

Prácticas de REF CURSOR

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!!!!

REF CURSORS

1. Crear un REF CURSOR para que visualice el primer nombre de región de la tabla REGIONS

Ejemplo

```
SET SERVEROUTPUT ON
DECLARE
  --DECLARAMOS EL TIPO DE CURSOR
  TYPE C_VARIABLE IS REF CURSOR;
  --VARIABLE PARA PONER LA REGION
  REGIONES REGIONS%ROWTYPE;
  --DECLARAMOS UNA VARIABLE DEL TIPO DEL CURSO VARIABLE
  C1 C_VARIABLE;
BEGIN
  --ABRIMOS EL CURSO CON LA SELECT CORRESPONDIENTE
  OPEN C1 FOR SELECT * FROM REGIONS;
  --PONEMOS LA PRIMERA FILA EN LA VARIABLE C1
  FETCH C1 INTO REGIONES;
  --VISUALIZAMOS
  DBMS_OUTPUT.PUT_LINE(REGIONES.REGION_NAME);
  --CERRAMOS EL CURSOR
  CLOSE C1;
END;
/

Europe

Procedimiento PL/SQL terminado correctamente.
```

2. Con el mismo cursor visualizamos la primera ciudad de la tabla LOCATIONS

Ejemplo:

SET SERVEROUTPUT ON

DECLARE

```
--DECLARAMOS EL TIPO DE CURSOR
TYPE C_VARIABLE IS REF CURSOR;
--VARIABLE PARA PONER LA REGION
REGIONES REGIONS%ROWTYPE;
--VARIABLE PARA PONER LA LOCATION
LOCALIDADES LOCATIONS%ROWTYPE;
--DECLARAMOS UNA VARIABLE DEL TIPO DEL CURSO VARIABLE
C1 C_VARIABLE;
```

BEGIN

```
--ABRIMOS EL CURSO CON LA SELECT CORRESPONDIENTE
OPEN C1 FOR SELECT * FROM REGIONS;
--PONEMOS LA PRIMERA FILA EN LA VARIABLE C1
FETCH C1 INTO REGIONES;
--VISUALIZAMOS
DBMS_OUTPUT.PUT_LINE(REGIONES.REGION_NAME);
--CERRAMOS EL CURSOR
CLOSE C1;
--HACEMOS LO MISMO CON LA TABLA LOCATIONS
--ABRIMOS EL CURSO CON LA SELECT CORRESPONDIENTE
OPEN C1 FOR SELECT * FROM LOCATIONS;
--PONEMOS LA PRIMERA FILA EN LA VARIABLE C1
FETCH C1 INTO LOCALIDADES;
--VISUALIZAMOS
DBMS_OUTPUT.PUT_LINE(LOCALIDADES.CITY);
--CERRAMOS EL CURSOR
CLOSE C1;
```

END;

/

Europe

Roma

Procedimiento PL/SQL terminado correctamente.

3. Crear un procedimiento que tenga un parámetro de tipo numérico. Si es un 1 debe visualizar todos los nombres de regiones. Si es un 2 debe visualizar todas las ciudades. Por supuesto, usamos un REF_CURSOR para visualizar los datos. Debemos usar un solo bucle para hacer el trabajo

Ejemplo

```
CREATE OR REPLACE PROCEDURE EJEMPLO_CURSOR(TIPO NUMBER)
IS
  TYPE C_VARIABLE IS REF CURSOR;
  DATO VARCHAR2(50);
  C1 C_VARIABLE;
BEGIN
  IF TIPO=1 THEN
    OPEN C1 FOR SELECT REGION_NAME FROM REGIONS;
  END IF;
  IF TIPO=2 THEN
    OPEN C1 FOR SELECT CITY FROM LOCATIONS;
  END IF;
  FETCH C1 INTO DATO;
  WHILE C1%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(DATO);
    FETCH C1 INTO DATO;
  END LOOP;
END;
/

--Para probarlo, por ejemplo con las ciudades
EXECUTE EJEMPLO_CURSOR(2);
```

4. Crear una función que reciba como argumento un salario (tipo NUMBER) y que devuelva un REF CURSOR de tipo EMPLOYEES con los empleados que ganen más de esa salario. (Pista: usamos SYS_REFCURSOR). Crear luego un bloque PL/SQL anónimo que llame a la función para probarla.

Por ejemplo, la función

```
CREATE OR REPLACE FUNCTION DEVOLVER_EMPLEADOS(SALARIO NUMBER)
RETURN SYS_REFCURSOR
IS
  TYPE EMPLEADOS IS REF CURSOR RETURN EMPLOYEES%ROWTYPE;
  EMPL EMPLEADOS;
BEGIN
  OPEN EMPL FOR SELECT * FROM EMPLOYEES WHERE SALARY > SALARIO;
  RETURN EMPL;
```

```
END;  
/
```

Ejemplo del bloque PL/SQL para probarlo. En este caso indicando un salario de 5000

```
DECLARE  
  EMPLEADOS SYS_REFCURSOR;  
  SALARIO NUMBER;  
  EMPLEADO EMPLOYEES%ROWTYPE;  
BEGIN  
  SALARIO:=5000;  
  EMPLEADOS:=DEVOLVER_EMPLEADOS(SALARIO);  
  LOOP  
    FETCH EMPLEADOS INTO EMPLEADO;  
    DBMS_OUTPUT.PUT_LINE(EMPLEADO.FIRST_NAME||  
'||EMPLEADO.SALARY);  
    EXIT WHEN EMPLEADOS%NOTFOUND;  
  END LOOP;  
END;  
/
```

5. Vamos ahora a crear un paquete que tengan las siguientes características:
 - Declarar una variable de tipo REF_CURSOR de tipo EMPLOYEES.
 - Creamos también una función dentro del paquete que tenga como argumento el REF_CURSOR creado anteriormente y que devuelve la media de los salarios.
 - Desde un bloque PL/SQL anónimos creamos una variable REF_CURSOR del tipo definido en el paquete. La rellenamos con una SELECT de EMPLOYEES que contengan alguna condición, por ejemplo los que trabajen en un departamento. Pasamos ese cursor a la función del paquete para que nos devuelva la suma de la media de los salarios.

Por ejemplo, el paquete podría quedar así:

```
CREATE OR REPLACE PACKAGE PAQ_CURSOR  
IS  
  TYPE EMPL IS REF CURSOR RETURN EMPLOYEES%ROWTYPE;  
  FUNCTION MEDIA_SALARIOS(EMPLEADOS EMPL) RETURN NUMBER;  
END;  
/  
  
CREATE OR REPLACE PACKAGE BODY PAQ_CURSOR  
IS
```

```

FUNCTION MEDIA_SALARIOS(EMPLEADOS EMPL) RETURN NUMBER
IS
  --DECLARAMOS LAS VARIABLES A UTILIZAR
  SALARIOS NUMBER:=0;
  MEDIA NUMBER:=0;
  NUM_EMPL INTEGER:=0;
  EMPLEADO EMPLOYEES%ROWTYPE;
BEGIN
  LOOP
    --VAMOS LEYENDO CADA EMPLEADO
    FETCH EMPLEADOS INTO EMPLEADO;

    --SALIMOS SI YA NO HAY EMPLEADOS
    EXIT WHEN EMPLEADOS%NOTFOUND;

    --VAMOS SUMANDO EL SALARIO DE CADA EMPLEADO
    SALARIOS:=SALARIOS+EMPLEADO.SALARY;
    --VAMOS CONTANDO EL NÚMERO DE EMPLEADOS
    NUM_EMPL:=NUM_EMPL+1;
  END LOOP;
  --AVERIGAMOS LA MEDIA Y LA DEVOLVEMOS
  MEDIA:=SALARIOS/NUM_EMPL;
  RETURN MEDIA;
END MEDIA_SALARIOS;
END;
/

```

Y el bloque PL/SQL que llama podría quedar de la siguiente forma:

```

DECLARE
  EMPLEADOS PAQ_CURSOR.EMPLE;
  MEDIA NUMBER;
BEGIN
  OPEN EMPLEADOS FOR SELECT * FROM EMPLOYEES WHERE
  DEPARTMENT_ID=100;
  MEDIA:=paq_cursor.media_salarios(EMPLEADOS);
  DBMS_OUTPUT.PUT_LINE('LA MEDIA DE LOS EMPLEADOS DE ESTE
  REF_CURSRO ES:'||MEDIA);
END;
/

```