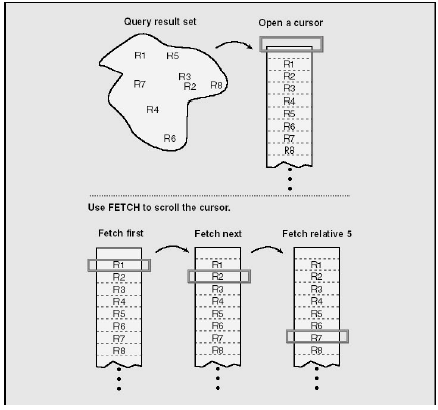
**Cursores**

**Introducción**

Un cursor nos permite movernos por los registros de una consulta (el resultado obtenido de una sentencia SELECT) uno por uno. Esto es muy útil si lo que queremos es realizar una acción si el código identificador de un registro es autonumérico y no lo conocemos hasta que se inserta el registro y queremos insertar ese identificador en cualquier otra tabla.



**Características**

Las instrucciones de SQL Server producen un conjunto completo de resultados, pero hay ocasiones en que los resultados se procesan mejor de fila en fila. Abrir un cursor sobre un conjunto de resultados permite procesar el conjunto de resultados de fila en fila. Puede asignar un cursor a una variable o parámetro con un tipo de datos cursor.

Los cursores permiten operar individualmente registro por registro a partir de un resultado de una consulta.

DECLARE CURSOR define los atributos de un cursor de servidor de Transact-SQL, como su comportamiento de desplazamiento y la consulta utilizada para generar el conjunto de resultados sobre el que opera el cursor. La instrucción OPEN llena el conjunto de resultados y la instrucción FETCH devuelve una fila del conjunto de resultados. La instrucción CLOSE libera el conjunto de resultados actual asociado al cursor. La instrucción DEALLOCATE libera los recursos que utiliza el cursor.

DECLARE au\_cursor CURSOR FOR

SELECT \* FROM authors ORDER by au\_lname

OPEN au\_cursor

WHILE (more rows)

FETCH NEXT FROM au\_cursor    -- Do until end

CLOSE au\_cursor

DEALLOCATE au\_cursor

**Sentencia OPEN**

Abre un cursor del servidor Transact-SQL y lo llena ejecutando la instrucción Transact-SQL especificada en la instrucción DECLARE CURSOR o SET cursor\_variable.

**Sentencia FETCH**

Recupera una fila específica de un cursor de servidor de Transact-SQL. Esta sentencia se utiliza para leer los registros del cursor: Su sintaxis es la siguiente:

FETCH

          [ [ NEXT | PRIOR | FIRST | LAST

                    | ABSOLUTE { n | @nvar }

                    | RELATIVE { n | @nvar }

               ]

               FROM

          ]

{ { [ GLOBAL ] cursor\_name } | @cursor\_variable\_name }

[ INTO @variable\_name [ ,...n ] ]

Sus argumentos son:

**NEXT**

Devuelve la fila de resultados inmediatamente después de la fila actual y se posiciona en la fila devuelta.

**PRIOR**

Devuelve la fila de resultados inmediatamente anterior de la fila actual y se posiciona en la fila devuelta.

**FIRST**

Devuelve la primera fila en el cursor y la convierte en la fila actual.

**LAST**

Devuelve la última  fila en el cursor y la convierte en la fila actual.

**ABSOLUTE { n | @nvar}**

Si n o @nvar es positivo, devuelve la fila n de del cursor y hace la fila devuelta la fila actual (o sea, se posiciona en la fila n). Si n o @ nvar es negativo, devuelve la fila n antes de que finalice el cursor y hace que la fila devuelta sea la fila actual. Si n o @ nvar es 0, no se devuelven filas. n debe ser una constante entera y @ nvar debe smallint, tinyint o int.

**RELATIVE { n | @nvar}**

Trabaja similar a ABSOLUTE pero devuelve la fila n desde la fila actual donde esta posicionado el cursor.

**Sentencia CLOSE**

Cierra un cursor abierto mediante la liberación del conjunto actual de resultados y todos los bloqueos de cursor mantenidos en las filas en las que está colocado.

CLOSE deja las estructuras de datos accesibles para que se puedan volver a abrir, pero las recuperaciones y las actualizaciones posicionadas no se permiten hasta que se vuelva a abrir el cursor.

CLOSE debe ejecutarse en un cursor abierto, por lo que no se permite en cursores que sólo están declarados o que ya están cerrados.

**Sentencia DEALLOCATE**

Quita una referencia a un cursor. Cuando se ha quitado la última referencia al cursor, Microsoft SQL Server libera las estructuras de datos que componen el cursor.

**Un cursor es** una estructura de datos creada en memoria RAM producto de una sentencia SELECT y que nos permite navegar dentro de las filas para obtener la información.

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

#### Ejemplo de cursor

A continuación, juntaremos todos los pasos descriptos previamente y crearemos el ejemplo más simple de un cursor el cual recorrerá la tabla Production.ProductDescription de la base de datos AdventureWorks y mostrará por pantalla el valor del registro Description luego que el mismo ha sido copiado a la variable @Description.

|  |  |
| --- | --- |
|  | DECLARE @Description AS nvarchar(400) |
|  | DECLARE ProdInfo CURSOR FOR SELECT [Description] FROM Production.ProductDescription | |

|  |  |
| --- | --- |
|  | OPEN ProdInfo |
|  | FETCH NEXT FROM ProdInfo INTO @Description | |

|  |  |  |
| --- | --- | --- |
|  | WHILE @@fetch\_status = 0 | |
|  | BEGIN |

-

|  |  |
| --- | --- |
|  | PRINT @Description |
|  | FETCH NEXT FROM ProdInfo INTO @Description | |

|  |  |  |
| --- | --- | --- |
|  | END | |
|  | | CLOSE ProdInfo | |

|  |  |
| --- | --- |
|  | DEALLOCATE ProdInfo |

En el ejemplo puede verse la utilización de la variable @@FETCH\_STATUS para verificar la existencia de la fila leída, por otra parte la sentencia FETCH NEXT devolverá la próxima fila de datos, siempre que la misma exista.