



BASE DE DATOS AVANZADO I

Unidad 3: Introducción a la programación Transact-SQL y Cursores

Tema 7: Construcción de Cursores





Tema 07: Construcción de Cursores

3.2. Tema 7: Construcción de Cursores

3.2.1. Construcción de cursores explícitos e implícitos

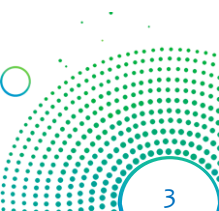
3.2.2. Cursores y actualización de datos.





Capacidades

1. Identifica las sentencias y estructuras de programación del lenguaje Transact-SQL
2. Diseña e implementa bloques de programas para optimizar las operaciones con la base de datos.
3. Implementa cursores





1. Cursores

- Un cursor es una variable que nos permite recorrer con un conjunto de resultados obtenidos a través de una sentencia SELECT fila por fila.
- El uso de los cursores es una técnica que permite tratar fila por fila el resultado de una consulta, contrariamente al SELECT SQL que trata a un conjunto de fila.
- Los cursores pueden ser implementados por instrucciones TRANSACT-SQL (cursores ANSI-SQL) o por la API OLE-DB..





1. Cursores

¿Cómo se crea un CURSOR?

- Cuando trabajemos con cursores debemos seguir los siguientes pasos.
 - Declarar el cursor, utilizando **DECLARE**
 - Abrir el cursor, utilizando **OPEN**
 - Leer los datos del cursor, utilice **FETCH ... INTO**
 - Cerrar el cursor, utilizando **CLOSE**
 - Liberar el cursor, utilizando **DEALLOCATE**





1. Cursores

¿Cómo se crea un CURSOR?

- Sintaxis:

- Declaración del cursor

- DECLARE <nombre_cursor> CURSOR

- FOR <sentencia_sql>

- Apertura del cursor

- OPEN <nombre_cursor>

- Lectura de la primera fila del cursor

- FETCH <nombre_cursor> INTO <lista_variables>

- WHILE (@@FETCH_STATUS = 0)

- BEGIN

- Lectura de la siguiente fila de un cursor

- FETCH <nombre_cursor> INTO <lista_variables>

- ...

- END -- Fin del bucle WHILE

- Cierra el cursor

- CLOSE <nombre_cursor>

- Libera los recursos del cursor

- DEALLOCATE <nombre_cursor>





1. Cursores

Declarar un CURSOR

- Sintaxis para declarar un cursor

-- Declaración del cursor

```
DECLARE <nombre_cursor> [INSENSITIVE][SCROLL] CURSOR  
FOR <sentencia_sql>  
FOR [READ ONLY | UPDATE[ OF lista de columnas]]
```





1. Cursores

Abrir un Cursor: OPEN

- Esta instrucción permite hacer operativo el cursor y crear eventualmente las tablas temporales asociadas. La variable @@CURSOR_ROWS se asigna después de OPEN.
- Sintaxis para abrir un cursor:

```
-- Abrir cursor  
OPEN [ GLOBAL ] <nombre_cursor>
```




1. Cursores

Leer un Registro: FETCH

- Es la instrucción que permite extraer una fila del cursor y asignar valores a variables con su contenido. Tras FETCH, la variable @@FETCH_STATUS está a 0 si FETCH no retorna errores.
- Sintaxis para leer un cursor

-- Leer una fila del cursor

```
FETCH [NEXT | PRIOR | LAST | ABSOLUTE n | RELATIVE n ]  
[FROM] [GLOBAL ] <nombre del cursor> [INTO lista de  
variables ]  
NEXT
```





1. Cursores

- Leer un Registro: *FETCH*
- Cuando trabajamos con cursores, la función @@FETCH_STATUS nos indica el estado de la última instrucción *FETCH* emitida, los valores posibles son:

Valor devuelto	Descripción
0	La instrucción <i>FETCH</i> se ejecutó correctamente
-1	La instrucción <i>FETCH</i> no se ejecutó correctamente o la fila estaba más allá del conjunto de resultados.
-2	Falta la fila recuperada.



2. Manejo de cursores

Cerrar un Cursor: CLOSE

- Cierra el cursor y libera la memoria. Esta operación debe interponerse lo antes posible con el fin de liberar los recursos cuanto antes.
- Sintaxis para cerrar un cursor:

```
-- Cerrar cursor  
CLOSE <nombre_cursor>
```





2. Manejo de cursores

Liberar el Cursor: DEALLOCATE

- Es un comando de limpieza, no forma parte de la especificación ANSI.
- Sintaxis para liberar un cursor_

-- Liberar cursor

DEALLOCATE <nombre_cursor>





2. Manejo de cursores

- Ejemplo
- Trabajando con un cursor dinámico, defina un cursor dinámico que permita visualizar: el primer registro, el registro en la posición 6 y el último registro de la Producto

```
/*Cursores*/  
--Declarar cursor  
DECLARE mi_cursor CURSOR SCROLL FOR  
    SELECT * FROM Compras.productos  
--Abrir cursor  
OPEN mi_cursor  
--Navegar los registros solicitados  
FETCH FIRST FROM mi_cursor  
FETCH ABSOLUTE 6 FROM mi_cursor  
FETCH LAST FROM mi_cursor  
--Cerrar el cursor  
CLOSE mi_cursor  
--Liberar el cursor  
DEALLOCATE mi_cursor  
go
```

Resultados							
	IdProducto	NomProducto	IdProveedor	IdCategoria	CantxUnidad	PrecioUnidad	UnidadesEnExiste
1	1	Te Dharamsala	1	1	10 cajas x 20 bolsas	18	39
	IdProducto	NomProducto	IdProveedor	IdCategoria	CantxUnidad	PrecioUnidad	Un
1	6	Memelada de grosellas de la abuela	3	2	12 - frascos 8 l	25	12
	IdProducto	NomProducto	IdProveedor	IdCategoria	CantxUnidad	PrecioUnidad	Unidades
1	77	Salsa verde original Frankfurter	12	2	12 cajas	13	32



2. Manejo de cursores

Ejemplo

- Trabajando con un cursor, listar los clientes registrados en la base de datos, incluya el nombre del país.

```
--declaro variables de trabajo
DECLARE @id varchar(5), @nombre varchar(50),
        @pais varchar(50)
--declaro el cursor
DECLARE mi_cursor CURSOR FOR
SELECT  c.idcliente, c.NomCliente, p.nombrepais
FROM Ventas.clientes c JOIN Ventas.países p
ON c.idpais=p.idpais
--abrir
OPEN mi_cursor
--leer el primer registro
FETCH mi_cursor INTO @id, @nombre, @pais
--mientras pueda leer el registro
WHILE @@FETCH_STATUS=0
BEGIN
    --imprimir el registro
    PRINT @id + ',' + @nombre + ',' + @pais
    PRINT REPLICATE('.',50)
    --leer el registro siguiente
    FETCH mi_cursor INTO @id, @nombre, @pais
END
--cerrar
CLOSE mi_cursor
--liberar
DEALLOCATE mi_cursor;
GO
```

Mensajes

ALFKI,Alfreds Futterkiste,Argentina

.....
ANATR,Ana Trujillo Emparedados y helados,España

.....
ANTON,Antonio Moreno Taqueria,Colombia

.....
AROUT,Around the Horn,USA

.....



2. Manejo de cursores

Ejemplo

- Cursor anidado. Muestre el nombre y dirección del cliente, por cada uno mostrar Id de Pedido y fecha de pedido generados.



2. Manejo de cursores

Ejemplo

```
/*Cursor de consulta Anidados*/
Declare Cursor2 Cursor For Select Idcliente, NomCliente, DirCliente from Ventas.clientes
Declare @vc_Idcli char(5), @vc_NomCli Varchar(50), @vc_DirCli varchar(50)
Open Cursor2
Fetch Cursor2 Into @vc_Idcli, @vc_Nomcli, @vc_Dircli
Print Space(25)+'REPORTE DE CLIENTES'
Print Space(25)+Replicate('*',25)
While @@FETCH_STATUS=0
Begin
    Print 'Nombre del Cliente: '+@vc_NomCli
    Print 'Dirección Cliente    ; '+@vc_DirCli
    Print ''
    Declare CursorAni1 Cursor For Select IdPedido, FechaPedido
    From Ventas.pedidoscabe Where IdCliente=@vc_Idcli
    Declare @vc_IdPed int, @vc_FecPed Date
    Open CursorAni1
    Fetch CursorAni1 Into @vc_IdPed, @vc_FecPed
    While @@FETCH_STATUS =0
    Begin
        Print space(20) + ltrim(@vc_IdPed) + space(20) + convert(varchar(16),@vc_FecPed,6)
        Print space(20) + '-----'
        Fetch CursorAni1 Into @vc_IdPed, @vc_FecPed
    End
    Close CursorAni1
    Deallocate CursorAni1
    Fetch Cursor2 Into @vc_Idcli, @vc_Nomcli, @vc_Dircli
End
Print Replicate('*',50)
Close Cursor2
Deallocate Cursor2
go
```


2. Manejo de cursores

Ejemplo

Messages	
REPORTE DE CLIENTES	

Nombre del Cliente: Alfreds Futterkiste	
Dirección Cliente :Obere Str. 57	
10643	25 Ago 97

10692	03 Oct 97

10702	13 Oct 97

10835	15 Ene 98

10952	16 Mar 98



Ejercicio

- En la empresa Recobra, se pide implementar un cursor que permita insertar el código, nombre de cliente, monto de la deuda y mora de los clientes con factura pendiente de pago (estado de factura 2) en la tabla Tb_Morosos (qué debe crearse previamente), considerar los siguientes criterios:
 - Si es menos a 200 soles será del 10% de la deuda.
 - Si es mayor igual a 200 y menos a 1000, será del 15% de la deuda.
 - Si es mayor o igual a 1000 será del 20% de la deuda.



Ejercicio: Solución

```
-- Paso 1: Crear la tabla
Create Table l'B_Moroso
(
  codCli   ctlar(5),
  nomCli   varehlar{38),
  monto    sm allmoney,
  Mora     sm allmoney
)
go
....
```

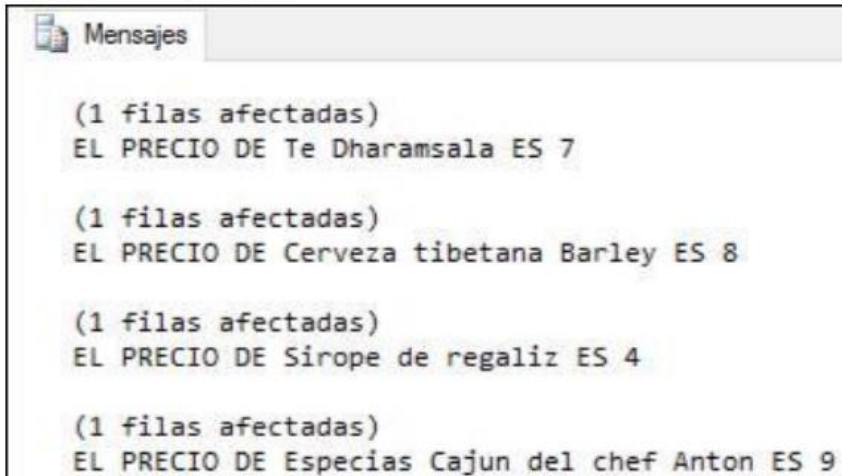
Ejercicio: Solución

En el ejemplo siguiente, actualizamos el precio de los productos: si su stock es mayor o igual a 1000, se descuenta el precio al 50%, sino se descuenta el precio al 20%.

```
-- DECLARACION DE VARIABLES PARA EL CURSOR
DECLARE @ID INT, @NOMBRE VARCHAR (255), @PRECIO DECIMAL, @ST INT
-- DECLARACIÓN DEL CURSOR DE ACTUALIZACION
DECLARE CPRODUCTO CURSOR FOR
    SELECT IDPRODUCTO, NOMPRODUCTO, PRECIOUNIDAD,
           UNIDADESENEXISTENCIA
    FROM COMPRAS.PRODUCTOS
    FOR UPDATE
-- APERTURA DEL CURSOR
```

```
OPEN CPRODUCTO
-- LECTURA DE LA PRIMERA FILA DEL CURSOR
FETCH CPRODUCTO INTO @ID, @NOMBRE, @PRECIO, @ST
-- MIENTRAS PUEDA LEER EL REGISTRO
WHILE (@@FETCH_STATUS = 0)
BEGIN
    IF (@ST >= 1000)
        SET @PRECIO = 0.5 * @PRECIO
    ELSE
        SET @PRECIO = 0.80 * @PRECIO
    UPDATE COMPRAS.PRODUCTOS SET
    PRECIOUNIDAD = @PRECIO WHERE
    CURRENT OF CPRODUCTO --
    IMPRIMIR
    PRINT 'EL PRECIO DE ' + @NOMBRE + ' ES ' + LTRIM(STR(@PRECIO))
    -- LECTURA DE LA SIGUIENTE FILA DEL CURSOR
    FETCH CPRODUCTO INTO @ID, @NOMBRE, @PRECIO, @ST
END
-- CIERRE DEL CURSOR
CLOSE CPRODUCTO
-- LIBERAR LOS RECURSOS
DEALLOCATE CPRODUCTO
```

El resultado sería:



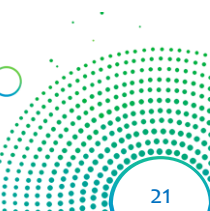
Mensajes

(1 filas afectadas)
EL PRECIO DE Te Dharamsala ES 7
(1 filas afectadas)
EL PRECIO DE Cerveza tibetana Barley ES 8
(1 filas afectadas)
EL PRECIO DE Sirope de regaliz ES 4
(1 filas afectadas)
EL PRECIO DE Especias Cajun del chef Anton ES 9



Conclusiones

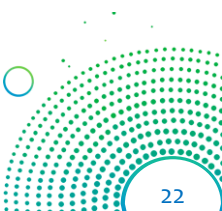
- Un cursor es una estructura utilizada para el recorrido (y potencial procesamiento) de los registros que es el resultado de una consulta..
- Un cursor se utiliza para el procesamiento individual de las filas devueltas por el sistema gestor de base de datos para una consulta.
- Un cursor se Declara, se lee, se recorre, se cierra y se libera de memoria.
- @@FetchStatus es una variable global que devuelve un valor que representa el estado del cursor. El valor correcto es 0.





Bibliografía

Microsoft (2017) DECLARE CURSOR (Transact-SQL) Recuperado de: <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/declare-cursor-transact-sql?view=sql-server-2017>



GRACIAS

**SEDE LIMA CENTRO**

Av. Uruguay 514
Cercado – Lima
Teléfono: 419-2900

SEDE INDEPENDENCIA

Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA

Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO

Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO

Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE BELLAVISTA

Av. Mariscal Oscar R. Benavides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA

Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535