**Why and When to use Cursor?**  
  
There are some conditions when we want to get record from one table and need to insert into another with performing some logic or some conditions .For example if we want to get value from one table row by row  and need to perform some logic over that and update /insert into another table then we can use cursors. Cursor basically works as for/While loop.

**Advantages of using Cursor:**

* Using Cursor we can perform row by row processing so we can perform row wise validation or operations on each row.
* Cursors can provide the first few rows before the whole result set is assembled. Without using cursors, the entire result set must be delivered before any rows are displayed by the application. So using cursor, better response time is achieved.
* If we make updates to our without using cursors in your application then we must send separate SQL statements to the database server to apply the changes. This can cause the possibility of concurrency problems if the result set has changed since it was queried by the client. In turn, this raises the possibility of lost updates. So using cursor, better concurrency Control can be achieved.
* Cursors can be faster than a while loop but at the cost of more overhead.

**Disadvantages of using Cursor:** 

* Cursor in SQL is temporary work area created in the system memory, thus it occupies memory from your system that may be available for other processes. So occupies more resources and temporary storage.
* Each time when a row is fetched from the cursor it may result in a network round trip. This uses much more network bandwidth than the execution of a single SQL statement like SELECT or DELETE etc that makes only one round trip.
* Repeated network round trips can degrade the speed of the operation using the cursor.

**Cursor Example:**

Implementation: Let's create the cursor to check its working.

* First of all create a table in Sql server with the script mentioned below:

 CREATE TABLE BookDetails

(

                [BookId] [int] IDENTITY(1,1) NOT NULL,

                [BookName] [varchar](100) NULL,

                [Author] [varchar](100) NULL,

                [Publisher] [varchar](200) NULL,

                [Price] [decimal](18, 2) NULL

)

* Then insert some records in the BookDetails table.

**Example demonstration the use of Cursor**

Suppose we want to get the Book Name, its Author, Price and based on price, calculating the discount(10% of price) and the calculated discounted price. For this we can create a cursor that will fetch the records one by one and calculate the discount and the discounted price and get all the details that we can display on any data control like GridView, Repeater, DataList etc.

**Let's create a cursor**

CREATE PROCEDURE GetBookDetails\_Sp

AS

BEGIN

 CREATE TABLE #temp

                                (

                                                BookName                         varchar(100),

                                                Author                 varchar(100),

                                                Price                                     int,

                                                Discount                             int,

                                                DiscountedPrice             int,

                                )

                SET NOCOUNT ON

                DECLARE @name varchar(100)

                DECLARE @auth varchar(100)

                DECLARE @prc int

                DECLARE curBookDetails CURSOR

                STATIC FOR

                SELECT BookName,Author,Price from BookDetails

                OPEN curBookDetails

                IF @@CURSOR\_ROWS > 0

                 BEGIN

                                FETCH NEXT FROM curBookDetails INTO @name,@auth,@prc

                 WHILE @@Fetch\_status = 0

                 BEGIN

                                INSERT INTO #temp (BookName,Author,Price,Discount,DiscountedPrice) VALUES (@name,@auth,@prc,((@prc\*10)/100),@prc-((@prc\*10)/100))

                                FETCH NEXT FROM curBookDetails INTO @name,@auth,@prc

                 END

                END

                SELECT \* FROM #temp

                CLOSE curBookDetails

                DROP TABLE #temp

                DEALLOCATE curBookDetails

                SET NOCOUNT OFF

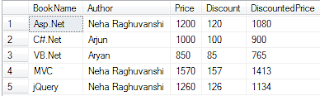
END

Note: I am using temporary table "#temp" to temporary store the BookName, Author, Price, calculated Discount and calculated DiscountedPrice. And then getting the records that we can use to fill in any data control to be displayed on the appliciation.

* To check the results execute the stored procedure using the command mentioned below:

exec GetBookDetails\_Sp

* On execution it will  get the Book details as shown in image below:

[](https://2.bp.blogspot.com/-fMoEKs4-Wnw/UpeG6DvGGdI/AAAAAAAABQM/7ZpAEm6FTGk/s1600/Cursor+example+in+SQL+SERVER.png)

Note:  @@Fetch\_status: is a transact-SQL statement that returns the status of the last cursor FETCH statement issued against any cursor currently opened by the connection. It returns the Boolean value 0 if the Fetch statement was successful and returns -1 if the FETCH statement failed or the row was beyond the result set and will return -2 if the row fetched is missing. For more details read the [@@FETCH\_STATUS details](http://technet.microsoft.com/en-us/library/ms187308.aspx)

Note: @@CURSOR\_ROWS: is a transact-SQL statement that returns the number of qualifying rows currently in the last cursor opened on the connection. For more details read the [@@CURSOR\_ROWS details](http://technet.microsoft.com/en-us/library/ms176044.aspx)

***Now over to you:***

*" I hope you have got what is Cursor in Sql Server with the example and if you have any point regarding cursor then please suggest. If you like my work; you can appreciate by leaving your comments, hitting Facebook like button, following on Google+, Twitter, Linked in and Pinterest, stumbling my posts on stumble upon and subscribing for receiving free updates directly to your inbox . Stay tuned and stay connected for more technical updates."*