**Example SQL Server Cursor**

A **cursor** allows you to iterate a set of rows returned by a query and process each row accordingly. You cannot fetch rows in the reversed order.

SQL Server Cursor Components

Based on the example above, cursors include these components:

* **DECLARE statements** - Declare variables used in the code block
* **SET\SELECT statements** - Initialize the variables to a specific value
* **DECLARE CURSOR statement** - Populate the cursor with values that will be evaluated
  + NOTE - There are an equal number of variables in the DECLARE CURSOR FOR statement as there are in the SELECT statement.  This could be 1 or many variables and associated columns.
* **OPEN statement** - Open the cursor to begin data processing
* **FETCH NEXT statements** - Assign the specific values from the cursor to the variables
  + NOTE - This logic is used for the initial population before the WHILE statement and then again during each loop in the process as a portion of the WHILE statement
* **WHILE statement** - Condition to begin and continue data processing
* **BEGIN...END** statement - Start and end of the code block
  + NOTE - Based on the data processing multiple BEGIN...END statements can be used
* Data processing - In this example, this logic is to backup a database to a specific path and file name, but this could be just about any DML or administrative logic
* CLOSE statement - Releases the current data and associated locks, but permits the cursor to be re-opened
* DEALLOCATE statement - Destroys the cursor

## **Types of Cursors**

## **Static Cursors**

A static cursor populates the result set at the time of cursor creation and query result is cached for the lifetime of the cursor. A static cursor can move forward and backward direction. A static cursor is slower and use more memory in comparison to other cursor. Hence you should use it only if scrolling is required and other types of cursors are not suitable.

No UPDATE, INSERT, or DELETE operations are reflected in a static cursor (unless the cursor is closed and reopened). By default static cursors are scrollable. SQL Server static cursors are always read-only.

## **Dynamic Cursors**

A dynamic cursor allows you to see the data updation, deletion and insertion in the data source while the cursor is open. Hence a dynamic cursor is sensitive to any changes to the data source and supports update, delete operations. By default dynamic cursors are scrollable.

## **Forward Only Cursors**

A forward only cursor is the fastest cursor among the all cursors but it doesn't support backward scrolling. You can update, delete data using Forward Only cursor. It is sensitive to any changes to the original data source.

There are three more types of Forward Only Cursors.Forward\_Only KEYSET, FORWARD\_ONLY STATIC and FAST\_FORWARD.

A **FORWARD\_ONLY STATIC Cursor** is populated at the time of creation and cached the data to the cursor lifetime. It is not sensitive to any changes to the data source.

A **FAST\_FORWARD Cursor** is the fastest cursor and it is not sensitive to any changes to the data source.

## **Keyset Driven Cursors**

A keyset driven cursor is controlled by a set of unique identifiers as the keys in the keyset. The keyset depends on all the rows that qualified the SELECT statement at the time of cursor was opened. A keyset driven cursor is sensitive to any changes to the data source and supports update, delete operations. By default keyset driven cursors are scrollable.

DECLARE @name VARCHAR(50) -- database name

DECLARE @path VARCHAR(256) -- path for backup files

DECLARE @fileName VARCHAR(256) -- filename for backup

DECLARE @fileDate VARCHAR(20) -- used for file name

SET @path = 'C:\Backup\'

SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)

DECLARE db\_cursor CURSOR FOR

SELECT name

FROM MASTER.dbo.sysdatabases

WHERE name NOT IN ('master','model','msdb','tempdb')

OPEN db\_cursor

FETCH NEXT FROM db\_cursor INTO @name

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @fileName = @path + @name + '\_' + @fileDate + '.BAK'

BACKUP DATABASE @name TO DISK = @fileName

FETCH NEXT FROM db\_cursor INTO @name

END

CLOSE db\_cursor

DEALLOCATE db\_cursor

## **SQL SERVER – Examples of Cursors**

CREATE TABLE Employee

(

EmpID int PRIMARY KEY,

EmpName varchar (50) NOT NULL,

Salary int NOT NULL,

Address varchar (200) NOT NULL,

)

GO

INSERT INTO Employee(EmpID,EmpName,Salary,Address)VALUES(1,'Mohan',12000,'Noida')

INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(2,'Pavan',25000,'Delhi')

INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(3,'Amit',22000,'Dehradun')

INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(4,'Sonu',22000,'Noida')

INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(5,'Deepak',28000,'Gurgaon')

GO

SELECT \* FROM Employee

## **Static Cursor - Example**

SET NOCOUNT ON

DECLARE @Id int

DECLARE @name varchar(50)

DECLARE @salary int

DECLARE cur\_emp CURSOR

STATIC FOR

SELECT EmpID,EmpName,Salary from Employee

OPEN cur\_emp

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM cur\_emp INTO @Id,@name,@salary

WHILE @@Fetch\_status = 0

BEGIN

PRINT 'ID : '+ convert(varchar(20),@Id)+', Name : '+@name+ ', Salary : '+convert(varchar(20),@salary)

FETCH NEXT FROM cur\_emp INTO @Id,@name,@salary

END

END

CLOSE cur\_emp

DEALLOCATE cur\_emp

SET NOCOUNT OFF

## **Dynamic Cursor - Example**

--Dynamic Cursor for Update

SET NOCOUNT ON

DECLARE @Id int

DECLARE @name varchar(50)

DECLARE Dynamic\_cur\_empupdate CURSOR

DYNAMIC

FOR

SELECT EmpID,EmpName from Employee ORDER BY EmpName

OPEN Dynamic\_cur\_empupdate

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM Dynamic\_cur\_empupdate INTO @Id,@name

WHILE @@Fetch\_status = 0

BEGIN

IF @name='Mohan'

Update Employee SET Salary=15000 WHERE CURRENT OF Dynamic\_cur\_empupdate

FETCH NEXT FROM Dynamic\_cur\_empupdate INTO @Id,@name

END

END

CLOSE Dynamic\_cur\_empupdate

DEALLOCATE Dynamic\_cur\_empupdate

SET NOCOUNT OFF

Go

Select \* from Employee

-- Dynamic Cursor for DELETE

SET NOCOUNT ON

DECLARE @Id int

DECLARE @name varchar(50)

DECLARE Dynamic\_cur\_empdelete CURSOR

DYNAMIC

FOR

SELECT EmpID,EmpName from Employee ORDER BY EmpName

OPEN Dynamic\_cur\_empdelete

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM Dynamic\_cur\_empdelete INTO @Id,@name

WHILE @@Fetch\_status = 0

BEGIN

IF @name='Deepak'

DELETE Employee WHERE CURRENT OF Dynamic\_cur\_empdelete

FETCH NEXT FROM Dynamic\_cur\_empdelete INTO @Id,@name

END

END

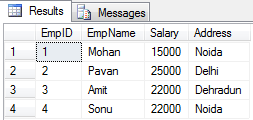
CLOSE Dynamic\_cur\_empdelete

DEALLOCATE Dynamic\_cur\_empdelete

SET NOCOUNT OFF

Go

Select \* from Employee



## **Forward Only Cursor - Example**

--Forward Only Cursor for Update

SET NOCOUNT ON

DECLARE @Id int

DECLARE @name varchar(50)

DECLARE Forward\_cur\_empupdate CURSOR

FORWARD\_ONLY

FOR

SELECT EmpID,EmpName from Employee ORDER BY EmpName

OPEN Forward\_cur\_empupdate

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM Forward\_cur\_empupdate INTO @Id,@name

WHILE @@Fetch\_status = 0

BEGIN

IF @name='Amit'

Update Employee SET Salary=24000 WHERE CURRENT OF Forward\_cur\_empupdate

FETCH NEXT FROM Forward\_cur\_empupdate INTO @Id,@name

END

END

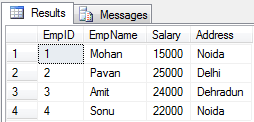
CLOSE Forward\_cur\_empupdate

DEALLOCATE Forward\_cur\_empupdate

SET NOCOUNT OFF

Go

Select \* from Employee



-- Forward Only Cursor for Delete

SET NOCOUNT ON

DECLARE @Id int

DECLARE @name varchar(50)

DECLARE Forward\_cur\_empdelete CURSOR

FORWARD\_ONLY

FOR

SELECT EmpID,EmpName from Employee ORDER BY EmpName

OPEN Forward\_cur\_empdelete

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM Forward\_cur\_empdelete INTO @Id,@name

WHILE @@Fetch\_status = 0

BEGIN

IF @name='Sonu'

DELETE Employee WHERE CURRENT OF Forward\_cur\_empdelete

FETCH NEXT FROM Forward\_cur\_empdelete INTO @Id,@name

END

END

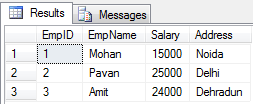
CLOSE Forward\_cur\_empdelete

DEALLOCATE Forward\_cur\_empdelete

SET NOCOUNT OFF

Go

Select \* from Employee



## **Keyset Driven Cursor - Example**

-- Keyset driven Cursor for Update

SET NOCOUNT ON

DECLARE @Id int

DECLARE @name varchar(50)

DECLARE Keyset\_cur\_empupdate CURSOR

KEYSET

FOR

SELECT EmpID,EmpName from Employee ORDER BY EmpName

OPEN Keyset\_cur\_empupdate

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM Keyset\_cur\_empupdate INTO @Id,@name

WHILE @@Fetch\_status = 0

BEGIN

IF @name='Pavan'

Update Employee SET Salary=27000 WHERE CURRENT OFKeyset\_cur\_empupdate

FETCH NEXT FROM Keyset\_cur\_empupdate INTO @Id,@name

END

END

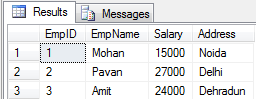
CLOSE Keyset\_cur\_empupdate

DEALLOCATE Keyset\_cur\_empupdate

SET NOCOUNT OFF

Go

Select \* from Employee



-- Keyse Driven Cursor for Delete

SET NOCOUNT ON

DECLARE @Id int

DECLARE @name varchar(50)

DECLARE Keyset\_cur\_empdelete CURSOR

KEYSET

FOR

SELECT EmpID,EmpName from Employee ORDER BY EmpName

OPEN Keyset\_cur\_empdelete

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM Keyset\_cur\_empdelete INTO @Id,@name

WHILE @@Fetch\_status = 0

BEGIN

IF @name='Amit'

DELETE Employee WHERE CURRENT OF Keyset\_cur\_empdelete

FETCH NEXT FROM Keyset\_cur\_empdelete INTO @Id,@name

END

END

CLOSE Keyset\_cur\_empdelete

DEALLOCATE Keyset\_cur\_empdelete

SET NOCOUNT OFF

Go Select \* from Employee

