### CURSORES

Para poder procesar una orden SQL, Oracle asigna un área de memoria que recibe el nombre de área de contexto. Esta área contiene informaciones necesarias para completar el procesamiento, incluyendo el número de filas procesadas por la orden, un puntero a la versión analizada de la orden y, en el caso de las consultas, el conjunto activo, que es el conjunto de filas resultado de la consulta.

Un cursor es un puntero al área de contexto. Mediante el cursor, un programa PL/SQL puede controlar el área de contexto y lo que en ella sucede a medida que se procesa la orden.

Los cursores pueden ser implícitos (devuelven una sola fila) o explícitos (devuelven más de una).

**IMPLÍCITO**

Mostrar el oficio del empleado REY.

DECLARE

p\_oficio VARCHAR2(15);

BEGIN

select oficio into p\_oficio from emp where upper(apellido)='REY';

dbms\_output.put\_line(p\_oficio);

END;

**EXPLÍCITO**

La declaración de un cursor define su nombre, y asocia el cursor con una orden SELECT. La sintaxis es:

CURSOR nombre\_cursor IS orden\_SELECT;

Mostar apellido y salario de todos los empleados

**declare**

**v\_ape EMP.APELLIDO%TYPE;**

**v\_sal EMP.SALARIO%TYPE;**

**--declaramos nuestro cursor con una consulta**

**--la consulta debe tener los mismos datos para luego**

**--hacer el fetch**

**cursor CURSOREMP is**

**select APELLIDO, SALARIO from EMP;**

**begin**

**--1) abrir el cursor**

**open CURSOREMP;**

**--BUCLE INFINITO**

**loop**

**--2) EXTRAEMOS LOS DATOS DEL CURSOR**

**fetch CURSOREMP into v\_ape, v\_sal;**

**--3) preguntamos si hemos terminado**

**exit when CURSOREMP%notfound;**

**--DIBUJAMOS LOS DATOS**

**dbms\_output.put\_line('Apellido: ' || v\_ape**

**|| ', Salario: ' || v\_sal);**

**end loop;**

**--4) Cerramos cursor**

**close CURSOREMP;**

**end;**

rey--650000

negro--370500

jimenez--386750

cerezo--318500

arroyo--208000

sala--162500

martin--162500

tovar--195000

jimeno--123500

fernandez--390000

gil--390000

sanchez--104000

alonso--143000

muñoz--169000

**ATRIBUTOS**

Un cursor posee una serie de atributos predefinidos para conocer su estado. Son los siguientes:

**%NOTFOUND:** Se activa si el último FETCH no ha recuperado ninguna fila del cursor.

Ej: LOOP

FETCH CURSOR INTO VARIABLE;

EXIT WHEN CURSOR%NOTFOUND;

END LOOP;

**%FOUND:** Se activa si el último FETCH ha recuperado alguna fila del cursor

ej: LOOP

FETCH CURSOR INTO VARIABLE

IF CURSOR%FOUND THEN

----;

ELSE

EXIT;

END IF;

END LOOP;

**%ROWCOUNT:** Devuelve el número de filas recuperadas hasta ese momento con la instrucción FETCH, al inicio vale 0:

ej: LOOP

FETCH CURSOR INTO VARIABLE

IF CURSOR%ROWCOUNT = 5

THEN EXIT

END IF

END LOOP;

**%ISOPEN:** Es verdadero cuando el cursor está abierto, falso si está cerrado.

Ej: IF CURSOR%ISOPEN THEN

CLOSE CURSOR;

ELSE

OPEN CURSOR;

END IF;

**Atributos con consultas de acción**

Los atributos se pueden usar con las instrucciones INSERT, UPDATE o DELETE para comprobar si devuelven o no datos y el número de registros afectados por la consulta.

Ejemplo:

begin

update emp

set comision=comision\*1.10

where dept\_no = 10;

if **SQL%FOUND** then

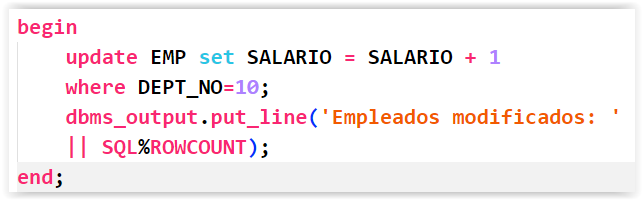
dbms\_output.put\_line('Se incrementa 10% a: '||to\_char(**SQL%ROWCOUNT**)|| ' empleados ' );

end if;

end;

RESULTADO:

Se incrementa 10% a: 6 empleados



Realizar un código PL/SQL dónde pediremos el número, nombre y localidad de un departamento.

Si el departamento existe, modificamos su nombre y localidad

si el departamento no existe, lo insertamos.

BLOQUE CON CURSOR EXPLICITO

**declare**

**v\_id DEPT.DEPT\_NO%TYPE;**

**v\_nombre DEPT.DNOMBRE%TYPE;**

**v\_localidad DEPT.LOC%TYPE;**

**v\_existe DEPT.DEPT\_NO%TYPE;**

**cursor CURSORDEPT is**

**select DEPT\_NO from DEPT**

**where DEPT\_NO=v\_id;**

**begin**

**v\_id := &iddepartamento;**

**v\_nombre := '&nombre';**

**v\_localidad := '&localidad';**

**open CURSORDEPT;**

**fetch CURSORDEPT into v\_existe;**

**if (CURSORDEPT%found) then**

**dbms\_output.put\_line('UPDATE');**

**update DEPT set DNOMBRE=v\_nombre, LOC=v\_localidad**

**where DEPT\_NO=v\_id;**

**else**

**dbms\_output.put\_line('INSERT');**

**insert into DEPT values (v\_id, v\_nombre, v\_localidad);**

**end if;**

**close CURSORDEPT;**

**end;**

**BLOQUE CON CURSOR IMPLICITO**

**declare**

**v\_id DEPT.DEPT\_NO%TYPE;**

**v\_nombre DEPT.DNOMBRE%TYPE;**

**v\_localidad DEPT.LOC%TYPE;**

**v\_existe DEPT.DEPT\_NO%TYPE;**

**begin**

**v\_id := &iddepartamento;**

**v\_nombre := '&nombre';**

**v\_localidad := '&localidad';**

**select COUNT(DEPT\_NO) into v\_existe from DEPT**

**where DEPT\_NO=v\_id;**

**if (v\_existe = 0) then**

**dbms\_output.put\_line('Insert');**

**else**

**dbms\_output.put\_line('Update');**

**end if;**

**end;**

Realizar un código pl/sql para modificar el salario del empleado ARROYO.

Si el empleado cobra más de 250.000, le bajamos el sueldo en 10.000.

Si no, le subimos el sueldo en 10.000

**declare**

**v\_salario EMP.SALARIO%TYPE;**

**v\_idemp EMP.EMP\_NO%TYPE;**

**begin**

**select EMP\_NO, SALARIO into v\_idemp, v\_salario from EMP**

**where UPPER(APELLIDO)='ARROYO';**

**if v\_salario > 250000 then**

**v\_salario := v\_salario - 10000;**

**else**

**v\_salario := v\_salario + 10000;**

**end if;**

**update EMP set SALARIO=v\_salario**

**where EMP\_NO=v\_idemp;**

**dbms\_output.put\_line('Salario modificado: ' || v\_salario);**

**end;**

Ejemplo Hospital y suma salarial

--Necesito mostrar la suma salarial de los doctores de la paz.

--Realizar el siguiente código pl/sql.

--Necesitamos modificar el salario de los doctores de LA PAZ.

--Si la suma salarial supera 1.000.000 bajamos salarios en 10.000 a todos

--Si la suma salarial no supera el millón, subimos salarios en 10.000

--Mostrar el número de filas que hemos modificado (subir o bajar)

--Doctores con suerte: 6, Doctores más pobres: 6

**SOLUCION 1**

**declare**

**v\_suma\_salarial *NUMBER*;**

**begin**

**select sum(DOCTOR.SALARIO) into v\_suma\_salarial**

**from DOCTOR**

**inner join HOSPITAL**

**on DOCTOR.HOSPITAL\_COD = HOSPITAL.HOSPITAL\_COD**

**where lower(HOSPITAL.NOMBRE)='la paz';**

**dbms\_output.put\_line('Suma salarial La paz: ' || v\_suma\_salarial);**

**if v\_suma\_salarial > 1000000 then**

**update DOCTOR set SALARIO = SALARIO - 10000**

**where hospital\_cod=**

**(select HOSPITAL\_COD from HOSPITAL where UPPER(NOMBRE)='LA PAZ');**

**dbms\_output.put\_line('Bajando salarios: ' || SQL%ROWCOUNT);**

**else**

**update DOCTOR set SALARIO = SALARIO + 10000**

**where hospital\_cod=**

**(select HOSPITAL\_COD from HOSPITAL where UPPER(NOMBRE)='LA PAZ');**

**dbms\_output.put\_line('Doctores ricos ' || SQL%ROWCOUNT);**

**end if;**

**end;**

**SOLUCION 2**

**declare**

**v\_suma\_salarial *NUMBER*;**

**v\_codigo HOSPITAL.HOSPITAL\_COD%TYPE;**

**begin**

**select HOSPITAL\_COD into v\_codigo from HOSPITAL**

**where lower(NOMBRE)='la paz';**

**select sum(SALARIO) into v\_suma\_salarial**

**from DOCTOR**

**where HOSPITAL\_COD=v\_codigo;**

**dbms\_output.put\_line('Suma salarial La paz: ' || v\_suma\_salarial);**

**if v\_suma\_salarial > 1000000 then**

**update DOCTOR set SALARIO = SALARIO - 10000**

**where hospital\_cod=v\_codigo;**

**dbms\_output.put\_line('Bajando salarios: ' || SQL%ROWCOUNT);**

**else**

**update DOCTOR set SALARIO = SALARIO + 10000**

**where hospital\_cod=v\_codigo;**

**dbms\_output.put\_line('Doctores ricos ' || SQL%ROWCOUNT);**

**end if;**

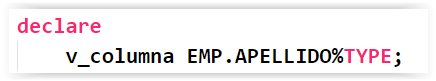
**end;**

**Tipo de dato** **ROWTYPE (TABLE RECORD)**

La declaración de una variable como ROWTYPE implica que dicha variable será de tipo FILA de una tabla determinada.

Cuando declaramos una variable como TYPE, estamos indicando que una variable será del tipo COLUMNA de una tabla.

Ejemplo:

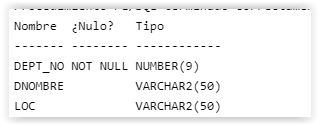


La variable **v\_columna** tendrá el mismo tipo de dato que la columna APELLIDO de la tabla EMP.

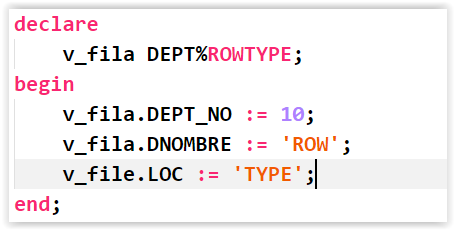
Las variables declaradas como ROWTYPE son iguales, solo que tendrán la información de toda la fila de una tabla. Por ejemplo, si declaramos una variable ROWTYPE de la tabla DEPT:



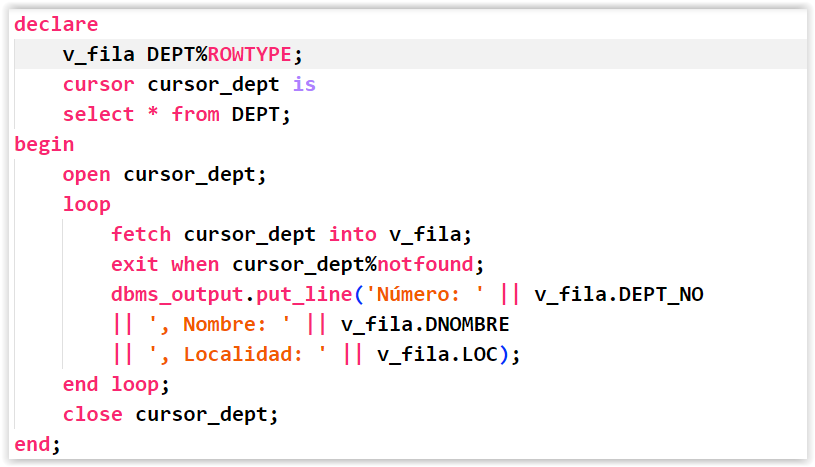
La variable **v\_fila** será una variable que contendrá las columnas de la tabla DEPT:



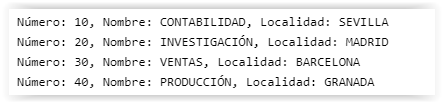
Una vez declarada la variable, podemos acceder a sus propiedades mediante el nombre de la variable seguido de un punto y el nombre de la columna de la tabla:



Por ejemplo, podemos realizar un cursor explícito para recorrer todas las filas de la tabla DEPT con este tipo de dato:



Y podremos visualizar el resultado:



**USO DEL BUCLE FOR…IN CON CURSORES**

Sintaxis:

FOR REGISTRO IN NOMBRE\_CURSOR LOOP

...

END LOOP;

Simplifica el uso de cursores al realizar las siguientes operaciones:

* Declara implícitamente la variable registro.
* Se ejecuta implícitamente el OPEN del cursor antes de entrar en el bucle.
* Se ejecuta implícitamente un FECTH en cada repetición del bucle.
* Sale automáticamente del bucle sino hay más tuplas.
* Se ejecuta implícitamente el CLOSE del cursor cuando sale del bucle.

Ejemplo:

DECLARE

CURSOR CURSOR\_DOCTOR IS

SELECT apellido,especialidad

FROM doctor;

BEGIN

FOR V\_REG\_DOCTOR IN CURSOR\_DOCTOR LOOP

DBMS\_OUTPUT.PUT\_LINE(V\_REG\_DOCTOR.apellido || ' ' ||

V\_REG\_DOCTOR.especialidad);

END LOOP;

END;

RESULTADO:

Lopez A. Cardiologia

Miller G. Ginecologia

Cajal R Cardiologia

Galo C. Pediatria

Best K. Urologia

Cabeza D. Psiquiatria

Niqo P. Pediatria

Adams C. Neurologia

diaz b. Cardiologia

**EJEMPLO**

**declare**

**cursor cursor\_emp is**

**select apellido, salario, oficio,**

**salario + comision as total**

**from EMP;**

**begin**

**for v\_registro in cursor\_emp**

**loop**

**dbms\_output.put\_line('Apellido ' || v\_registro.apellido**

**|| ', Salario: ' || v\_registro.salario**

**|| ', Oficio: ' || v\_registro.oficio**

**|| ', Total: ' || v\_registro.total);**

**end loop;**

**end;**