Cursor is a database object to retrieve data from a result set one row at a time, instead of the T-SQL commands that operate on all the rows in the result set at one time. We use cursor when we need to update records in a database table in singleton fashion means row by row.

Life Cycle of Cursor

1. Declare Cursor

A cursor is declared by defining the SQL statement that returns a result set.

1. Open

A Cursor is opened and populated by executing the SQL statement defined by the cursor.

1. Fetch

When cursor is opened, rows can be fetched from the cursor one by one or in a block to do data manipulation.

1. Close

After data manipulation, we should close the cursor explicitly.

1. Deallocate

Finally, we need to delete the cursor definition and released all the system resources associated with the cursor.

Syntax to Declare Cursor

Declare Cursor SQL Command is used to define the cursor with many options that impact the scalability and loading behavior of the cursor. The basic syntax is given below

1. DECLARE cursor\_name CURSOR
2. [LOCAL | GLOBAL] --define cursor scope
3. [FORWARD\_ONLY | SCROLL] --define cursor movements (forward/backward)
4. [STATIC | KEYSET | DYNAMIC | FAST\_FORWARD] --basic type of cursor
5. [READ\_ONLY | SCROLL\_LOCKS | OPTIMISTIC] --define locks
6. FOR select\_statement --define SQL Select statement
7. FOR UPDATE [col1,col2,...coln] --define columns that need to be updated

Syntax to Open Cursor

A Cursor can be opened locally or globally. By default it is opened locally. The basic syntax to open cursor is given below:

1. OPEN [GLOBAL] cursor\_name --by default it is local

Syntax to Fetch Cursor

Fetch statement provides the many options to retrieve the rows from the cursor. NEXT is the default option. The basic syntax to fetch cursor is given below:

1. FETCH [NEXT|PRIOR|FIRST|LAST|ABSOLUTE n|RELATIVE n]
2. FROM [GLOBAL] cursor\_name
3. INTO @Variable\_name[1,2,..n]

Syntax to Close Cursor

Close statement closed the cursor explicitly. The basic syntax to close cursor is given below:

1. CLOSE cursor\_name --after closing it can be reopen

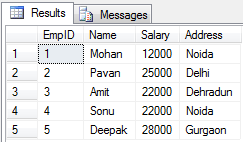
Syntax to Deallocate Cursor

Deallocate statement delete the cursor definition and free all the system resources associated with the cursor. The basic syntax to close cursor is given below:

1. DEALLOCATE cursor\_name --after deallocation it can't be reopen

SQL SERVER – Simple Examples of Cursors

1. CREATE TABLE Employee
2. (
3. EmpID int PRIMARY KEY,
4. EmpName varchar (50) NOT NULL,
5. Salary int NOT NULL,
6. Address varchar (200) NOT NULL,
7. )
8. GO
9. INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(1,'Mohan',12000,'Noida')
10. INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(2,'Pavan',25000,'Delhi')
11. INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(3,'Amit',22000,'Dehradun')
12. INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(4,'Sonu',22000,'Noida')
13. INSERT INTO Employee(EmpID,EmpName,Salary,Address) VALUES(5,'Deepak',28000,'Gurgaon')
14. GO
15. SELECT \* FROM Employee



1. SET NOCOUNT ON
2. DECLARE @Id int
3. DECLARE @name varchar(50)
4. DECLARE @salary int
5. DECLARE cur\_emp CURSOR
6. STATIC FOR
7. SELECT ,EmpName,Salary from Employee
8. OPEN cur\_emp
9. IF @@CURSOR\_ROWS > 0
10. BEGIN
11. FETCH NEXT FROM cur\_emp INTO @Id,@name,@salary
12. WHILE @@Fetch\_status = 0
13. BEGIN
14. PRINT 'ID : '+ convert(varchar(20),@Id)+', Name : '+@name+ ', Salary : '+convert(varchar(20),@salary)
15. FETCH NEXT FROM cur\_emp INTO @Id,@name,@salary
16. END
17. END
18. CLOSE cur\_emp
19. DEALLOCATE cur\_emp
20. SET NOCOUNT OFF

