

Módulo: Bases de Datos

Unidad 6: Programación de Bases de Datos

Sesión 7: Cursores

Descripción:

Durante esta unidad se ha estudiado como implementar cursores dentro de bloques PL/SQL. Para practicar todos estos conceptos se propone la realización del siguiente caso práctico:

Cree un bloque PL/SQL que realice las siguientes acciones:

- En la sección de declaraciones, declare e inicialice una variable llamada v_deptno de tipo NUMBER. Asigne un valor de identificador de departamento válido.
- Declare un cursor llamado c_emp_cursor, que recupere last_name, salary manager_id de los empleados que trabajan en el departamento especificado en v_deptno.
- En la sección ejecutable, utilice el bucle FOR de cursor para realizar operaciones en los datos recuperados. Si el salario del empleado es menor que 5.000 y si el identificador de su superior es 101 o 124, aparece el mensaje "<<last_name>> Merece un aumento". De lo contrario, aparece el mensaje "<<last_name>> No merece un aumento"

Criterios de Evaluación:

- RA05_d: Se han definido y utilizado guiones para automatizar tareas.
- RA05_i: Se han utilizado cursores.

Objetivos:

- Definir y utilizar cursores.

Recursos:

- Acceso a Internet.
- Oracle Express.
- SQL Developer

Conceptos a revisar previamente:

- Realizar el estudio de los apartados de la unidad:
 - Cursores. Funciones de tratamiento de cursores.

- Realizar Ejercicios Resueltos:
 - Cursores



Resolución de la práctica:

▪ Ejercicio de Cursores

Realizamos el recorrido del cursor por medio del bucle FOR que crea un registro llamado “emp_record” en el que va copiando en cada vuelta del bucle el valor del cursor para poder usarlo, hasta que se termine el cursor.

```
SET SERVEROUTPUT ON;

DECLARE
v_deptno employees.department_id%TYPE:= &introduzca_el_valor_del_departamento;

CURSOR c_emp_cursor IS
  SELECT      last_name,      salary,      manager_id      FROM      EMPLOYEES      WHERE
department_id=v_deptno;

BEGIN

  FOR emp IN c_emp_cursor
  LOOP
    IF (emp.salary<5000) AND (emp.manager_id IN (101,124))
    THEN
      DBMS_OUTPUT.PUT_LINE(emp.last_name || ' merece un aumento');
    ELSE
      DBMS_OUTPUT.PUT_LINE(emp.last_name || ' no merece un aumento');
    END IF;
  END LOOP;
END;
```

En el siguiente ejemplo recorreremos el cursor con un bucle normal, por lo que tenemos que abrir el cursor primero, después creamos el bucle que va cogiendo el dato correspondiente del cursor con FETCH y lo guarda en unas variables que creamos.

El bucle tiene la instrucción EXIT justo después del FETCH para comprobar si no hay más datos en el cursor y entonces salirse del bucle antes de usar las variables que no tendrían el dato correcto.

Debemos recordar cuando termine el bucle de cerrar el cursor.

```
--EXPLICITO
SET SERVEROUTPUT ON;

DECLARE
v_deptno employees.department_id%TYPE:= &introduzca_el_valor_del_departamento;
```

```
v_last_name employees.last_name%TYPE;
v_salary employees.salary%TYPE;
v_manager_id employees.manager_id%TYPE;

CURSOR c_emp_cursor IS
  SELECT last_name, salary, manager_id FROM EMPLOYEES WHERE
  department_id=v_deptno;

BEGIN
  OPEN c_emp_cursor;

  LOOP
    FETCH c_emp_cursor INTO v_last_name, v_salary, v_manager_id;
    EXIT WHEN c_emp_cursor%NOTFOUND;
    IF (v_salary<5000) AND (v_manager_id IN (101,124))
    THEN
      DBMS_OUTPUT.PUT_LINE(v_last_name || ' merece un aumento');
    ELSE
      DBMS_OUTPUT.PUT_LINE(v_last_name || ' no merece un aumento');
    END IF;
  END LOOP;

  CLOSE c_emp_cursor;
END;
```