

Prácticas de SQL DINÁMINO NATIVO

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

1. Creamos un procedimiento denominado LANZAR_COMANDO con un argumento de tipo VARCHAR2.
 - Lanzar cualquier tipo de comando, CREATE, ALTER, etc...
 - Ejecutarlo con EXECUTE IMMEDIATE
 - Con una excepción que indique si el comando es incorrecto

Ejemplo

```
CREATE OR REPLACE PROCEDURE LANZAR_COMANDO(COMANDO VARCHAR2)
AUTHID CURRENT_USER
IS
BEGIN
    EXECUTE IMMEDIATE COMANDO;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000,'ERROR EN EL COMANDO'||sqlerrm);
END;
/
```

2. Para probarlo, Podemos lanzar algún comando, por ejemplo:

```
EXECUTE LANZAR_COMANDO('CREATE TABLE T2 (CODIGO NUMBER, NOMBRE
VARCHAR2(100) )');

DESC T2
Nombre ¿Nulo? Tipo
-----
CODIGO      NUMBER
NOMBRE      VARCHAR2(100)
```

3. Crear un procedimiento que cree un sinónimo de una tabla, que tenga dos argumentos:
 - El nombre de una tabla
 - El nombre de el sinónimo que queremos crear sobre la tabla.

Ejemplo

```
CREATE OR REPLACE PROCEDURE CREAM_SINONIMO (TABLA VARCHAR2, SINONIMO
VARCHAR2)
AUTHID CURRENT_USER
IS
    COMANDO VARCHAR2(100);
BEGIN
    COMANDO:='CREATE SYNONYM '||SINONIMO||' FOR '||TABLA;
    EXECUTE IMMEDIATE COMANDO;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000,'ERROR EN EL COMANDO'||sqlerrm);
END;
/
```

4. Para probarlo, podemos lanzarlo y probar que ha funcionado.

```
EXECUTE CREAM_SINONIMO('EMPLOYEES','EMPLE');
SELECT * FROM EMPL;
```

5. Creamos una tabla denominada EMPLEADOS2 a partir de la tabla EMPLOYEES

```
CREATE TABLE EMLEADOS2 AS SELECT * FROM EMPLOYEES;
```

6. Crear una función que se llame SUBIR_SUELDO que reciba dos argumentos:
- Código de departamento
 - Tanto por ciento que se sube el salario
 - Debe subir ese tanto por ciento a todos los empleados de ese departamento
 - Debe devolver el número de empleados que se ha visto aumentado el salario
 - Por supuesto, aunque no haría falta, debemos usar SQL dinámico

Ejemplo

```
CREATE OR REPLACE FUNCTION SUBIR_SUELDO(COD_DEPARTAMENTO
NUMBER,SUBIDA NUMBER)
RETURN NUMBER
IS
    COMANDO VARCHAR2(100);
    NUM_FILAS INTEGER;
BEGIN
    COMANDO:='UPDATE EMPLEADOS2 SET SALARY=(SALARY)+SALARY*'||SUBIDA||'/100
WHERE DEPARTMENT_ID='||COD_DEPARTAMENTO;
    EXECUTE IMMEDIATE COMANDO;
    NUM_FILAS:=SQL%ROWCOUNT;
    RETURN NUM_FILAS;
```

```
EXCEPTION
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20000,'ERROR EN EL COMANDO'||sqlerrm);
END;
/
```

7. Para probar que funciona Podemos hacer el siguiente bloque. También Podemos comprobar la table, para ver que se han actualizado los salarios

```
DECLARE
  NUMERO_UPDATES INTEGER;
BEGIN
  NUMERO_UPDATES:=SUBIR_SUELDO(10,2);
  DBMS_OUTPUT.PUT_LINE('SE HAN MODIFICADO '||NUMERO_UPDATES||
  EMPLEADOS');
END;
/
```

Procedimiento PL/SQL terminado correctamente.

SE HAN MODIFICADO 6 EMPLEADOS