

# Gestión de bases de datos

Cursores

# Introducción

PL/SQL utiliza cursores para gestionar las instrucciones SELECT. Un cursor es un conjunto de registros devuelto por una instrucción SQL.

Técnicamente, los cursores son fragmentos de memoria reservados para procesar los resultados de una consulta SELECT. Podemos distinguir **dos tipos de cursores**:

- ❑ **Cursores implícitos.** Este tipo de cursores se utiliza para operaciones SELECT INTO. Se usan cuando la consulta devuelve **un único registro**.
- ❑ **Cursores explícitos.** Son los cursores que son declarados y controlados por el programador. Se utilizan cuando la consulta devuelve un conjunto de registros.

# Cursores implícitos

Los cursores implícitos se utilizan para realizar consultas SELECT que devuelven un único registro. Importante:

- ❑ Con cada cursor implícito debe existir la palabra clave INTO.
- ❑ Las variables que reciben los datos devueltos por el cursor tienen que contener el mismo tipo de dato que las columnas de la tabla.
- ❑ Los cursores implícitos solo pueden devolver una única fila. En caso de que se devuelva más de una fila (o ninguna fila) se producirá una excepción.

# Cursores implícitos. Ejemplo.

```
DECLARE
  V_NOMBRE VARCHAR2(50);
BEGIN
  SELECT NOMBRE INTO V_NOMBRE
  FROM ESTUDIANTES
  WHERE CODIGO = 1;
  DBMS_OUTPUT.PUT_LINE( 'EL NOMBRE DEL
ESTUDIANTE ES: ' || V_NOMBRE );
END;
/
```

# Cursores implícitos. Ejemplo.

```
DECLARE
    V_NOMBRE ESTUDIANTES%ROWTYPE;
BEGIN
    SELECT * INTO V_NOMBRE
    FROM ESTUDIANTES
    WHERE CODIGO = 1;
    DBMS_OUTPUT.PUT_LINE( 'EL NOMBRE DEL
ESTUDIANTE ES: ' || V_NOMBRE.NOMBRE );
END;
/
```

# Cursores implícitos. Excepciones.

Los cursores implícitos sólo pueden devolver una fila, por lo que pueden producirse determinadas excepciones. Las más comunes que se pueden encontrar son:

- ❑ **NO\_DATA\_FOUND:** Se produce cuando una sentencia `SELECT` intenta recuperar datos pero ninguna fila satisface sus condiciones. Es decir, cuando “no hay datos”.
- ❑ **TOO\_MANY\_ROWS:** Dado que cada cursor implícito sólo es capaz de recuperar una fila, esta excepción detecta la existencia de más de una fila.

# Cursores implícitos. Excepciones.

La sintaxis de la excepción es la siguiente:

```
EXCEPTION
    WHEN <caso_provoca_excepción> THEN
        <acciones_a_realizar>
```

Hay que recordar que EXCEPTION debe ir después de BEGIN y antes de terminar con END;

```
[DECLARE]
...
BEGIN
...
EXCEPTION
...
END;
/
```

# Cursores implícitos. Excepciones.

## Ejercicio 1.

1. Definir un cursor implícito para obtener por pantalla el nombre y los apellidos del estudiante con DNI='00000000T' de la tabla ESTUDIANTES (fichero estudiantes.sql).

**NOTA:** tener en cuenta que el DNI puede no existir en la tabla.

2. Prueba ahora con el DNI 97898989T.



# Cursores implícitos. Excepciones.

## Ejercicio 2.

1. Definir un cursor implícito para obtener por pantalla el dni y los apellidos del estudiante cuyo nombre sea Marta en la tabla ESTUDIANTES (fichero estudiantes.sql).

**NOTA:** tener en cuenta tanto que puede no existir ningún estudiante que se llame así o que pueda haber varios con el mismo nombre.

2. Prueba ahora con el nombre Blanca.

# Cursores explícitos.

Los cursores explícitos se utilizan para realizar consultas SELECT que devuelven varias filas. Se necesita:

- ☐ Declarar el cursor:

```
CURSOR nombre_cursor IS  
    SELECT...
```

- ☐ Declarar la variable tipo cursor donde se guardará el valor de la fila.

- ☐ Abrir el cursor con la instrucción OPEN.

- ☐ Leer los datos del cursor con la instrucción FETCH.

- ☐ Cerrar el cursor con la instrucción CLOSE.

# Cursores explícitos. Ejemplo.

```
DECLARE
    CURSOR C_NOMBRES IS
        SELECT NOMBRE
        FROM ESTUDIANTES;
    V_NOMBRE ESTUDIANTES.NOMBRE%TYPE;
BEGIN
    OPEN C_NOMBRES;
    FETCH C_NOMBRES INTO V_NOMBRE;
    CLOSE C_NOMBRES;
END;
/
```

## Cursores. Atributos.

- ❑ **%NOTFOUND**: devuelve true si fetch no devolvió una fila. Devuelve false si fetch devolvió una fila.
- ❑ **%FOUND**: devuelve true si fetch devolvió una fila. Devuelve false si fetch no devolvió una fila.
- ❑ **%ISOPEN**: devuelve true si el cursor está abierto. Devuelve false en caso contrario.
- ❑ **%ROWCOUNT**: devuelve el número de filas que se han recorrido en el cursor (cuántos fetch se han aplicado sobre el cursor).

# Cursores explícitos.

## Ejercicio 3.

Definir un cursor explícito que seleccione el nombre, apellidos y DNI de la tabla ESTUDIANTES. Abrir el cursor, extraer el primer dato, mostrar cuántos registros se han procesado (uno) y cerrarlo. Luego mostrar el nombre del estudiante del registro procesado en el cursor.

¿Qué sucede si ponemos dos FETCH?

# Cursores. Recorrerlo con LOOP.

Por medio de LOOP vamos a iterar a través del cursor. Teniendo cuidado de tener una condición para salir del bucle.

```
OPEN nombre_cursor;
```

```
LOOP
```

```
    FETCH nombre_cursor INTO lista_variables;
```

```
    EXIT WHEN nombre_cursor%NOTFOUND;
```

```
END LOOP;
```

```
CLOSE nombre_cursor;
```

# Cursores.

## Ejercicio 4.

Definir un cursor explícito que seleccione el nombre, apellidos y DNI de la tabla ESTUDIANTES. Recorrerlo y mostrar todos los datos recuperados.

Escribir al final el número de filas recuperadas en total.

# Cursores. Recorrerlo con WHILE.

Podemos usar WHILE LOOP para iterar un cursor. Cuidado que la instrucción FETCH aparece dos veces.

```
OPEN nombre_cursor;  
FETCH nombre_cursor INTO lista_variables;  
WHILE nombre_cursor%FOUND LOOP  
    FETCH nombre_cursor INTO lista_variables;  
END LOOP;  
CLOSE nombre_cursor;
```



# Cursores.

## Ejercicio 5.

Definir un cursor explícito que seleccione el nombre y la fecha de nacimiento de la tabla ESTUDIANTES. Recorrerlo y mostrar todos los datos recuperados con WHILE LOOP.

Escribir al final el número de filas recuperadas en total.

# Cursores. Recorrerlo con FOR.

Podemos usar FOR LOOP para iterar un cursor. Es la forma más corta ya que **el cursor implícitamente ejecuta las instrucciones OPEN, FETCH y CLOSE;**

```
FOR variable IN nombre_cursor LOOP  
    /* PROCESAMIENTO DE LOS REGISTROS */  
END LOOP;
```

**IMPORTANTE:** el atributo %NOTFOUND no se aplicará dentro de un FOR puesto que si no hay registros, se saldrá del FOR si evaluar nada dentro.

# Cursores.

## Ejercicio 6.

Definir un cursor explícito que seleccione el nombre, apellidos y fecha de nacimiento de la tabla ESTUDIANTES. Recorrerlo y mostrar todos los datos recuperados con un bucle FOR.

Escribir el número de filas recuperadas en total.

## Cursores.

### Ejercicio 7.

Se quiere mostrar por pantalla el nombre y apellidos de los estudiantes que se llamen de cierta forma. Para ello se pedirá al usuario que introduzca el nombre a buscar.

El formato requerido para mostrar por pantalla es el siguiente (todo en mayúsculas): “APELLIDOS, NOMBRE”.

Ej.: CARRASCO PEREZ, MARTA.

En el supuesto de que SELECT no se traiga ningún registro, mostrar por pantalla “NO HAY DATOS”. Utiliza un bucle WHILE.

Lanza la ejecución para el nombre “Marta” y después para el nombre “Luis”.

# **Cursores.**

## **Ejercicio 8.**

Realiza el mismo ejercicio anterior pero con un bucle FOR.



# CEU

*Centro de Estudios  
Profesionales*

---

*Fundación San Pablo Andalucía*