Prácticas de SQL DINÁMINO NATIVO Parte 2

PL/SQL 12c-18c avanzado

NOTA: Aunque siempre pongo las soluciones, os recomiendo que primero intentéis hacer el ejemplo por vosotros mismos y luego lo comparéis con el mío. ¡¡¡En muchas ocasiones, el mismo proceso se puede hacer de varias formas!!!!

SQL dinámico nativo en PL/SQL. Parte 2

1. Crear un paquete denominado que se llame GEST_EMPLE que en la cabecera tenga definido:
   - Un tipo NESTED TABLE denominado EMPLEADOS que contenga elementos de tipo EMPLOYEE
   - Una función denominada LISTA_EMPLEADOS, que reciba una cadena y devuelva una lista de empleados de tipo EMPLEADOS que hemos definido en el punto anterior
2. Ejemplo

```
CREATE OR REPLACE PACKAGE GEST_EMPLE
IS
   TYPE EMPLEADOS IS TABLE OF EMPLOYEES%ROWTYPE;
   FUNCTION LISTA_EMPLEADOS(CONDICION VARCHAR2) RETURN EMPLEADOS;
END;
/
```

3. En el body del paquete detallamos la función:
   - La cadena es la condición por la que debemos buscar los empleados
   - El resultado se debe devolver en un NESTE TABLE de tipo EMPLEADOS definido en la cabecera
   - Pista: podemos usar BULK COLLECT y la cláusula INTO para hacer el trabajo

Ejemplo

```
CREATE OR REPLACE PACKAGE BODY GEST_EMPLE
IS

   FUNCTION LISTA_EMPLEADOS(CONDICION VARCHAR2) RETURN EMPLEADOS
   IS
      RESULTADO EMPLEADOS;
      COMANDO VARCHAR2(100);
   BEGIN
```

```
    COMANDO:='SELECT * FROM EMPLOYEES WHERE '|| CONDICION;
    EXECUTE IMMEDIATE COMANDO BULK COLLECT INTO RESULTADO ;
    RETURN RESULTADO;
  END;
END;
/
```

4. Creamos un pequeño bloque PL/SQL que pruebe la función y que visualice los empleados recuperados.

```
DECLARE
  EMPLEADOS GEST_EMPLE.EMPLEADOS;
BEGIN
  EMPLEADOS:=GEST_EMPLE.LISTA_EMPLEADOS('DEPARTMENT_ID=30');
  FOR I IN 1..EMPLEADOS.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE(EMPLEADOS(I).FIRST_NAME);
  END LOOP;
END;
/


Den
Alexander
Shelli
Sigal
Guy
Karen
```

5. Vamos a crear ahora dentro del paquete un procedimiento denominado DATOS_EMPLEADOS. Este procedimiento visualizará datos de empleados. Tiene un argumento de tipo NUMBER.
    - Será el código de departamento de los departamentos que queremos visualizar
    - Debemos usar un REF CURSOR para recorrer los datos.
    - PISTA. Usaremos la cláusula USING

Ejemplo

```
CREATE OR REPLACE PACKAGE GEST_EMPLE
IS
  TYPE EMPLEADOS IS TABLE OF EMPLOYEES%ROWTYPE;
  FUNCTION LISTA_EMPLEADOS(CONDICION VARCHAR2) RETURN EMPLEADOS;
  PROCEDURE DATOS_EMPLEADOS(COLUMNAS VARCHAR2, CODIGO NUMBER);
END;
/


CREATE OR REPLACE PACKAGE BODY GEST_EMPLE
```

```
IS

   FUNCTION LISTA_EMPLEADOS(CONDICION VARCHAR2) RETURN EMPLEADOS
   IS
      RESULTADO EMPLEADOS;
      COMANDO VARCHAR2(100);
   BEGIN
     COMANDO:='SELECT * FROM EMPLOYEES WHERE '|| CONDICION;
     EXECUTE IMMEDIATE COMANDO BULK COLLECT INTO RESULTADO ;
     RETURN RESULTADO;
   END;

   PROCEDURE DATOS_EMPLEADOS(COLUMNAS VARCHAR2, CODIGO NUMBER)
   IS
      COMANDO VARCHAR2(100);
      TYPE TIPO_RESULTADOS IS REF CURSOR;
      RESULTADOS TIPO_RESULTADOS;

      TYPE TIPO_DATOS IS TABLE OF employees%rowtype;
      DATOS TIPO_DATOS;
   BEGIN
      COMANDO:='SELECT * FROM EMPLOYEES WHERE DEPARTMENT_ID=:CODIGO';
      OPEN RESULTADOS FOR COMANDO USING CODIGO;
      FETCH RESULTADOS BULK COLLECT INTO DATOS;
      FOR I IN 1..DATOS.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE(DATOS(I).first_name);

      END LOOP;
   END;
END;
/
```

6. Podemos probarlo con este bloque PL/SQL

```
SET SERVEROUTPUT ON
EXECUTE GEST_EMPLE.DATOS_EMPLEADOS('FIRST_NAME,SALARY',30);
```