SQL Server 2012 – Database Development

Lesson 2 : Working with Batches and Cursors

Capgemini

Lesson Objectives

- Understanding Batches
- Introduction to database programming
- Working with Cursors

Batches

- A batch is one or multiple T-SQL statements executed by SQL Server as a single unit.
- Batches are separated by a GO statement, which marks the end of one batch and beginning of another.
- This command must appear on a separate line from all other commands.
- Batches help in reducing network traffic as it compiles all the SQL statement into a single executable unit

Batches

- A compilation error such as syntax error prevents batch from getting executed
- A run time error has two kind of impact
  - Stop the batch from getting executed
  - If the error happens to be constraint violation , only the current statement is ignored
- All DDL statement have be in a separate batch , it cannot be combined with any other statements in a batch

Batches Example

- USE pubs
  GO
  /* Signals the end of the batch */

- SELECT * FROM auth_titles
  GO
  - /* Signals the end of the batch */

Transact-SQL Programming

- T-SQL supports programmatic functionality within the relational databases
- T-SQL integrates SQL and programming capabilities which helps in storing complex business logic inside the DBMS itself
- This feature allows multiple applications to reuse the business logic , irrespective of technology
- Business Logic can be stored in the form of procedures, functions and triggers

Transact-SQL Programming Constructs

- Basic Transact SQL Block

```
/*  .....
        Block Comments
        ......    */
   Declare  <variable declarations>
Begin
       Begin
                        ........ <statements >
            -- Single line comments
           End
     Begin
                          --exception handling code
         End
End
```

Programming Constructs in T-SQL

- Local Variables

```
DECLARE @Var1 INT
DECLARE @Var2 INT
SET @Var1 = 1
SET @Var2 = 2
SELECT @Var1 'Var1', @Var2 'Var2'
GO
```

```
DECLARE @Var1 INT, @Var2 INT
SET @Var1 = 1
SET @Var2 = 2
SELECT @Var1 'Var1', @Var2 'Var2'
GO
```

```
DECLARE @iVariable INT = 1, @vVariable VARCHAR(100) = 'myvar',
@dDateTime DATETIME = GETDATE()

SELECT @iVariable iVar, @vVariable vVar, @dDateTime dDT
GO
```

Variables

- Global Variables
  - Declared with @@
  - A lot of global variables are system defined -
- Some Predefined Global variables
  - @@ERROR
  - @@FETCH_STATUS
  - @@IDENTITY
  - @@ROWCOUNT
  - @@SERVERNAME
  - @@SPID
  - @@TRANCOUNT
  - @@VERSION

Using control of flow language

- if...else statement
- CASE statement

```
IF boolean_expression

{sql_statement|statement_block}
 ELSE boolean_expression
            {sql_statement|statement_block}]
```

```
USE AdventureWorks2012;
IF EXISTS
(
SELECT * FROM
Production.ProductInventory WHERE
Quantity = 0
)
BEGIN;
PRINT 'Replenish Inventory';
END;
```

Case Statement

```
1. CASE input_expression
     WHEN when_expression THEN result_expression
     [  ...n ]
     [     ELSE else_result_expression      ]
     END

2. CASE
      WHEN Boolean_expression THEN result_expression
      [ ...n ]
      [     ELSE else_result_expression      ]
      END
```

Case Statement

```
SELECT EmployeeID, 'Marital Status' =
CASE MaritalStatus
      WHEN 'M' THEN 'Married'
      WHEN 'S' THEN 'Single'
      ELSE 'Not specified'
END
FROM HumanResources.Employee
GO
```

While Statement

- Executes a batch repeatedly as long as the given condition holds true
- Uses BREAK and CONTINUE statements to break the loop or to continue the loop

```
WHILE Boolean_expression
      { sql_statement | statement_block }
   [ BREAK ]
      { sql_statement | statement_block }
 [ CONTINUE ]
     { sql_statement | statement_block }
```

Example

```
Use Adventureworks2012
WHILE (SELECT AVG(Rate)+1 from EmployeePayHistory) < 20
BEGIN
    UPDATE HumanResources.EmployeePayHistory
    SET Rate = Rate + 1
    FROM HumanResources.EmployeePayHistory
      IF (Select MAX(Rate)+1 from HumanResources.EmployeePayHistory)
              > 127

           BREAK
      ELSE
            CONTINUE
END
```

RETURN statement

- Unconditionally terminates a query, stored procedure, or batch.
- None of the statements in a stored procedure or batch following the RETURN statement are executed.
- When used in a stored procedure can specify an integer value to return to the calling application, batch, or procedure.
- If no value is specified on RETURN, a stored procedure returns the value 0.

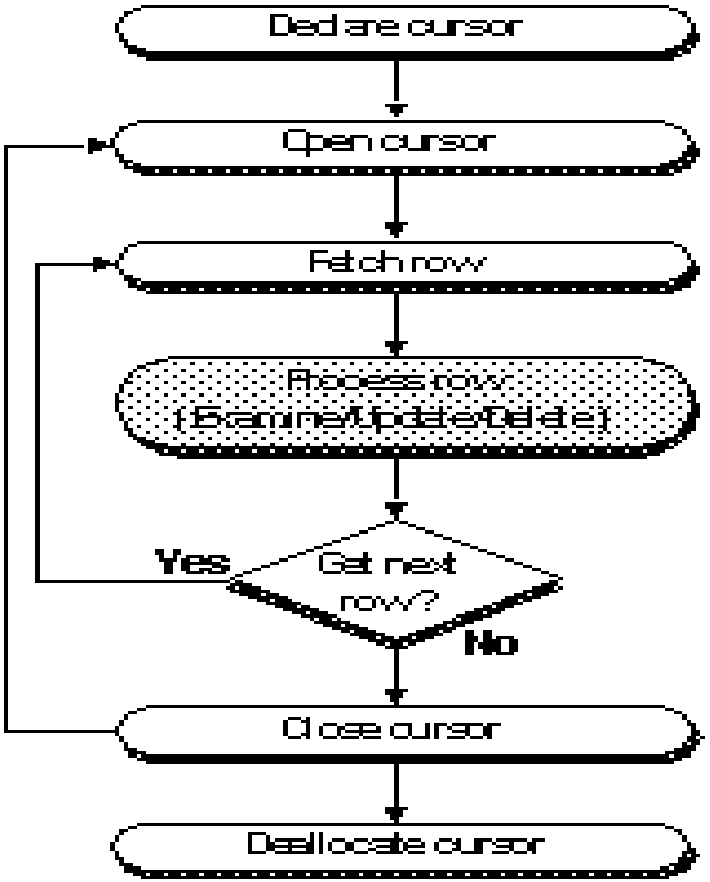
Usage

- Cursors: Accessing Data Row by Row
  - Is a data structure which helps in defining a result set and perform a complex business logic on each row of the result set
  - A cursor can be viewed as a pointer to one row in a set of rows
  - The main advantage is that we can process the data row-by -row.
  - They are NOT database objects
- NOTE : cursors are the SLOWEST way to access data inside SQL Server . Therefore they used be used only when there is an absolute neec

Cursor

PERFORMER_NAME: VARCHAR (60)	PLACE_OF_BIRTH: VARCHAR (60)
Jennifer Warnes	Seattle, Washington, USA
Joni Mitchell	Fort MacLeod, Alberta, Canada
William Acherman	Germany
Kitaro	Toyohashi, Japan
Bing Crosby	Tacoma, Washington, United States
Patsy Cline	Winchester, Virginia, United States
Jose Carreras	Barcelona, Spain
Luciano Pavarotti	Modena, Italy
Placido Domingo	Madrid, Spain

Steps for using cursor



Gets the result set

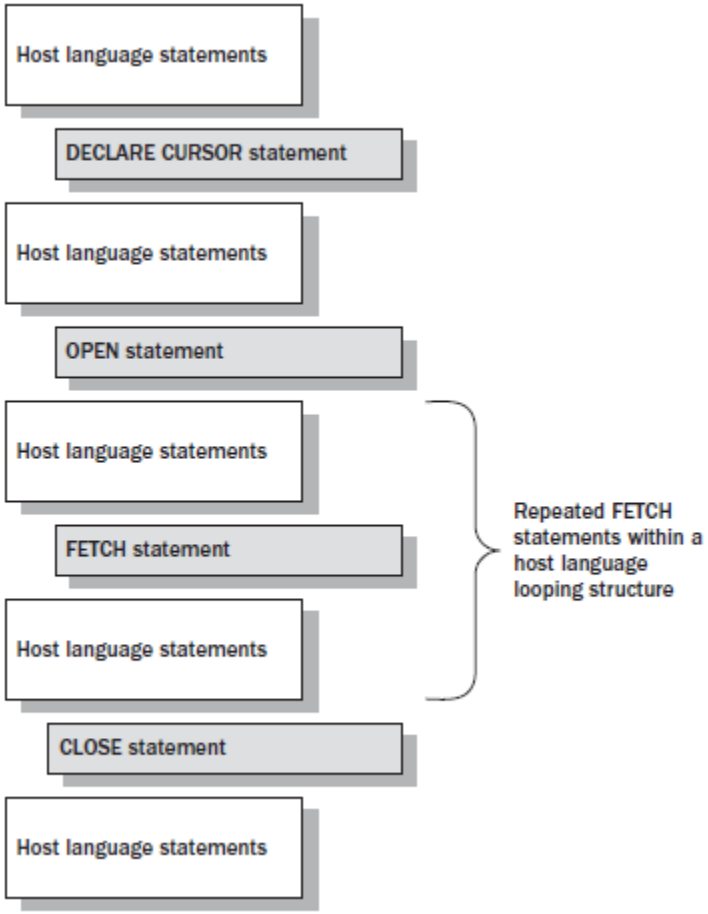
Cursor Syntax

```
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 ] ] ] [;]
```

CD Inventory Table

COMPACT_DISC: VARCHAR (60)	CATEGORY: VARCHAR (15)	PRICE: NUMERIC (5,2)	ON_HAND: INT
Famous Blue Raincoat	Vocal	16.99	13
Blue	Vocal	14.99	42
Court and Spark	Vocal	14.99	22
Past Light	Instrumental	15.99	17
Kojiki	Instrumental	15.99	6
That Christmas Feeling	Vocal	14.99	8
Patsy Cline: 12 Greatest Hits	Vocal	16.99	32
Carreras Domingo Pavarotti in Concert	Vocal	15.99	27
After the Rain: The Soft Sounds of Erik Satie	Instrumental	16.99	21
Out of Africa	Instrumental	16.99	29
Leonard Cohen The Best of	Vocal	15.99	12
Fundamental	Vocal	15.99	34
Blues on the Bayou	Vocal	14.99	27
Orlando	Instrumental	14.99	5

	COMPACT_DISC	CATEGORY	PRICE	ON_HAND
FETCH PRIOR				
FETCH FIRST FETCH NEXT	After the Rain: The Soft Sounds of Erik Satie	Instrumental	16.99	21
	Blue	Vocal	14.99	42
	Blues on the Bayou	Vocal	14.99	27
	Carreras Domingo Pavarotti in Concert	Vocal	15.99	27
FETCH ABSOLUTE 5	Court and Spark	Vocal	14.99	22
	Famous Blue Raincoat	Vocal	16.99	13
	Fundamental	Vocal	15.99	34
	Kojiki	Instrumental	15.99	6
	Leonard Cohen The Best of	Vocal	15.99	12
FETCH RELATIVE 10	Orlando	Instrumental	14.99	5
	Out of Africa	Instrumental	16.99	29
	Past Light	Instrumental	15.99	17
	Patsy Cline: 12 Greatest Hits	Vocal	16.99	32
FETCH LAST	That Christmas Feeling	Vocal	14.99	8



In each case, the pointer is based on a FETCH statement that is the first to be executed after the cursor has been opened.

Example

Without Cursor

```
SELECT name, database_id
 FROM sys.databases ;
```

With Cursor

```
DECLARE @DatabaseID as INT;
DECLARE @DatabaseName as NVARCHAR(50);
DECLARE @DatabaseCursor as CURSOR;
SET @DatabaseCursor = CURSOR FOR
SELECT name, database_id
  FROM sys.databases ;
OPEN @DatabaseCursor;
FETCH NEXT FROM @DatabaseCursor INTO @DatabaseName, @DatabaseID;
WHILE @@FETCH_STATUS = 0
BEGIN
 PRINT cast(@DatabaseID as VARCHAR (50)) + ' ' + @DatabaseName;
 FETCH NEXT FROM @DatabaseCursor INTO @DatabaseName, @DatabaseID;
END
CLOSE @DatabaseCursor;
DEALLOCATE @DatabaseCursor;
```

Types

Cursor Type	Description
Static	Cursor can move to any record but the changes on the data can't be seen.
Dynamic	Most resource extensive. Cursor can move anywhere and all the changes on the data can be viewed.
forward-only	Cursor moves one step forward. Can't move backwards.
Key setdriven	Only Updated data can be viewed. Deleted and Inserted data cannot be viewed.

Summary

- In this lesson, you have learnt:
- T-SQL batch statements
- Cursors and Types of Cursors

Review Question

- Question 1: what are the rules of working with Batch?
- Question 2:Can DDL statements be used in batch?
- Question 3: What are types of cursors?