



**Universidad Nacional del Litoral**

*Facultad de Ingeniería y Ciencias Hídricas*

Departamento de Informática

## ***Bases de Datos***

Guía de Trabajo Nro. 5

*T-SQL: Cursores*

Msc. Lic. Hugo Minni  
2008

SQL fue diseñado como un lenguaje orientado a conjuntos. Los cursores permiten tomar un conjunto de datos y tratarlos no como un conjunto de datos, sino fila-a-fila.

Un cursor es una construcción de programación que permite que cada fila de una tabla sea manipulada separadamente de las demás filas.

Esto puede ser útil cuando las filas en una tabla no pueden ser manipuladas fácilmente con comandos orientados a conjuntos.

Un beneficio de usar un cursor es que se pueden realizar múltiples operaciones sobre una única fila antes de moverse a otra.

El cursor se puede usar para leer o modificar filas.

## Creación de un cursor

### 1) Declarar el cursor

La declaración debe ocurrir en su propio batch.

Asigna memoria y estructuras de datos en el servidor SQL.

#### Sintaxis

```
Declare <Nombre-Cursor>
    [insensitive | scroll ] cursor
    for <Sentencia Select>
    [for <read only | update [of <Lista-de-columnas]> ]
```

## Tipos de cursores declarables

```
Declare curPub
  cursor
  for
    Select price
    From titles
```

### **Para actualización**

(Todas las columnas pueden ser modificadas)

```
Declare curPub
  cursor
  for
    Select price
    From titles
  for update
```

### **Para actualización**

(Solo la columna explicitada puede ser modificada)

```
Declare curPub
  cursor
  for
    Select price
    From titles
  For update OF price
```

### **Solo lectura**

```
Declare curPub
  cursor
  for
    Select price
    From titles
  for Read Only
```

### **Permiten navegar en ambas direcciones**

```
Declare curPub
  cursor
  Scroll
  for
    Select price
    From titles
```

## 2) Abrir el cursor

En este punto, el RDBMS lleva a cabo la sentencia `Select` pero no retorna datos aún.

```
Open Cursor curPublicaciones
```

Este comando no retorna resultado, pero impacta en la variable global @@CURSOR\_ROWS:

### La variable global @@CURSOR\_ROWS

Contiene la cantidad de filas en el data set.

Reporta cuantas filas cualifican para el último cursor abierto.

#### Valores posibles

Valor	Significado
Positivo	SQL Server ya sabe exactamente cuantas filas hay en el conjunto resultado.
Cero	No hay filas en el conjunto resultado o el último cursor abierto ya ha sido cerrado o destruido.
Negativo	Si el conjunto resultado está siendo cargado asincrónicamente, este valor es negativo si SQL Server aún no terminó el proceso de carga.

## 3) Fetch de filas y procesamiento

El comando `fetch` permite seleccionar datos de las columnas definidas por el cursor y almacenarlos en variables locales.

Las siguientes operaciones de `fetch` están disponibles:

a) Para cursores declarados sin la cláusula `scroll`:

Operación	Descripción
<code>fetch next</code> <code>from curPub</code>	Fetch la próxima fila.

b) Para cursores declarados con la cláusula `scroll`:

Operación	Descripción
<code>fetch next</code> <code>from curPub</code>	Accede a la próxima fila. (Dirección <code>FETCH default</code> )
<code>fetch prior</code> <code>from curPub</code>	Accede a la fila inmediatamente anterior a la actual. (No la última fila fetchheada)
<code>fetch first</code> <code>from curPub</code>	Accede a la primer fila del conjunto resultado. Es útil para cuando se desea comenzar del principio del cursor sin cerrarlo y reabrirlo.
<code>fetch last</code> <code>from curPub</code>	Accede a la última fila del conjunto resultado
<code>fetch absolute n</code> <code>from curPub</code>	Se hace un <code>fetch</code> a la n-ésima fila del conjunto resultado.
<code>fetch relative n</code> <code>from curPub</code>	Fetch a la fila que está a n filas de la fila actual.

```
Fetch next
  from curPublicaciones
  Into @title, @type, @price
```

### La variable global @@fetch\_status

Retorna el status de la última operación `fetch` contra un cursor.  
Esta *variable global* posee un valor en el contexto de sus sesión y es modificada cada vez que se hace un `fetch` a una fila.

#### Valores posibles

Valor	Significado
0	El último fetch tuvo éxito.
-1	El último fetch falló porque el cursor ha alcanzado el final del conjunto resultado o la fila especificada no existe.
-2	La fila a la que se intenta acceder ha sido eliminada o su clave ha sido actualizada luego de abierto el cursor.

## 4) Cerrar el cursor

```
Close curPublicaciones
```

## 5) Destruir el cursor

Libera la memoria del server usada por el cursor.

```
deallocate curPublicaciones
```

**cursor1.sql**

```
Declare curPublicaciones cursor
for
    Select title, type, price
    From titles
    Order By type, title
    For read only

Open curPublicaciones

Set NoCount On

Declare @type Char(12),
        @price money,
        @title varchar(80),
        @Cad varchar(40)

print "Titulo                                Precio"
print "-----"

Fetch next
    from curPublicaciones
    Into @title, @type, @price

While @@fetch_status = 0
Begin
    -- Si el tipo es business, incremento el precio en un 25% (pero solo para mostrarlo)

    If @type = 'business'
        Select @price = @price * 1.25
    -- End If

    -- Si el precio es nulo, muestro un mensaje, sino muestro el titulo junto a su precio

    If @price is Null
        Select @Cad = Convert(Char(22), @title) + " No posee precio "
    Else
        Select @Cad = Convert(Char(22), @title) + Convert(Char(7), @price)
    -- End If

    Print @Cad

    Fetch next
        from curPublicaciones
        Into @title, @type, @price
End
-- End While

Close curPublicaciones
deallocate curPublicaciones
```

## La cláusula current of

---

Las sentencias Update y Delete soportan una sintaxis especial de la *cláusula* Where que le indica a SQL Server que el *contexto del query* es la *fila más recientemente alcanzada* a través del cursor especificado.

Mientras el programa se mueve a través de las filas del *conjunto resultado*, puede necesitar modificar o eliminar algunas de ellas.

Las *sentencias* Update y Delete con la *cláusula* Current Of afectan solo una fila, la última sobre la que se hizo *fetch*.

### UpdateDeTitle.sql

Se desea modificar el titulo 'The Gourmet Microwave' a 'The Gourmet Microwave Sintetizador'

```
-- select title_id, title from titles where title like 'The gourmet%'
```

```
Declare curPublicaciones Cursor
```

```
For
```

```
    Select title
```

```
    From Titles
```

```
Declare @title varchar(80)
```

```
Open curPublicaciones
```

```
Fetch Next
```

```
    From curPublicaciones
```

```
    Into @title
```

```
While @@fetch_status = 0
```

```
    Begin
```

```
        If @title = 'The Gourmet Microwave'
```

```
            Update titles
```

```
                Set title = title + ' Parte II'
```

```
                where current of curPublicaciones
```

```
        -- End If
```

```
        Fetch Next
```

```
            From curPublicaciones
```

```
            Into @title
```

```
    End
```

```
-- End While
```

```
Close curPublicaciones
```

```
Deallocate curPublicaciones
```

## Ejercicio 1

---

Implemente un batch que actualice los precios de las publicaciones de la editorial '0736'. Por cada publicación de esta editorial, se desea incrementar en un 25% el precio de las publicaciones que cuestan \$10 o menos y decrementar también en un 25% las publicaciones que cuestan más de \$10.

## Ejercicio 2

---

Defina un batch que actualice los precios de las publicaciones de la siguiente manera: Si las ventas anuales son mayores a 1000, rebaje el precio en un 10%. Para las publicaciones con menor venta anual, rebaje el precio en un 50% pero solo si el tipo es `popular_comp` o el título incluye la palabra `computer`. Para el resto de las publicaciones implemente un descuento del 25%. Informe como estaban las publicaciones antes de la actualización y como quedaron después, y la cantidad de publicaciones rebajadas.