

SQL Dinâmico no Oracle – Parte 3

Eduardo Corrêa Gonçalves

Instituto Brasileiro de Geografia e Estatística – IBGE

E-mail: eduardo.correa@ibge.gov.br

1. Introdução

Este artigo apresenta um roteiro para implementar consultas dinâmicas que retornam múltiplas linhas no Oracle.

2. Consultas Dinâmicas que Retornam Múltiplas Linhas

Nos artigos anteriores, foi demonstrada a utilização do comando Execute Immediate na implementação de SQL's dinâmicos. Este comando pode ser utilizado para a montagem e execução de diversos tipos de instrução SQL, tais como: *create*, *drop*, *grant*, *insert*, *update*, *delete* e *select's* que retornem apenas uma linha. No entanto, o Execute Immediate não pode ser utilizado na implementação de consultas dinâmicas que resultam num conjunto composto por muitos registros. Neste caso é preciso utilizar os comandos **OPEN-FOR**, **FETCH** e **CLOSE**.

O código abaixo ilustra uma forma de implementar um SQL dinâmico que retorna múltiplas linhas. Seu funcionamento é explicado logo a seguir.

```
CREATE OR REPLACE
PROCEDURE P_SQL_DINAMICO(vTAB IN VARCHAR, vCAMPO IN VARCHAR) is

--ESTA PROCEDURE SELECIONA O CAMPO "vCAMPO" DA TABELA "vTAB"
--(recupera apenas as 100 primeiras linhas desta tabela)

type tpTeste is REF CURSOR; --PRIMEIRO É PRECISO ESPECIFICAR UM
                             --TIPO "REF CURSOR"
cCursorTeste tpTeste;       --DEPOIS DECLARE UMA VARIÁVEL DESSE TIPO

AUX VARCHAR(4000);
vSQL VARCHAR2(255);

begin
-- passo 1: monta o SELECT dinâmico num VARCHAR2
vSQL := 'SELECT ' || vCAMPO || ' FROM ' || vTAB || ' WHERE ROWNUM <= 100';

-- passo 2: abre e executa o cursor dinâmico usando a cursor variable
open cCursorTeste for vSQL;
loop
  fetch cCursorTeste into AUX;
  exit when cCursorTeste%notfound;

  DBMS_OUTPUT.PUT_LINE(trim(AUX));

end loop;

close cCursorTeste;

end P_SQL_DINAMICO;
```

Agora será apresentada a explicação sobre o funcionamento da procedure P_SQL_DINAMICO. Esta procedure recebe dois parâmetros como entrada: o nome de uma tabela qualquer (vTABELA) e o nome de um campo desta tabela (vCAMPO). No corpo da procedure, um SELECT dinâmico é montado para imprimir os 100 primeiros valores de vCAMPO (ou seja, os valores presentes nos 100 primeiros registros de vTABELA).

O primeiro passo para implementar a rotina, consiste na declaração de uma **variável do tipo cursor** (*cursor variable*), nas linhas 5 e 6. Uma variável do tipo cursor representa um recurso do PL/SQL que assemelha-se a um **ponteiro**. Este tipo de variável aponta para um endereço de memória que, por sua vez, contém alguma informação de interesse (ou seja, a *cursor variable* armazena um endereço e não um item). Quando o Oracle executa uma consulta que retorna muitas linhas, uma área de trabalho é aberta para o processamento dos dados obtidos pela consulta e uma *cursor variable* é um objeto capaz de acessar esta área de trabalho.

Dentro do corpo da procedure, o SQL dinâmico é montado numa variável VARCHAR2 em função dos parâmetros entrada (passo 1 - linha 11). A seguir, no passo 2 (linhas 12 a 19) os comandos OPEN-FOR, FETCH e CLOSE são utilizados para executar o SQL dinâmico. O comando OPEN-FOR (linha 13) é o mais importante na implementação da consulta dinâmica. Ele realiza as seguintes tarefas:

1. Associa a variável cursor com a consulta montada numa variável VARCHAR2 (no caso de nossa procedure exemplo, associa a *cursor variable* "cCursorTeste" com a string "vSQL").
2. Executa a consulta e identifica o conjunto de registros resultante.
3. Posiciona o cursor na primeira linha do conjunto.

Depois de executar a consulta dinâmica com o OPEN-FOR, basta fazer um loop com o comando FETCH, da mesma maneira que você faria com um cursor convencional (linhas 14 a 18). Ao final do processamento, o cursor deve ser fechado com o comando CLOSE (linha 19).