

# FUNCIONAMIENTO DE CURSORES EN SQL SERVER

Empecemos con la sintaxis. SQL Server maneja la sintaxis estándar ISO y una sintaxis extendida propia. Las opciones en negrita indican valores **default**.

## ISO Syntax

```
DECLARE cursor_name [ INSENSITIVE ] [ SCROLL ] CURSOR
    FOR select_statement
    [ FOR { READ ONLY | UPDATE [ OF column_name [ ,...n ] ] } ]
[;]
```

## Transact-SQL Extended Syntax

```
DECLARE cursor_name CURSOR [ LOCAL | GLOBAL ]
    [ FORWARD ONLY | SCROLL ]
    [ STATIC | KEYSET | DYNAMIC | FAST_FORWARD ]
    [ READ ONLY | SCROLL_LOCKS | OPTIMISTIC ]
    [ TYPE_WARNING ]
    FOR select_statement
    [ FOR UPDATE [ OF column_name [ ,...n ] ] ]
[;]
```

**INSENSITIVE:** El cursor realiza una copia de los datos en la BD **tempdb** y cualquier modificación en los datos no se refleja en las filas del cursor.

**SCROLL:** El cursor permite recorrer el cursor de diferentes maneras: (FIRST, LAST, PRIOR, NEXT, RELATIVE, ABSOLUTE).

**READ ONLY:** No permite modificar los datos del cursor a través de sentencias **UPDATE** o **DELETE WHERE CURRENT OF**.

**UPDATE:** Permite modificar las filas del cursor a través de la cláusula **WHERE CURRENT OF**. Si se especifican columnas solo se podrán modificar dichas columnas.

**LOCAL/GLOBAL:** Especifica que el **ámbito** de validez del cursor es el **Stored Procedure** o **Trigger** en el cual el cursor fue creado. Si es definido como **GLOBAL** el ámbito es la sesión.

## **FORWARD\_ONLY**

Especifica que el cursor solo puede ser recorrido en orden, desde el primero al último registro.

## **STATIC**

Tiene el mismo comportamiento que la opción **INSENSITIVE**.

## **DYNAMIC**

Define un cursor que refleja todos los cambios de datos realizados en las filas de su conjunto de resultados a medida que se desplaza por el cursor y se captura un nuevo registro, independientemente de si los cambios se producen desde el cursor o por otros usuarios fuera del cursor.

## **KEYSET**

Especifica que el conjunto de filas del cursor es fijo. Se crea un conjunto de claves que identifican unívocamente las filas creando una tabla en **tempdb** conocido como **keyset**. Provee funcionalidad entre un cursor dinámico y estático. Cuando se accede a cada fila el keyset es utilizado para leer las filas desde la fuente de datos. Cambios a los valores de datos son visibles cuando se recorre el conjunto de resultados:

- Si se elimina una fila, un intento de capturar la fila devuelve un **@@FETCH\_STATUS -2** porque la fila eliminada aparece como un hueco en el conjunto de resultados. La clave de la fila existe en el conjunto de claves, pero ya no existe en el conjunto de resultados.
- Las inserciones realizadas fuera del cursor (por otros procesos) solo son visibles si el cursor se cierra y se vuelve a abrir. Las inserciones realizadas desde el interior del cursor son visibles al final del conjunto de resultados.
- Las actualizaciones de los valores de clave desde fuera del cursor se asemejan a una eliminación de la fila anterior seguida de una inserción de la nueva fila. La fila con los nuevos valores no está visible y si se intenta capturar la fila con los valores antiguos devuelven una @@FETCH\_STATUS de -2. Los nuevos valores son visibles si la actualización se realiza a través del cursor especificando la cláusula WHERE CURRENT OF (desde dentro del cursor).

**Nota:** Si el query hace referencia a una tabla sin un índice único, el cursor es convertido a STATIC.

#### **FAST\_FORWARD**

Especifica un cursor del tipo *FORWARD\_ONLY*, *READ\_ONLY* con optimizaciones de performance habilitadas. No se puede especificar si se especifica *SCROLL* o *FOR\_UPDATE*. No permite modificaciones desde el cursor.

#### **READ\_ONLY**

No permite actualizaciones desde el cursor.

#### **SCROLL\_LOCKS**

Asegura que *UPDATES* y *DELETES* se realicen ya que SQL Server bloquea la fila cuando es leída.

#### **OPTIMISTIC**

Especifica que las actualizaciones o eliminaciones realizadas a través del cursor no se realizan si la fila se ha actualizado desde que se leyó en el cursor.

#### **TYPE\_WARNING**

Especifica que un mensaje de advertencia se envía al cliente cuando el cursor es convertido de un tipo a otro. Ej. Cursor definido como *KEYSET* sin PK.

#### **Nota**

SQL Server convierte implícitamente el cursor a otro tipo cuando la cláusula *SELECT* tiene conflictos con el tipo de cursor requerido.