

Prácticas de DBMS_SQL parte 1

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!!!!

DBMS_SQL parte 1

1. Añadir procedimiento al paquete GEST_EMPLE, denominado CREAM_INDICE, que nos permita crear un índice, que tenga 3 argumentos de tipo VARCHAR2:
 - Nombre de la tabla
 - Nombre del índice a crear
 - Nombre de la columna del índice
 - Utilizamos el paquete DBMS_LOB
 - Si hay algún error, generar un mensaje, indicándolo

Ejemplo

```

REATE 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);
    PROCEDURE CREAM_INDICE(TABLA VARCHAR2, INDICE VARCHAR2, COLUMNA
    VARCHAR2);
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;

PROCEDURE CREAR_INDICE(TABLA VARCHAR2, INDICE VARCHAR2, COLUMNA
VARCHAR2)
IS
    ID_CURSOR INTEGER;
    NUM_FILAS INTEGER;
BEGIN
    ID_CURSOR:=DBMS_SQL.OPEN_CURSOR;
    DBMS_SQL.PARSE(ID_CURSOR,'CREATE INDEX '||INDICE||' ON '||TABLA||'
('||COLUMNA||')',DBMS_SQL.NATIVE);
    NUM_FILAS:=DBMS_SQL.EXECUTE(ID_CURSOR);
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000,'ERROR EN EL COMANDO'||sqlerrm);
END;

END;
/

```

2. Para probarlo, por ejemplo creamos un índice sobre la tabla REGIONS. Comprobamos luego que se ha creado

```

SET SERVEROUTPUT ON
EXECUTE GEST_EMPLE.CREAR_INDICE('REGIONS','I_REGION','REGION_NAME');

```

3. Crear un procedimiento dentro del paquete llamado ACT_EMPLE que permita actualizar una columna de un empleado de la tabla EMPLOYEES. Le pasamos tres argumentos:

- Número de empleado
- Nombre de la columna a modificar
- Nuevo valor para la columna
- Se debe usar BIND_VARIABLE para el código del empleado y para el nuevo valor

Ejemplo

```
PROCEDURE ACT_EMPLE(CODIGO NUMBER, COLUMNA VARCHAR2, VALOR VARCHAR2)
IS
    ID_CURSOR INTEGER;
    NUM_FILAS INTEGER;
BEGIN
    ID_CURSOR:=DBMS_SQL.OPEN_CURSOR;
    DBMS_SQL.PARSE(ID_CURSOR,'UPDATE EMPLOYEES SET '||COLUMNA||'=:VALOR
WHERE EMPLOYEE_ID=:CODIGO',DBMS_SQL.NATIVE);
    DBMS_SQL.BIND_VARIABLE(ID_CURSOR,':VALOR',VALOR);
    DBMS_SQL.BIND_VARIABLE(ID_CURSOR,':CODIGO',CODIGO);

    NUM_FILAS:=DBMS_SQL.EXECUTE(ID_CURSOR);
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000,'ERROR EN EL COMANDO'||sqlerrm);
END;
```

4. Para ejecutarlo, podemos probar este bloque PL/SQL. Luego comprobamos en la tabla que ha funcionado.

```
EXECUTE GEST_EMPLE.ACT_EMPLE(100,'SALARY','100000');
```

5. Vamos a crear ahora un ejemplo con BIND_ARRAY
6. Creamos la siguiente tabla denominada SALARIOS_EMPLE

COLUMNA	TIPO
COD_EMPLEADO	NUMBER
SALARIO BRUTO	NUMBER
IMPUESTOS	NUMBER
SALARIO NETO	NUMBER

```
CREATE TABLE SALARIOS_EMPLE
(
    COD_EMPLEADO NUMBER,
    SALARIO_BRUTO NUMBER,
    IMPUESTOS NUMBER,
```

```
SALARIO_NETO NUMBER
)
```

7. Crear un procedimiento denominada CARGAR_SALARIOS. Este procedimiento deberá leer los datos de la tabla EMPLOYEES y cargarlos en la table SALARIOS_EMPLE
 - Como argumento le pasamos un número que será el tanto por ciento a aplicar de impuestos
 - Debe cargar en un 4 NESTED TABLE los datos de la tabla EMPLOYEES, en concreto el EMPLOYEE_ID, el SALARIO, el impuesto y el salario neto con el tanto por ciento que hemos pasado como argumento.
 - Mediante SQL dinámico y BIND_ARRAY cargamos los datos en la tabla CARGAR_SALARIOS

Ejemplo

```
CREATE OR REPLACE PROCEDURE CARGAR_SALARIOS(IMPUESTO NUMBER)
IS
    EMPLEADOS DBMS_SQL.NUMBER_TABLE;
    SALARIOS DBMS_SQL.NUMBER_TABLE;
    IMPUESTOS DBMS_SQL.NUMBER_TABLE;
    NETO DBMS_SQL.NUMBER_TABLE;

    ID_CURSOR INTEGER;
    NUM_FILAS INTEGER;
BEGIN
    SELECT EMPLOYEE_ID,SALARY,SALARY*IMPUESTO/100,SALARY -
    (SALARY*IMPUESTO/100) BULK COLLECT INTO
    EMPLEADOS,SALARIOS,IMPUESTOS,NETO
    FROM EMPLOYEES;
    ID_CURSOR:=DBMS_SQL.OPEN_CURSOR;
    DBMS_SQL.PARSE(ID_CURSOR,'INSERT INTO SALARIOS_EMPLE VALUES
    (:EMPLEADOS,:SALARIOS,:IMPUESTOS,:NETO)',DBMS_SQL.NATIVE);
    DBMS_SQL.BIND_ARRAY(ID_CURSOR,':EMPLEADOS',EMPLEADOS);
    DBMS_SQL.BIND_ARRAY(ID_CURSOR,':SALARIOS',SALARIOS);
    DBMS_SQL.BIND_ARRAY(ID_CURSOR,':IMPUESTOS',IMPUESTOS);
    DBMS_SQL.BIND_ARRAY(ID_CURSOR,':NETO',NETO);
    NUM_FILAS:=DBMS_SQL.EXECUTE(ID_CURSOR);
    DBMS_SQL.CLOSE_CURSOR(ID_CURSOR);
END;
```

Para probarlo

```
EXECUTE CARGAR_SALARIOS(5);
```

Luego miramos la tabla SALARIOS_EMPLE