

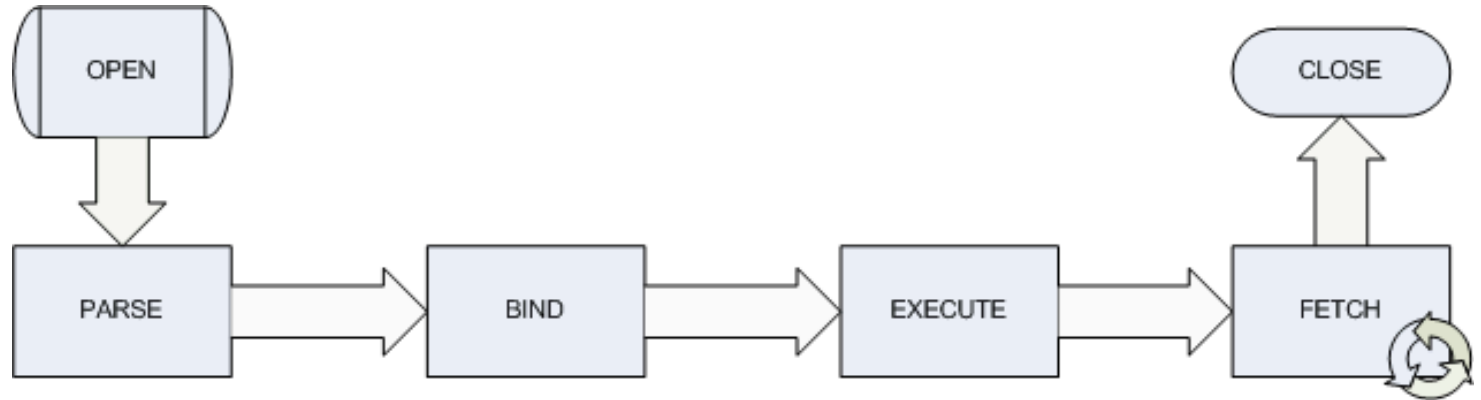


TECNICAS DE ALMACENAMIENTO DE DATOS

CURSORES

- Los cursores permiten trabajar con consultas SELECT que devuelven varias filas
- admiten el uso de parámetros
- incluir la cláusula ORDER BY si es necesario procesar filas en orden

Control de Cursores



Control de Cursores

- **Open**
Se asigna una estructura de memoria al cursor.
- **Parse**
Se asocia la sentencia SQL con el cursor.
- **Bind**
Se asocian los valores a las variables. No se realiza verificación durante el enlace. Si se pasan datos no válidos, se mostrará un error durante la ejecución.
- **Execute**
La sentencia SQL se ejecuta.
- **Fetch**
Si la sentencia devuelve los datos, es en esta etapa donde se obtienen.
- **Close**
Los recursos asociados con el cursor se liberan.

Control de Cursores

Pasos para trabajar con un cursor:

- Declarar el cursor
- Abrir el cursor con la instrucción **OPEN**
- Leer los datos del cursor con la instrucción **FETCH**
- Cerrar el cursor y liberar los recursos con la instrucción **CLOSE**

Declaración de Cursores

--CURSOR SIN PARAMETROS

```
DECLARE
  CURSOR NOMBRE_CURSOR
  IS
    SELECT CAMPO1 [ ,CAMPO2 ,CAMPO3 ,...]
      FROM TABLA;
BEGIN
  /* SENTENCIAS ...*/
END;
```

--CURSOR CON PARAMETROS

```
DECLARE
  CURSOR NOMBRE_CURSOR (PARAM NUMBER)
  IS
    SELECT CAMPO1 [ ,CAMPO2 ,CAMPO3 ,...]
      FROM TABLA
      WHERE CAMPO_X = PARAM ;
BEGIN
  /* SENTENCIAS ...*/
END;
```

Apertura de Cursores

- Si la consulta no devuelve ninguna fila, no se producirá ninguna excepción al abrir el cursor

--CURSOR SIN PARAMETROS

```
OPEN nombre_cursor;
```

--CURSOR CON PARAMETROS

```
OPEN nombre_cursor(valor1, valor2);
```

Leer datos del cursor

- Incluir en el FETCH el mismo número de variables que las definidas en el cursor y en el mismo orden

--FETCH INTO VARIABLES

```
FETCH nombre_cursor INTO var1[,var2,...) ;
```

--FETCH INTO REGISTRO

```
FETCH nombre_cursor INTO registro;
```


Cierre de Cursores

- Al finalizar el procesamiento de las filas se debe cerrar el cursor
- No se pueden recuperar datos de un cursor una vez que ha sido cerrado (exception INVALID_CURSOR)

--CURSOR CON/SIN PARAMETROS

`CLOSE nombre_cursor;`

Variables para Cursores

DECLARE

```
CURSOR NOMBRE_CURSOR  
IS  
SELECT CAMPO1 , CAMPO2  
FROM TABLA;
```

```
VARIABLE1 TABLA.CAMPO1%TYPE;  
VARIABLE2 TABLA.CAMPO2%TYPE;
```

BEGIN

```
OPEN NOMBRE_CURSOR;  
FETCH NOMBRE_CURSOR INTO VARIABLE1 , VARIABLE2;  
CLOSE NOMBRE_CURSOR;
```

END;

Variables para Cursores

DECLARE

CURSOR **NOMBRE_CURSOR**

IS

SELECT *

FROM TABLA;

REGISTRO TABLA%ROWTYPE;

BEGIN

OPEN **NOMBRE_CURSOR**;

FETCH **NOMBRE_CURSOR** INTO **REGISTRO**;

DBMS_OUTPUT.PUT_LINE (**REGISTRO.CAMPO1**) ;

DBMS_OUTPUT.PUT_LINE (**REGISTRO.CAMPO2**) ;

CLOSE **NOMBRE_CURSOR**;

END ;

Utilización de Cursores

- No se puede leer un cursor que está cerrado
- El motor arrojará un error al intentar cerrar un cursor ya cerrado o que no haya sido abierto
- Al recuperar datos de un cursor se recomienda comprobar el resultado de lectura utilizando los **atributos** de los cursores

Atributos de Cursores

Toman los valores TRUE, FALSE o NULL dependiendo de la situación

Atributo	Tipo	Descripción
%ISOPEN	Booleano	TRUE si el cursor está abierto.
%NOTFOUND	Booleano	TRUE si la recuperación más reciente no devuelve ninguna fila.
%FOUND	Booleano	TRUE si la recuperación más reciente devuelve una fila.
%ROWCOUNT	Númerico	Proporciona el número total de filas devueltas hasta ese momento.

Atributos de Cursores

--VERIFICACION DE CURSOR ABIERTO

```
IF NOT NOMBRE_CURSOR%ISOPEN THEN  
    OPEN NOMBRE_CURSOR;  
END IF;
```

--VERIFICACION DE CURSOR CON DATOS

```
LOOP  
    FETCH NOMBRE_CURSOR INTO REGISTRO;  
    EXIT WHEN NOMBRE_CURSOR%NOTFOUND;  
  
    DBMS_OUTPUT.PUT_LINE (REGISTRO.CAMPO) ;  
END LOOP;
```

--VERIFICACION ANTES DE CERRAR CURSOR

```
IF NOMBRE_CURSOR%ISOPEN THEN  
    CLOSE NOMBRE_CURSOR;  
END IF;
```

Manejo de Cursores

Podemos iterar a través de los registros del cursor mediante LOOP

```
--LOOP + EXIT
OPEN nombre_cursor;
LOOP
    FETCH nombre_cursor INTO registro;
    EXIT WHEN nombre_cursor%NOTFOUND;
    /* Procesamiento de los registros recuperados */
END LOOP;
CLOSE nombre_cursor;
```

Manejo de Cursores

Podemos iterar a través de los registros del cursor mediante LOOP y WHILE. La instrucción FETCH aparece dos veces

```
--LOOP + WHILE
```

```
OPEN nombre_cursor;
```

```
FETCH nombre_cursor INTO registro;
```

```
WHILE nombre_cursor%FOUND;
```

```
LOOP
```

```
    /* Procesamiento de los registros recuperados */
```

```
    FETCH nombre_cursor INTO registro;
```

```
END LOOP;
```

```
CLOSE nombre_cursor;
```


EJERCICIO

Utilizando cursores mostrar en pantalla los nombres de los empleados que son programadores (JOB_ID = 'IT_PROG').
Mostrar únicamente 3 registros.

EJEMPLO DE SALIDA:

Programador #1: Hunold

Programador #2: Ernst

Programador #3: Austin

EJERCICIO

DECLARE

CURSOR CUR_PROG

IS

SELECT * FROM EMPLOYEES

WHERE JOB_ID = 'IT_PROG';

REG_PROG EMPLOYEES%ROWTYPE;

BEGIN

OPEN CUR_PROG;

LOOP

FETCH CUR_PROG INTO REG_PROG;

EXIT WHEN CUR_PROG%NOTFOUND OR CUR_PROG%ROWCOUNT > 3;

DBMS_OUTPUT.PUT_LINE('Programador #' || CUR_PROG%ROWCOUNT ||
' : ' || REG_PROG.LAST_NAME);

END LOOP;

CLOSE CUR_PROG;

END;