

Diseño de Bases de Datos Multimedia



Cursores. Uso en Oracle PL/SQL

Grado en Ingeniería Multimedia



Universitat d'Alacant
Universidad de Alicante



Departamento de
Lenguajes y Sistemas Informáticos



ÍNDICE

- ¿Qué es y para qué sirve un cursor?.
- Declaración de un cursor en un bloque PL/SQL.
- Uso de cursores en un bloque.
- Trabajando en un cursor con bucles FOR.
- Trabajando en un cursor con OPEN, FETCH y CLOSE.

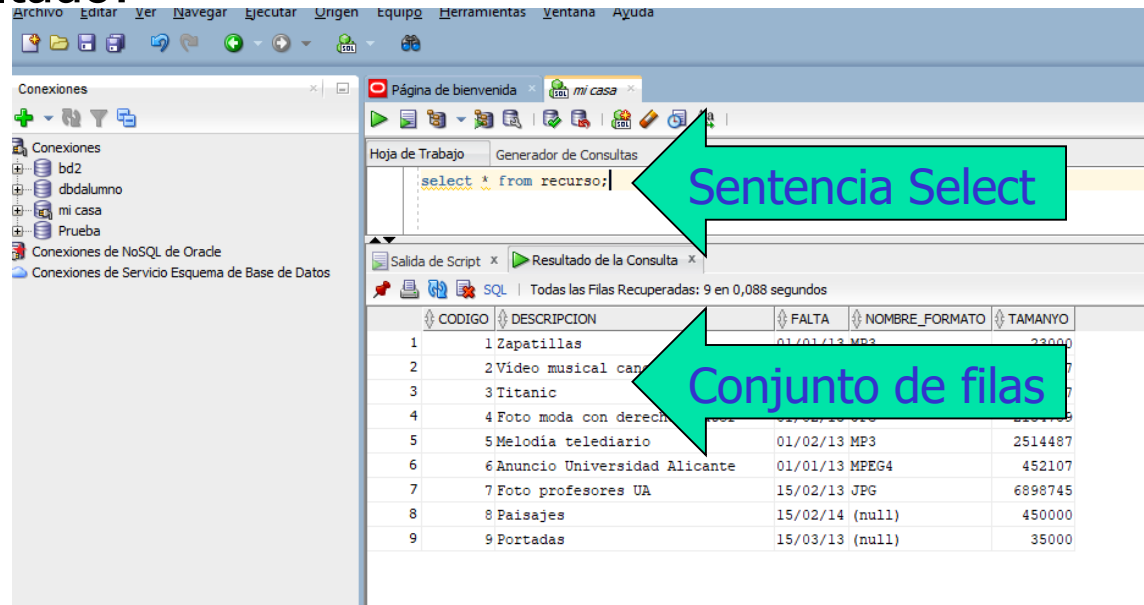


¿Qué es y para qué sirve un cursor?

- En bases de datos, el término “**cursor**” se refiere a una estructura de control que proporciona el gestor de la base de datos, y que utilizaremos para el recorrido de las filas que han resultado tras la ejecución de una consulta.
- Un cursor se utiliza para darnos la capacidad de procesamiento una a una de las filas devueltas por una sentencia SELECT de SQL.
- Este procesamiento fila a fila podremos realizarlo dentro de un bloque PL/SQL como los que vimos en la sesión anterior.

¿Qué es y para qué sirve un cursor?

- Si ejecutamos la sentencia `select SELECT * FROM RECURSO` sobre la base de datos de recursos multimedia, obtenemos este resultado:



Conexiones

- Conexiones
- bd2
- dbdalumno
- mi casa
- Prueba
- Conexiones de NoSQL de Oracle
- Conexiones de Servicio Esquema de Base de Datos

Hoja de Trabajo

Generador de Consultas

`select * from recurso;`

Salida de Script x

Resultado de la Consulta x

Todas las Filas Recuperadas: 9 en 0,088 segundos

	CODIGO	DESCRIPCION	FALTA	NOMBRE_FORMATO	TAMANYO
1		1 Zapatillas	01/01/13	MP3	23000
2		2 Video musical can			7
3		3 Titanic			7
4		4 Foto moda con derech			200000
5		5 Melodia telediario	01/02/13	MP3	2514487
6		6 Anuncio Universidad Alicante	01/01/13	MPEG4	452107
7		7 Foto profesores UA	15/02/13	JPG	6898745
8		8 Paisajes	15/02/14	(null)	450000
9		9 Portadas	15/03/13	(null)	35000

- Hasta ahora no podíamos manejar una a una las filas resultado de esta consulta. La definición de un cursor nos ayudará a ello.



Declarando cursores en un bloque PL/SQL

- Los cursores se declaran en la sección DECLARE de los bloques PL/SQL.
- Para ello, utilizaremos la siguiente sintaxis:

CURSOR nombre_cursor IS sentencia_select;

- Ejemplo:

DECLARE

**CURSOR Recursos_grandes IS
SELECTCodigo, Descripcion
FROM Recursos
WHERE Tamanyo >= 200000;**



Uso de cursores en un bloque PL/SQL

- El esqueleto de uso de un cursor en un bloque PL/SQL es como sigue:

DECLARE

- **Declararemos el cursor**

BEGIN

- **Abriremos el cursor.**
- **Cargaremos los datos de la fila activa en variables de memoria.**
- **Iteraremos con un bucle para cambiar la fila activa.**
- **Cerraremos el cursor.**

END;



Uso de cursores en un bloque PL/SQL

- Un cursor tiene asociados cuatro atributos, que podremos utilizar para poder controlar su estado y el resultado de las interacciones sobre él:

Atributo	Tipo de dato	Descripción
%ROWCOUNT	Numérico	Número de filas que hemos recuperado del cursor.
%FOUND	Booleano	Vale TRUE si hemos podido recuperar una fila del cursor.
%NOTFOUND	Booleano	Lo contrario de %FOUND.
%ISOPEN	Booleano	Vale TRUE si el cursor está abierto.

- Los atributos de un cursor se usan dentro del bloque PL/SQL anteponiendo el nombre del cursor al atributo, por ejemplo:

Recursos_grandes%ROWCOUNT



Trabajando con bucle FOR

- En PL/SQL podemos trabajar con cursores de dos maneras. La primera es utilizando un bucle FOR para recorrer el cursor. Normalmente no necesitaremos usar los atributos del cursor.

DECLARE

- **Declararemos el cursor y su variable tipo fila**

BEGIN

- **Iteraremos con el bucle FOR para cambiar la fila activa.**
- **Cargaremos en una variable de memoria tipo fila los valores del registro activo del cursor.**

END;



Trabajando con bucle FOR

- Ejemplo:

Desarrollemos un bloque PL/SQL que cree un cursor con los recursos más grandes (de más de 200.000 bytes de tamaño). Si el tamaño está entre 200.000 y 500.0000, modificaremos el tamaño poniendo su valor en Kbytes (dividiremos por 1024). Si su tamaño es superior a 500.000, lo modificaremos por su valor en Megabytes (dividiremos entre $1024*1024$).

Trabajando con bucle FOR

- Ejemplo: Desarrollemos un bloque PL/SQL que cree un cursor con los recursos más grandes (de más de 200.000 bytes de tamaño). Si el tamaño está entre 200.000 y 500.0000, modificaremos el tamaño su valor en Kbytes (dividiremos por 1024). Si su tamaño es superior a 500.000, lo modificaremos por su valor en Megabytes (dividiremos por $1024*1024$).

DECLARE

```
CURSOR Recursos_grandes IS
SELECT Codigo, Descripcion, Tamanyo
FROM Recurso
WHERE Tamanyo >= 200000;
Registro Recursos_grandes%ROWTYPE;
```

BEGIN

```
FOR Registro IN Recursos_grandes LOOP
  IF Registro.Tamanyo BETWEEN 200000 AND 500000 THEN
    UPDATE Recurso SET Tamanyo = Tamanyo/1024
    WHERE Codigo = Registro.Codigo;
  ELSIF Registro.Tamanyo > 500000 THEN
    UPDATE Recurso SET Tamanyo = Tamanyo/(1024*1024)
    WHERE Codigo = Registro.Codigo;
  END IF;
END LOOP;
END;
```

Declaramos el cursor y la variable de memoria que almacena cada fila

En cada iteración del FOR se activará una nueva fila en la variable "Registro"

Realizaremos acciones diferentes para Cada fila activa del cursor.

El bucle terminará cuando no haya más filas en el cursor (lleguemos al final)



Trabajando con OPEN FETCH y CLOSE

- La segunda manera es utilizando las sentencias OPEN, FETCH y CLOSE. Cada vez está más en desuso. Necesitaremos los atributos del cursor, al menos para saber si lo hemos recorrido completamente (%FOUND). Su estructura es:

DECLARE

- **Declararemos el cursor y la variable tipo fila.**

BEGIN

- **Abriremos el cursor usando la sentencia **OPEN**.**
- **Cargaremos los datos de la fila activa en variables de memoria utilizando la sentencia **FETCH**.**
- **Iteraremos con un bucle para cambiar la fila activa.**
- **Cerraremos el cursor utilizando la sentencia **CLOSE**.**

END;

Trabajando con OPEN FETCH y CLOSE

- Ejemplo: Desarrollemos un bloque PL/SQL que cree un cursor con los recursos más grandes (de más de 200.000 bytes de tamaño). Si el tamaño está entre 200.000 y 500.0000, modificaremos el tamaño su valor en Kbytes (dividiremos por 1024). Si su tamaño es superior a 500.000, lo modificaremos por su valor en Megabytes (dividiremos por $1024*1024$).

DECLARE

```
CURSOR Recursos_grandes IS SELECT Codigo, Descripcion, Tamanyo
FROM Recurso
WHERE Tamanyo >= 200000;
Registro Recursos_grandes%ROWTYPE;
```

BEGIN

```
OPEN Recursos_grandes;
FETCH Recursos_grandes INTO Registro;
WHILE Recursos_grandes%FOUND LOOP
    IF Registro.Tamanyo BETWEEN 200000 AND 500000 THEN
        UPDATE Recurso SET Tamanyo = Tamanyo/1024
        WHERE codigo = Registro.Codigo;
    ELSIF Registro.Tamanyo > 500000 THEN
        UPDATE Recurso SET Tamanyo = Tamanyo/(1024*1024)
        WHERE Codigo = Registro.Codigo;
    END IF;
    FETCH Recursos_grandes INTO Registro;
END LOOP;
CLOSE Recursos_grandes;
END;
```

Declaramos el cursor y la variable de memoria que almacena cada fila

Es necesario abrir el cursor, y FETCH almacena en memoria cada fila. El bucle Terminará cuando no haya más filas en el cursor (%FOUND = FALSE)

En cada iteración debemos cargar la variable de memoria con FETCH. Si no lo hacemos provocaremos un bucle infinito

Deberemos cerrar el cursor al salir del bucle.