



Uso de Cursores Explícitos Simples

MDY3131

Experiencia de Aprendizaje y Competencia Asociada



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES

Experiencia	Nombre	Unidad de Competencia Especialidad – Nivel de la Competencia de Empleabilidad
Nº 2	Construyendo Bloques Anónimos PL/SQL Complejos	Desarrolla operaciones sobre la base de datos para efectuar procesamiento de datos utilizando el lenguaje asociado de acuerdo a los requerimientos de información.
		Resolución de Problemas (N1)

- Cómo usar los Atributo %ISOPEN, %NOTFOUND, %FOUND y %ROWCOUNT para obtener información del estado de un Cursor Explícito.
- Cómo procesar las filas de un Cursor Explícito usando la estructura de iteración LOOP Simple.
- Cómo procesar las filas de un Cursor Explícito usando la estructura de iteración WHILE LOOP.
- Cómo procesar las filas de un Cursor Explícito usando la estructura de iteración FOR LOOP.

Manejo de Cursores Explícitos

1. Los bloques PL/SQL de los ejemplos usan tablas del esquema HR de la Base de Datos Oracle y algunas tablas nuevas. Por esta razón, en las sentencias DML (INSERT, UPDATE y DELETE) de los bloques se usan copias de esas tablas del esquema HR para no realizar los cambios de datos en las tablas originales.
2. De acuerdo con esto, antes de que Ud. ejecute los ejemplos debe crear las tablas necesarias con el archivo [script_crea_tablas_ejemplos.sql](#)

Atributos para Cursores Explícitos

%ISOPEN: para
saber si el cursor
se encuentra
abierto

%NOTFOUND:
para saber si el
FETCH más
reciente no
retornó filas

Para obtener
información del
estado de un
cursor explícito se
pueden usar estos
cuatro atributos

%FOUND: para
saber si el FETCH
más reciente
retornó una fila

%ROWCOUNT:
para saber el
número total de
filas leídas del
cursor

Atributo %ISOPEN

Se pueden leer filas
sólo cuando el
cursor se encuentra
abierto

El atributo de
cursor %ISOPEN
permite validar si el
cursos está abierto
antes de realizar
una lectura

Retorna el estado
del cursor. TRUE si
está abierto y FALSE
si no está abierto

- Sintaxis:

***nombre_cursor*%ISOPEN**

- Ejemplo

```
BEGIN
  IF NOT cur_dato_empleado%ISOPEN THEN
    OPEN cur_dato_empleado;
  END IF;
  LOOP
    FETCH cur_dato_empleado INTO reg_empleado;
```

Atributo %ROWCOUNT

Retorna el número
de filas que se han
leído desde el
cursor

El atributo
%ROWCOUNT se
incrementa en
uno en cada
lectura

Se puede
referenciar en
sentencias PL/SQL

- Sintaxis:

nombre_cursor%ROWCOUNT

- Ejemplo

```
BEGIN
  OPEN cur_dato_empleado;
  LOOP
    FETCH cur_dato_empleado INTO reg_empleado;
    EXIT WHEN cur_dato_empleado%ROWCOUNT > 10;
  END LOOP;
  CLOSE cur_dato_empleado;
END;
```

Atributo %NOTFOUND

Retorna el número
de filas que se han
leído desde el
cursor

El atributo
%ROWCOUNT se
incrementa en
uno en cada
lectura

Se puede
referenciar en
sentencias PL/SQL

- Sintaxis:

nombre_cursor%ROWCOUNT

- Ejemplo

```
BEGIN
  OPEN cur_dato_empleado;
  LOOP
    FETCH cur_dato_empleado INTO reg_empleado;
    EXIT WHEN cur_dato_empleado%NOTFOUND;
  END LOOP;
  CLOSE cur_dato_empleado;
END;
```


LOOP Simple para leer las Filas del Cursor

Para poder iterar a través del cursor se puede usar LOOP Simple

Esto permite leer las filas del cursor de acuerdo a la condición de salida del loop

- Sintaxis:

```
OPEN nombre_cursor;  
LOOP  
    FETCH nombre_cursor INTO lista_variables | registro_PL/SQL;  
    EXIT WHEN nombre_cursor%NOTFOUND | EXIT WHEN nombre_cursor%ROWCOUNT;  
    /* Procesamiento del SET ACTIVO del Cursor y ejecución de sentencias SQL y PL/SQL */  
END LOOP;  
CLOSE nombre_cursor;
```

LOOP Simple para leer las Filas del Cursor

- Ejemplo:

```
VAR b_por_aum1 NUMBER
VAR b_por_aum2 NUMBER
EXEC :b_por_aum1:=.25
EXEC :b_por_aum2:=.15
DECLARE
CURSOR cur_datos_emp IS
  SELECT e.employee_id, e.salary, ROUND(MONTHS_BETWEEN(SYSDATE,e.hire_date)/12),l.city
  FROM employees e LEFT OUTER JOIN departments d
    ON(e.department_id = d.department_id)
  LEFT OUTER JOIN locations l
    ON(d.location_id = l.location_id)
  ORDER BY e.employee_id;
TYPE tp_varray_porc_aum IS VARRAY(2)
  OF NUMBER;
varray_porc_aum tp_varray_porc_aum;
v_id_emp NUMBER(3);
v_salario NUMBER(8);
v_annos NUMBER(2);
v_ciudad VARCHAR2(30);
v_sal_act NUMBER(8);
v_bonif_annos NUMBER(8);
BEGIN
  OPEN cur_datos_emp;
  varray_porc_aum:= tp_varray_porc_aum(:b_por_aum1,:b_por_aum2);
  -- El bloque continúa en la siguiente PPT
```

LOOP Simple para leer las Filas del Cursor

LOOP

```
/* Las variables se inicializan en cero para que cuando el empleado no cumpla con la(s) condición(es)
   las operaciones matemáticas sean CERO, no NULL */
v_bonif_annos:=0;
v_sal_act:=0;
FETCH cur_datos_emp INTO v_id_emp,v_salario,v_annos,v_ciudad;
EXIT WHEN cur_datos_emp%NOTFOUND;
    IF v_annos >=15 THEN -- Calcula bonificación por años trabajados
        SELECT ROUND(v_salario*(porc_bonif/100))
            INTO v_bonif_annos
            FROM tramo_bonif_annos_trab
            WHERE v_annos BETWEEN rango_ini AND rango_fin;
    END IF;
v_sal_act:=v_salario+v_bonif_annos; -- salario a actualizar será el salario actual más bonificación por años
-- Calcula bonificación extra por la ciudad en que trabaja y se le suma al salario a actualizar
IF v_ciudad ='Toronto' THEN
    v_sal_act:= v_sal_act + ROUND(v_salario*varray_porc_aum(1));
ELSIF v_ciudad='London' THEN
    v_sal_act:= v_sal_act + ROUND(v_salario*varray_porc_aum(2));
END IF;
IF v_sal_act > 0 THEN -- Se actualiza si es que hubo algún cambio en el salario
    UPDATE empleados
        SET salary=v_sal_act
        WHERE employee_id=v_id_emp;
    END IF;
END LOOP;
CLOSE cur_datos_emp;
END;
```

WHILE LOOP para leer las Filas del Cursor

Cuando se usa esta Estructura de Iteración, la cláusula **FETCH** se usa dos veces

La iteración del **LOOP** se realiza mientras se cumpla la condición especificada

Se pueden usar el atributo **%FOUND** para validar si se leyó una fila del cursor

- Sintaxis:

```
OPEN nombre_cursor;  
FETCH nombre_cursor INTO lista_variables | registro_PL/SQL;  
WHILE nombre_cursor%FOUND | nombre_cursor%ROWCOUNT;  
  LOOP  
    /* Procesamiento del SET ACTIVO del Cursor y ejecución de sentencias SQL y PL/SQL */  
    FETCH nombre_cursor INTO lista_variables;  
  END LOOP;  
CLOSE nombre_cursor;
```

WHILE LOOP para leer las Filas del Cursor

- Ejemplo:

```
VAR b_por_aum1 NUMBER
VAR b_por_aum2 NUMBER
VAR b_por_aum3 NUMBER
EXEC :b_por_aum1:=.25
EXEC :b_por_aum1:=.15
DECLARE
CURSOR cur_datos_emp IS
    SELECT e.employee_id, e.salary, ROUND(MONTHS_BETWEEN(SYSDATE,e.hire_date)/12),l.city
    FROM employees e LEFT OUTER JOIN departments d
        ON(e.department_id = d.department_id)
    LEFT OUTER JOIN locations l
        ON(d.location_id = l.location_id)
    ORDER BY e.employee_id;
TYPE tp_varray_porc_aum IS VARRAY(2) OF NUMBER;
varray_porc_aum tp_varray_porc_aum;
v_id_emp NUMBER(3);
v_salario NUMBER(8);
v_años NUMBER(2);
v_ciudad VARCHAR2(30);
v_sal_act NUMBER(8);
v_bonif_años NUMBER(8);
BEGIN
    OPEN cur_datos_emp;
    varray_porc_aum:= tp_varray_porc_aum(:b_por_aum1,:b_por_aum2);
    FETCH cur_datos_emp INTO v_id_emp,v_salario,v_años,v_ciudad;
-- El bloque continúa en la siguiente PPT
```

WHILE LOOP para leer las Filas del Cursor

WHILE cur_datos_emp%FOUND LOOP

/* Las variables se inicializan en cero para que cuando el empleado no cumpla con la(s) condición(es) las operaciones matemáticas sean CERO, no NULL */

v_bonif_annos:=0;

v_sal_act:=0;

-- Calcula bonificación por años trabajados

IF v_annos >=15 THEN

SELECT ROUND(v_salario*(porc_bonif/100))

INTO v_bonif_annos

FROM tramo_bonif_annos_trab

WHERE v_annos BETWEEN rango_ini AND rango_fin;

END IF;

v_sal_act:=v_salario+v_bonif_annos; -- salario a actualizar será el salario actual más bonificación por años

-- Calcula bonificación extra por la ciudad en que trabaja y se le suma al salario a actualizar

IF v_ciudad ='Toronto' THEN

v_sal_act:= v_sal_act + ROUND(v_salario*varray_porc_aum(1));

ELSIF v_ciudad='London' THEN

v_sal_act:= v_sal_act + ROUND(v_salario*varray_porc_aum(2));

END IF;

IF v_sal_act > 0 THEN -- Se actualiza si es que hubo algún cambio en el salario

UPDATE empleados

SET salary=v_sal_act

WHERE employee_id=v_id_emp;

END IF;

FETCH cur_datos_emp INTO v_id_emp,v_salario,v_annos,v_ciudad;

END LOOP;

CLOSE cur_datos_emp;

END;

FOR LOOP para leer las Filas del Cursor

Es el método más fácil para leer filas de un Cursor ya que OPEN, FETCH, EXIT y CLOSE se ejecutan implícitamente

El loop finaliza automáticamente cuando se lee la última fila del cursor

El registro es declarado implícitamente y sus campos son las columnas del cursor

A las expresiones de la cláusula SELECT del cursor se le deben asignar alias.

- Sintaxis:

```
FOR nombre_registro IN nombre_cursor LOOP  
/* Procesamiento del SET ACTIVO del Cursor y ejecución de sentencias SQL y PL/SQL*/  
END LOOP;
```

Leer las Filas del Cursor con LOOP Simple

- Ejemplo:

```
VAR b_por_aum1 NUMBER
VAR b_por_aum2 NUMBER
EXEC :b_por_aum1:=.25
EXEC :b_por_aum2:=.15
DECLARE
CURSOR cur_datos_emp IS
  SELECT e.employee_id, e.salary, ROUND(MONTHS_BETWEEN(SYSDATE,e.hire_date)/12) annos_trab,
         l.city ciudad
  FROM employees e LEFT OUTER JOIN departments d
    ON(e.department_id = d.department_id)
   LEFT OUTER JOIN locations l
    ON(d.location_id = l.location_id)
  ORDER BY e.employee_id;
TYPE tp_varray_porc_aum IS VARRAY(2) OF NUMBER;
varray_porc_aum tp_varray_porc_aum;
v_sal_act NUMBER(8);
v_bonif_annos NUMBER(8);
BEGIN
  varray_porc_aum:= tp_varray_porc_aum(:b_por_aum1,:b_por_aum2);
```

-- El bloque continúa en la siguiente PPT

Leer las Filas del Cursor con LOOP Simple

```
FOR reg_datos_emp IN cur_datos_emp LOOP
/* Las variables se inicializan en cero para que cuando el empleado no cumpla con la(s) condición(es)
las operaciones matemáticas sean CERO, no NULL */
v_bonif_annos:=0;
v_sal_act:=0;
-- Calcula bonificación por años trabajados
IF reg_datos_emp.annos_trab >=15 THEN
SELECT ROUND(reg_datos_emp.salary*(porc_bonif/100))
INTO v_bonif_annos
FROM tramo_bonif_annos_trab
WHERE reg_datos_emp.annos_trab BETWEEN rango_ini AND rango_fin;
END IF;
-- salario a actualizar será el salario actual más bonificación por años
v_sal_act:=reg_datos_emp.salary+v_bonif_annos;
-- Calcula bonificación extra por la ciudad en que trabaja y se le suma al salario a actualizar
IF reg_datos_emp.ciudad ='Toronto' THEN
v_sal_act:= v_sal_act + ROUND(reg_datos_emp.salary*varray_porc_aum(1));
ELSIF reg_datos_emp.ciudad='London' THEN
v_sal_act:= v_sal_act + ROUND(reg_datos_emp.salary*varray_porc_aum(2));
END IF;
IF v_sal_act > 0 THEN -- Se actualiza si es que hubo algún cambio en el salario
UPDATE empleados
SET salary=v_sal_act
WHERE employee_id=reg_datos_emp.employee_id;
END IF;
END LOOP;
COMMIT;
END;
```

- Se explicó cómo usar los Atributo %ISOPEN, %NOTFOUND, %FOUND y %ROWCOUNT para obtener información del estado de un Cursor Explícito.
- Se explicó cómo procesar las filas de un Cursor Explícito usando la estructura de iteración LOOP Simple.
- Se explicó cómo procesar todas filas de un Cursor Explícito usando la estructura de iteración WHILE LOOP.
- Se explicó cómo procesar las filas de un Cursor Explícito usando la estructura de iteración FOR LOOP.