

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: SQL-программирование: Хранимые процедуры

Выполнил студент гр. 43501/1

(подпись) Г. А. Жемелев

Преподаватель

(подпись) А. В. Мяснов

“15” декабря 2016 г.

Санкт-Петербург

2016

Цель работы

Познакомиться с возможностями реализации сложной обработки данных на стороне сервера с помощью хранимых процедур.

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 систему контроля версий.
4. Продемонстрировать результаты преподавателю.

Ход работы

Индивидуальное задание, полученное у преподавателя, звучит следующим образом:

1. Вычислить максимально позднюю дату поставки заданного товара на основании текущих остатков на складах и скорости продажи.
2. Для заданного товара произвести поиск аналогов с аналогичными или лучшими характеристиками. В выводе списка аналогов выводить их средний рейтинг по отзывам.

Хранимая процедура, реализующая первый пункт задания имеет вид (также определен ряд исключений для контроля ошибок):

```
SET TERM ^;
```

```
CREATE EXCEPTION ex_TooNewArticle
```

```
'Нельзя рассчитать максимально позднюю дату поставки этого товара - сегодня он поступил впервые!'^
```

```
CREATE EXCEPTION ex_NoSells
```

```
'На данный момент не продано ни одной единицы этого товара!'^
```

```
CREATE EXCEPTION ex_NoArrivals
```

```
'Данный товар не поступал ни на один склад!'^
```

```
CREATE PROCEDURE dateToNextArrival (targetArticle TYPE OF COLUMN goods.Article)
```

```
RETURNS (dateTNA DATE) AS
```

```
    DECLARE VARIABLE sellRate FLOAT;
```

```
    DECLARE VARIABLE firstArrivalDate DATE;
```

```
    DECLARE VARIABLE totalSells INTEGER;
```

```
    DECLARE VARIABLE remains INTEGER;
```

```

BEGIN
    -- Найдем самую раннюю дату поставки целевого товара
    SELECT Arr_Date FROM arrivals WHERE article = :targetArticle
    ORDER BY Arr_Date ASC ROWS 1 INTO :firstArrivalDate;
    IF (firstArrivalDate IS NULL) THEN EXCEPTION ex_NoArrivals;
    -- Определим общее количество проданных единиц целевого товара
    SELECT SUM(od.Quantity) FROM orders_distribution od NATURAL JOIN orders o
    WHERE od.article = :targetArticle AND o.Status = 3 INTO :totalSells;
    -- Не делаем проверку на o.Order_Date >= :firstArrivalDate, т.к. предзаказы тоже
    -- идут в зачет
    IF (firstArrivalDate = 'TODAY') THEN EXCEPTION ex_TooNewArticle;
    IF (totalSells IS NULL OR totalSells = 0) THEN EXCEPTION ex_NoSells;
    -- Вычислим среднюю скорость продажи товара в шт./день
    sellRate = totalSells/CAST((date 'TODAY' - firstArrivalDate) AS FLOAT);
    -- Выясним, сколько единиц целевого товара осталось на складах
    SELECT SUM(Quantity) FROM availability
    WHERE article = :targetArticle INTO :remains;
    -- Теперь можем найти максимально позднюю дату поставки заданного товара
    dateTNA = date 'TODAY' + CAST(remains/sellRate AS INTEGER);
END^

SET TERM ;^

```

Проверим результат её выполнения для товара с артикулом 4 (рис. 1):

```
SQL> select * from arrivals where article = 4;
```

ID	ARR_DATE	ARTICLE	STO_ID	QUANTITY
3	2016-09-25	4	1	5
11	2016-11-03	4	2	3

```
SQL> select * from orders_distribution where article = 4;
```

ORDER_NO	ARTICLE	QUANTITY	FIXED_PRICE
23	4	2	13700.00
1234	4	3	8000.00

```
SQL> select * from orders where order_no in (23, 1234);
```

ORDER_NO	ORDER_DATE	PHONE	STATUS	FIXED_COST
1234	2016-11-05	89115671929	3	24000.00
23	2016-10-20	11712974702	3	76200.00

```
SQL> select * from availability where article = 4;
```

STO_ID	ARTICLE	QUANTITY
1	4	0
2	4	3

```
SQL> execute procedure dateToNextArrival(4);
```

```

    DATETNA
=====
2017-02-02

```

Рис. 1. Исходные данные и результат выполнения процедуры dateToNextArrival

Дата на момент исполнения процедуры – 15.12.2016. Значит, с момента первого поступления выбранного товара прошел 81 день. За это время было продано 5 единиц товара, и на складе осталось еще 3 единицы. Тогда количество дней до крайней даты поступления товара (чтобы он не кончился) рассчитывается как частное от деления числа оставшихся единиц товара на среднюю скорость продажи этого товара, выраженную в единицах в день, то есть:

$$N = \frac{3 \text{ ед.}}{5 \text{ ед./81 дн.}} = \frac{243}{5} \text{ дн.} = 48,6 \text{ дн.} \approx 49 \text{ дн.}$$

Через 49 дней от 15.12.2016 наступит 02.02.2017. Следовательно, результат работы процедуры верный.

Для выполнения второго пункта индивидуального задания была разработана хранимая процедура findAnalogues, выполняющая поиск аналогов заданного товара по заданной характеристике (аналоги имеют такое же или лучшее значение этой характеристики), а также вспомогательная процедура isSameOrBetter, используемая для определения того, какое значение характеристики считать лучшим в каждом конкретном случае. Соответствующий скрипт:

```
SET TERM ^;
CREATE PROCEDURE findAnalogues
(sourceArticle TYPE OF COLUMN goods.Article, compSpecID TYPE OF COLUMN
specifications.ID)
RETURNS
(resArticle TYPE OF COLUMN goods.Article, resName TYPE OF COLUMN goods.Name, avgRating
NUMERIC(3,2))
AS
    DECLARE VARIABLE sourceSpecVal TYPE OF COLUMN spec_arrangement.Spec_value;
    DECLARE VARIABLE curArticle TYPE OF COLUMN goods.Article;
    DECLARE VARIABLE curSpecID TYPE OF COLUMN specifications.ID;
    DECLARE VARIABLE curSpecVal TYPE OF COLUMN spec_arrangement.Spec_value;
    DECLARE VARIABLE compRes SMALLINT;
BEGIN -- Определяем значение характеристики исходного товара
    SELECT Spec_value FROM spec_arrangement WHERE article = :sourceArticle AND specID
= :compSpecID
    INTO :sourceSpecVal;
    -- Ищем товары с лучшими или такими же значениями данной характеристики
    FOR SELECT article, specID, Spec_value FROM spec_arrangement
    WHERE article <> :sourceArticle AND specID = :compSpecID
    INTO :curArticle, :curSpecID, :curSpecVal DO BEGIN
        EXECUTE PROCEDURE isSameOrBetter(sourceSpecVal, curSpecVal, curSpecID)
            RETURNING_VALUES :compRes;
        IF(compRes IN (1, 2)) THEN BEGIN
            resArticle = curArticle;
            resName = (SELECT Name FROM goods WHERE Article = :curArticle);
            avgRating = (SELECT AVG(CAST(Rating AS NUMERIC(3,2))) FROM reviews
                WHERE article = :curArticle);
        SUSPEND;
    END
```

```

END
END
END^

/* Данная процедура сравнивает два значения характеристик типа specType и возвращает:
    2, если они совпадают;
    1, если значение RHS лучше, чем LHS;
    0, если значения не совпадают, и нельзя сказать, что одно из них лучше другого.
*/
CREATE PROCEDURE isSameOrBetter(LHS TYPE OF COLUMN spec_arrangement.Spec_value,
                                RHS TYPE OF COLUMN spec_arrangement.Spec_value,
                                specType TYPE OF COLUMN specifications.ID)

RETURNS (res SMALLINT) AS
BEGIN
    IF(specType IN (1, 3, 4, 5)) THEN
        -- Целочисленные значения, для которых больше значит лучше
        IF(CAST(LHS AS INTEGER) < CAST(RHS AS INTEGER))
            THEN res = 1;
            ELSE IF(LHS = RHS) THEN res = 2; ELSE res = 0;
    ELSE IF(specType = 2) THEN
        -- Значения, на основе которых нельзя сказать, какой из двух товаров лучше
        IF(LHS = RHS) THEN res = 2; ELSE res = 0;
    ELSE res = 0;
END^

SET TERM ;^

```

Процедура findAnalogues возвращает множество кортежей, поэтому выполнять её следует с помощью оператора SELECT, например, таким образом:

```

SELECT
resArticle AS "Артикул", resName AS "Наименование", avgRating AS "Средний рейтинг"
FROM findAnalogues(10, 1) ORDER BY avgRating DESC;

```

В разрабатываемой БД наиболее полно характеристики определены для различных жестких дисков, поэтому испытать процедуру findAnalogues можно для различных артикулов из этой категории (рис. 2-4):

Артикул	Наименование	Средний рейтинг
21	MZ-7KE2T0BW	5.00
10 850	EVO 250Gb MZ-75E250BW	4.66
12	STDR2000200	<null>
11 850	EVO 250Gb MZ-N5E250BW	<null>

Рис. 2. Найденные аналоги товара с артикулом 9 (жесткий диск Samsung 850 EVO объемом 120 Гб) по характеристике «Объем»

Значение NULL в столбце рейтинга означает, что этот товар не был оценен.

Артикул	Наименование	Средний рейтинг
21	MZ-7KE2T0BW	5.00
12	STDR2000200	<null>
11	850 EVO 250Gb MZ-N5E250BW	<null>

Рис. 3. Найденные аналоги товара с артикулом 10 (жесткий диск Samsung 850 EVO объемом 250 Гб) по характеристике «Объем»

Артикул	Наименование	Средний рейтинг
21	MZ-7KE2T0BW	5.00
9	850 EVO 120Gb MZ-75E120BW	3.50
11	850 EVO 250Gb MZ-N5E250BW	<null>

Рис. 4. Найденные аналоги товара с артикулом 10 (жесткий диск Samsung 850 EVO объемом 250 Гб, IOPS(W) = 88000) по характеристике «IOPS(W)»

Артикул	Наименование	Средний рейтинг
21	MZ-7KE2T0BW	5.00

Рис. 5. Найденный аналог товара с артикулом 11 (жесткий диск Samsung 850 EVO объемом 250 Гб, IOPS(W) = 89000) по характеристике «IOPS(W)»

Товар с артикулом 21, соответствующий SSD Samsung MZ-7KE2T0BW объемом 2048 Гб с показателем IOPS(W) в 90000 оп/сек, обладает наилучшими показателями и не имеет аналогов в нашем магазине. Проверим это:

```
SQL> select resArticle AS "Артикул", resName AS "Наименование", avgRating AS
"Средний рейтинг" from findAnalogues(21, 1) order by avgrating DESC;
SQL> select resArticle AS "Артикул", resName AS "Наименование", avgRating AS
"Средний рейтинг" from findAnalogues(21, 4) order by avgrating DESC;
SQL>
```

Рис. 6. Поиск аналогов для товара с артикулом 21

Как видим, аналогов действительно не было найдено, что и ожидалось. Заметим, что при сравнении по характеристике, которой выбранный товар не обладает, результатом поиска будет пустое множество – ошибкой это не считается.

Выводы

Хранимые процедуры – это программы, хранящиеся в базе данных и выполняющиеся на стороне сервера. Вызов хранимой процедуры может производиться как со стороны клиента, так и из других процедур и триггеров. Хранимые процедуры в PSQL могут быть рекурсивными.

Хранимые процедуры позволяют выполнять сложную обработку данных в БД, при этом за счет того, что они хранятся и исполняются на стороне сервера, можно упростить клиентские приложения или даже сделать их «тонкими». При этом сетевой трафик существенно уменьшается: от клиента к серверу передаются вызовы процедур, а обратно возвращаются результаты их выполнения. За счет централизованного администрирования прикладных функций, которое достигается при использовании хранимых процедур, можно обеспечить дополнительный контроль целостности данных и кэширование.

Язык написания хранимых процедур специфичен для каждой СУБД (в Firebird это PSQL), что может стать существенной преградой при переходе на другую СУБД. Кроме того, в большинстве случаев для хранимых процедур не предусмотрено средств отладки и тестирования, что усложняет их разработку.