

6

Cursos ORACLE

Bases de datos II
Universidad del Cauca
Ing. Wilson Ortega

Definición(1)

- El servidor Oracle usa área de trabajo llamadas Areas SQL privadas para ejecutar las sentencias SQL y almacenar la información procesada. Es posible usar cursores para nombrar un área privada y consultar su información.

Definición (2)

- Todas las sentencias SQL ejecutadas por el servidor de Oracle tienen un cursor individual asociado:
 - Cursores implícitos: Se declaran y se gestionan por PL/SQL para todas las sentencias SELECT DML y PL/SQL
 - Cursores explícitos: Se declaran y se gestionan por el programador

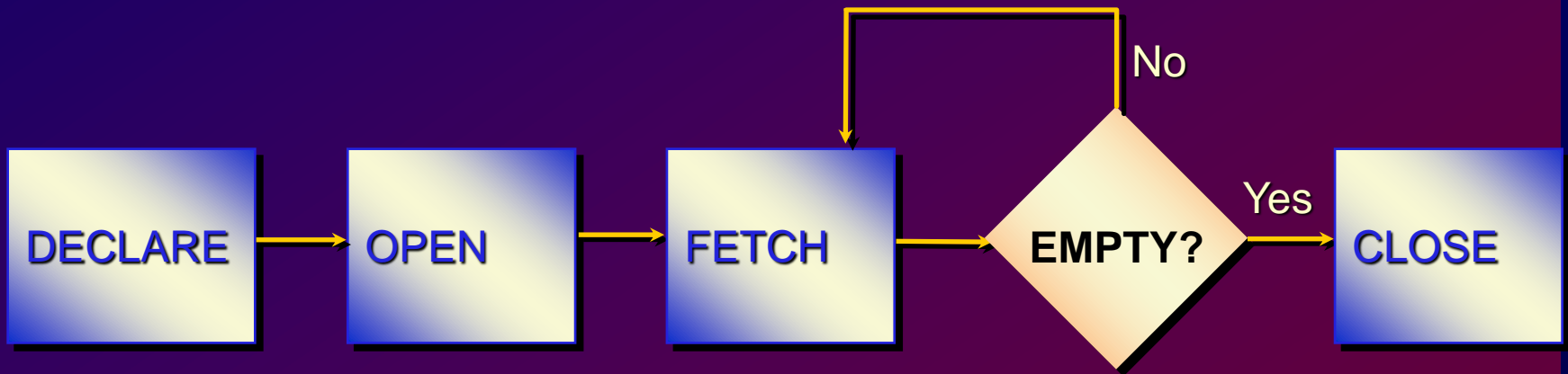
Cursores implícitos - Atributos

Retornan información de la ejecución de instrucciones INSERT, UPDATE, DELETE, SELECT INTO, COMMIT, or ROLLBACK

- %FOUND : La instrucción DML cambió filas?
- %NOTFOUND : La instrucción DML falló al cambiar filas?
- %ROWCOUNT: Cuántas filas se cambiaron?

```
BEGIN
  UPDATE ESTUDIANTE SET programa_id = 1 where programa_id IS NULL;
  DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' Estudiantes actualizados ');
END;
```

Control de cursores explícitos



- Declarar
- Abrir el cursor
- Recuperar datos del cursor, fila a fila
- Comprueba si hay registros
- Cierra el cursor
- Devuelve el control a FECH si encuentra registros

Control de cursores explícitos

Abrir cursor

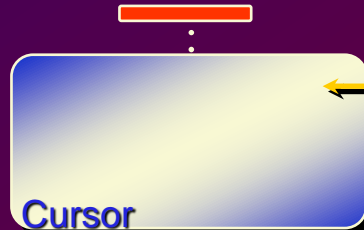


Puntero



Puntero

Continuar hasta que quede vacío



Puntero

Cerrar el cursor

Definición del cursor

Ej.

```
CURSOR nombre_cursor IS  
    select ...;
```

- Se deben definir en la sección de definición de variables y tipos del bloque
- Es posible usar variables en el where

Definición del cursor

Ejemplo

```
Ej.  
DECLARE  
CURSOR cur_estudiantes IS  
    select codigo, nombre from ESTUDIANTE  
    where programa_id = 5 ;  
  
CURSOR cur_programas IS  
    select nombre from PROGRAMA  
    order by nombre;  
BEGIN  
...
```


Recuperación de datos

```
...  
OPEN cursor_nombre;  
LOOP  
    FETCH cursor_nombre INTO variables  
    EXIT WHEN ...; -- ej. cursor vacío  
    ...  
    -- Procesar los datos  
    ...  
END LOOP;  
CLOSE cursor_nombre  
END;
```

```
FETCH cursor_nombre INTO v_est_codigo, v_est_nombre;
```

```
FETCH cursor_nombre INTO v_registro_est;
```

Usar el LOOP básico

Ejemplo

```
DECLARE
  CURSOR cur_est IS
    SELECT codigo, promedio
    FROM   ESTUDIANTE;
  v_reg_est cur_est%ROWTYPE;
BEGIN
  OPEN cur_est;
  LOOP
    FETCH cur_est INTO v_reg_est;
    EXIT WHEN cur_est%NOTFOUND;
    IF v_reg_est.promedio < 3 THEN
      dbms_output.put_line(v_reg_est.codigo || 'Reprobado' );
    END IF;
  END LOOP;
  CLOSE cur_est;
END;
/
```

Usar el FOR cursor

```
FOR registro_nombre IN cursor_nombre LOOP  
    sentencia1;  
    sentencia2;  
    . . .  
END LOOP;
```

- Facilita el procesamiento de cursores explícitos
- Apertura, recuperación y cierre implícitos
- No declarar la variable registro, se declara implícitamente

Usar el FOR cursor

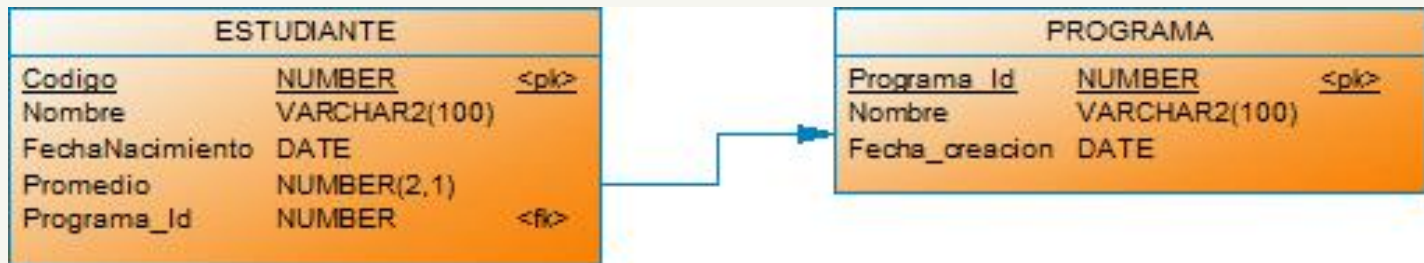
Ejemplo

```
DECLARE
  CURSOR cur_est IS
    SELECT codigo, promedio
    FROM   ESTUDIANTE;

BEGIN
  FOR reg_est IN cur_est LOOP
    IF reg_est.promedio < 3 THEN
      dbms_output.put_line(reg_est.codigo||' - Reprobado');
    END IF;
  END LOOP;

END;
```

Ejercicio



- Muestre el código y promedio de todos los estudiantes. Si el promedio es menor a 3 se imprime ('Reprobado') si es mayor o igual a 3 se imprime ('Aprobado')
 - Use un LOOP básico
 - Use un FOR Cursor

Atributos de cursores explícitos

- %FOUND: Se recuperó la siguiente fila?
- %ISOPEN: Está abierto?
- %NOTFOUND : Falló la recuperación de la siguiente fila?
- %ROWCOUNT : Cuantas filas se recuperaron?

Atributos de cursores explícitos

Ejemplo

```
SET SERVEROUTPUT ON;
DECLARE

CURSOR cur_est IS
SELECT codigo, nombre
FROM ESTUDIANTE;
reg_est cur_est%ROWTYPE;

BEGIN
  OPEN cur_est;
  LOOP
    FETCH cur_est INTO reg_est;
    EXIT WHEN cur_est%NOTFOUND OR cur_est%NOTFOUND IS NULL;
    DBMS_OUTPUT.PUT_LINE(cur_est%ROWCOUNT || ' - ' ||
      reg_est.nombre);
  END LOOP;
  CLOSE cur_est;
END;
```

```
1 - José
2 - Sara
3 - Pedro
```

Cursores con parámetros – Ejemplo(I)

```
SET SERVEROUTPUT ON;
DECLARE

CURSOR cur_est (p_id_pro PROGRAMA.programa_id%TYPE) IS
SELECT codigo, nombre
FROM ESTUDIANTE
WHERE programa_id = p_id_pro;
reg_est cur_est%ROWTYPE;

BEGIN
  OPEN cur_est(2);
  LOOP
    FETCH cur_est INTO reg_est;
    EXIT WHEN cur_est%NOTFOUND OR cur_est%NOTFOUND IS NULL;
    DBMS_OUTPUT.PUT_LINE(cur_est%ROWCOUNT || ' - ' ||
reg_est.nombre);
  END LOOP;
  CLOSE cur_est;
END;
```

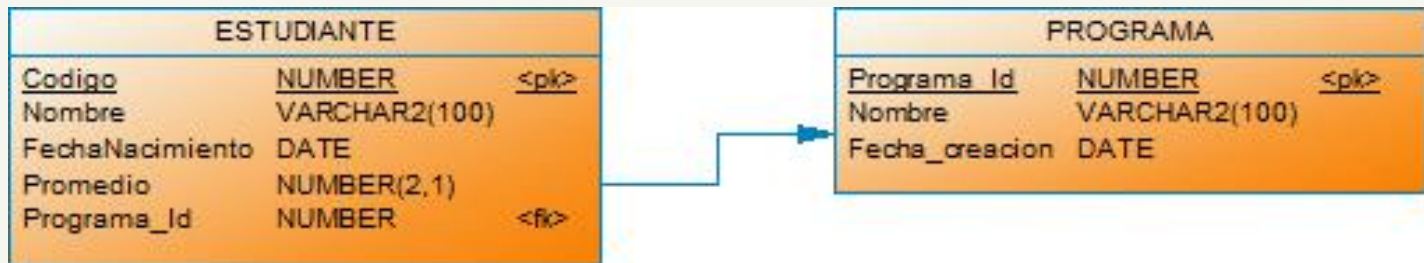

Cursores con parámetros – Ejemplo(II)

```
SET SERVEROUTPUT ON;
DECLARE

CURSOR cur_est (p_id_pro PROGRAMA.programa_id%TYPE) IS
SELECT codigo, nombre
FROM ESTUDIANTE
WHERE programa_id = p_id_pro;

BEGIN
  FOR reg_est IN cur_est(1) LOOP
    DBMS_OUTPUT.PUT_LINE(cur_est%ROWCOUNT || ' - ' || reg_est.nombre);
  END LOOP;
END;
```

Ejercicio



- Cree un procedimiento almacenado que recibe como parámetro el id de un programa y mediante un cursor parametrizado muestra la lista de estudiantes:
 - Código, Nombre, (aprobado / reprobado)

Bibliografía

- **Oracle® Database PL/SQL Language Reference -11g Release 1 (11.1) - 2009**