Санкт-Петербургский Государственный Политехнический
Университет Петра Великого
Институт Компьютерных Наук и Технологий

Кафедра Компьютерных Систем и Программных Технологий

Отчёт по лабораторной работе
Дисциплина: Базы данных
Тема: SQL-программирование: Хранимые процедуры

Выполнил студент группы 43501/3 _____ Круминьш Д.В.
(подпись)

Преподаватель _____ Мяснов А.В.
(подпись)

Санкт-Петербург
2016

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

Ход работы

1 Сделать подарки 5 клиентам, которые принесли наибольшее количество денег за заданный период. Подарком считаем наиболее популярный товар за тот же период, проданный по нулевой цене.

В данной процедуре имеются два входных параметра, задающие даты начала и конца периода времени в котором производится выборка.

```
1  CREATE procedure PROC1(
2  startDate date,
3  finishDate date)
4  as
5  /*Некоторые переменные для хранения данных*/
6  declare variable topuser integer;
7  declare variable topproduct integer;
8  declare variable counter integer;
9  begin
10
11 select max(sell_id) from sells
12 into counter;
13
14 /*Поиск самого популярного продукта за указанный период*/
15 SELECT FIRST 1 PRODUCT.PRODUCT_ID
16 FROM PRODUCT JOIN sells_entre on PRODUCT.PRODUCT_ID=SELLS_ENTRE.
      ↪ PRODUCT_ID
17 JOIN SELLS ON SELLS.SELL_ID=SELLS_ENTRE.sell_id
18 where sell_date between :startDate and :finishDate
19 GROUP BY PRODUCT_ID
20 ORDER BY COUNT(PRODUCT.PRODUCT_ID) DESC, SUM(PRODUCT_PRICE) DESC
21 into topproduct;
22
23 /*Поиск пользователей с наибольшими покупками*/
24 for
25 select first 5 my_user.user_id
26 from my_user join sells on my_user.user_id=sells.user_id
27 join sells_entre on sells_entre.sell_id=sells.sell_id
28 where sell_date between :startDate and :finishDate
29 group by user_id
30 order by sum(product_price ) desc
31 into :topuser
32 do
```

```
33 begin
34
35 /*Внесение новых записей о покупках(подарках)*/
36 :counter=:counter+1;
37 insert into sells(sell_id, user_id, sell_date)
38 values(:counter, :topuser, current_date);
39 insert into sells_entre(sell_id,product_id,storage_id,product_price,
      ↪ quantity)
40 values(:counter, :topproduct,1,0,1);
41 end
42 suspend;
43 end
```

Листинг 1: procedure_1

2 Реализовать процедуру индексации цен, учитывающую уровень инфляции и популярность товаров.

Данная процедура имеет возможность инфляции и деинфляции, имеется возможность задать иной процент изменения цены для популярных товаров, требуется 5 входных параметров:

- inflation - булевая переменная для определения типа изменения цены
- countLimitation - граница количества продаж для определения популярного продукта
- priceLimitation - граница общих продаж продукта для определения популярного продукта
- percents1 - процент для популярных продуктов
- percents2 - процент для остальных продуктов

```
1  CREATE procedure PROC2(
2  inflation boolean,
3  countLimitation integer,
4  priceLimitation integer,
5  percents1 integer,
6  percents2 integer
7  )
8  as
9  declare variable currentProduct integer;
10 declare variable currentCount integer;
11 declare variable currentPrice float;
12 begin
13 for
14 /*Сортировка продуктов по количеству и сумме продаж*/
15 SELECT PRODUCT.PRODUCT_ID, COUNT(PRODUCT.PRODUCT_ID) as tCount
16 FROM PRODUCT left JOIN sells_entre on PRODUCT.PRODUCT_ID=SELLS_ENTRE.
      ↪ PRODUCT_ID
17 left join SELLS ON SELLS.SELL_ID=SELLS_ENTRE.sell_id
18 GROUP BY PRODUCT_ID
19 ORDER BY tCount DESC, sum(product_price)  DESC
20 into currentProduct, currentCount
21 do
22 begin
```

```
23 /*Определение текущей цены товара*/
24 select price from product where product_id=:currentproduct into :
      ↪ currentPrice;
25 /*Если инфляция*/
26 if (:inflation) then
27 begin
28 /*Увеличиваем цены согласно условиям*/
29 if (:currentcount>=countLimitation and :currentprice>=priceLimitation)
      ↪  then
30 update product set price=price+(price*:percents1/100) where product_id
      ↪ =:currentProduct;
31 else
32 update product set price=price+(price*:percents2/100) where product_id
      ↪ =:currentProduct;
33 end
34 else
35 begin
36 /*Дэ-инфляция*/
37 if (:currentcount>=countLimitation and :currentprice>=priceLimitation)
      ↪  then
38 update product set price=price-(price*:percents1/100) where product_id
      ↪ =:currentProduct;
39 else
40 update product set price=price-(price*:percents2/100) where product_id
      ↪ =:currentProduct;
41 end
42 end
43 suspend;
44 end
```

Листинг 2: procedure_2

Вывод

В ходе данной работы я познакомился с хранимыми процедурами(ХП), которые сильно напоминают функции в любом высокоуровневом ЯП. ХП позволяют сохранить часто используемые однотипные операции сложной выборки данных из базы. Особенно это полезно, если операции отличаются только константами, использующимися при наложении условий на данные.

Использование ХП позволяет повысить безопасность путем предоставления пользователю доступа только к ним, а не непосредственно к таблицам базы данных. Так-же в некоторой степени уменьшится количество запросов к серверу, то есть скоратится сетевой трафик.

Плюсы:

- Скорость (во внешнем приложении тратится дополнительное время на передачу по сети и преобразование в нужный формат)
- Безопасность (сокрытие структуры данных, ограничение прав пользователей, меньшая вероятность SQL-injection)

Недостатки:

- Отделенность от клиентского кода

- Не выполняется контроль целостности кода хранимой процедуры при изменении схемы данных