было, то и никаких данных по нему не будет)  $\text{trunc}((\text{extract}(\text{month from delivery.date\_delivery})-1)/3+1)$  – получить месяц от даты поставки, потом вычесть 1, поделить на три и прибавить 1, от этого всего взять целую часть(trunc).

2. Сумма поставок в этом квартале в заданном году заданного поставщика.

3. Три суммы поставок за предыдущие три года (там год минус 1/2/3).

Далее все результаты запроса попадают в переменные и считается процент прироста в заданном году по отношению к каждой сумме из предыдущих трех лет (проверяем не ноль ли, чтобы не делить на ноль). Ну и выводим всю эту радость (SUSPEND).

### 3.2 Для каждого сочетания тип/цвет вычислить периоды максимального спроса в течение одного года.

```

1 CREATE PROCEDURE PRODUCT_DEMAND (choose_year int)
2 RETURNS (wine_type char, wine_color char, month_of_max_demand char)
3 AS
4     declare VARIABLE max_quantity integer;
5     declare VARIABLE wine_type_id integer;
6     declare VARIABLE wine_color_id integer;
7     declare VARIABLE month integer;
8 begin
9
10     for
11
12     select
13         wine.color_id ,
14         max(color.name) as color_name ,
15         wine.type_id ,
16         max(type.name) as type_name ,
17
18     NULLIF((select
19         max(sum(p.number)) as month_sum
20         from purchase p

```

```

21         where p.storage_id in
22             (select storage.storage_id from storage where storage.delivery_id in
23              (select delivery.delivery_id from delivery where delivery.id_pr_wi in
24               (select producer_wine.wine_id from producer_wine where producer_wine.wine_id in
25                (select w.wine_id from wine w where w.type_id = wine.type_id and w.color_id = wine.
↪ color_id) ) ) )
26         and extract(year from p.date_buy) = :choose_year
27         group by extract(month from p.date_buy) ,0) as max_sum
28
29     from wine
30     left join color on color.color_id = wine.color_id
31     left join type on type.type_id = wine.type_id
32     group by wine.color_id , wine.type_id
33
34     into :wine_color_id , :wine_color , :wine_type_id , :wine_type , :max_quantity
35
36     do
37
38     begin
39
40         for
41
42         select
43             extract(month from p.date_buy)
44             from purchase p
45         where p.storage_id in
46             (select storage.storage_id from storage where storage.delivery_id in
47              (select delivery.delivery_id from delivery where delivery.id_pr_wi in
48               (select producer_wine.wine_id from producer_wine where producer_wine.wine_id in
49                (select wine.wine_id from wine where wine.type_id = :wine_type_id and wine.color_id
↪ = :wine_type_id) ) ) )
50         and extract(year from p.date_buy) = :choose_year
51         group by extract(month from p.date_buy)
52         having NULLIF(sum(p.number) ,0) = :max_quantity
53
54         into :month
55
56         do
57
58         begin
59             month_of_max_demand = month_of_max_manth '|| ' || month;
60         END
61
62         END
63     SUSPEND;
64
65     END
66

```

Листинг 2: indiv 2

Сначала выполняем запрос, который находит максимальные продажи в указанном году по каждому типу вина и цвету. Потом мы по каждому результату ищем (второй запрос в цикле) месяцы указанного года, в которых продажи были равны максимальной по данному типу и цвету и собираем их в строку через пробел. Затем выводим все это.

## 4 Выводы

В ходе данной работы я познакомилась с хранимыми процедурами (ХП).

Плюсы использования ХП:

- Скорость (При обработке даже небольших объёмов данных во внешнем приложении мы тратим дополнительное время на передачу по сети и преобразование данных в нужный нам формат. ХП компилируется только один раз – при выполнении в первый раз. Последующие выполнения проходят быстрее, поскольку фаза компиляции пропускается. Это помогает уменьшить нагрузку на сеть, так как код хранимой процедуры загружается только один раз)
- Соккрытие структуры данных
- Гибкое управление правами доступа (Пользователь имеет доступ к данным, но только в пределах хранимой процедуры. Любой другой доступ запрещается. Таким образом, уменьшается вероятность нарушения целостности данных из-за ошибки клиентского приложения)

- Меньшая вероятность SQL injection
- Повторное использование SQL
- Простая отладка SQL

Минусы использования ХП:

- Отсутствие контроля версии самой ХП (Когда изменяется версия ХП, она изменяется без сохранения истории действий для предыдущих версий на стороне сервера. Это может создать некоторые неудобства при возникновении необходимости откатить изменения)
- Отсутствие простого способа «синхронизировать» введение в силу изменений и заставить все субъекты использовать последнюю версию, в частности, когда разработка ведется группой людей и каждый из членов команды имеет свою собственную локальную базу данных для разработки и тестирования (Контроль версий может стать решением, но все еще требуется ручное вмешательство через обновление локальной копии ХП на локальном сервере БД)