Contents

[1. Prerequisite 1](#_Toc373870126)

[2. Variable Demo 2](#_Toc373870127)

[Assign value into a variable using SELECT 2](#_Toc373870128)

[Assign value into a tabe variable 3](#_Toc373870129)

[3. Control flow demo 3](#_Toc373870130)

[IF...ELSE 3](#_Toc373870131)

[CASE...WHEN 4](#_Toc373870132)

[Simple CASE...WHEN 4](#_Toc373870133)

[CASE...WHEN in Order By clause. 4](#_Toc373870134)

[TRY...CATCH 4](#_Toc373870135)

[WHILE 5](#_Toc373870136)

[GOTO 6](#_Toc373870137)

[4. Stored Procedure 6](#_Toc373870138)

[Using OUTPUT parameter 6](#_Toc373870139)

[Using a Return Code 7](#_Toc373870140)

[Return Using SELECT 9](#_Toc373870141)

[5. Function Demo 10](#_Toc373870142)

[Scalar function 10](#_Toc373870143)

[Inline Table-Valued function 11](#_Toc373870144)

[6. Trigger demo 13](#_Toc373870145)

[DML Trigger for INSERT action 13](#_Toc373870146)

[DML Trigger for UPDATE action 13](#_Toc373870147)

[DML Trigger for DELETE action 14](#_Toc373870148)

[DDL Trigger 15](#_Toc373870149)

# Prerequisite

Demos for this day use:

1) AdventureWorks database from

2) In Fsoft\_Training database which is created in Day2, run below script



# Variable Demo

## Assign value into a variable using SELECT

Copy/Paste and **Execute** below statement



## Assign value into a tabe variable

Copy/Paste and **Execute** below statement

# Control flow demo

## IF...ELSE

Copy and execute below statement.



## CASE...WHEN

### Simple CASE...WHEN



### CASE...WHEN in Order By clause.



## TRY...CATCH



## WHILE

If the average list price of a product is less than $300, the WHILE loop doubles the prices and then selects the maximum price. If the maximum price is less than or equal to $500, the WHILE loop restarts and doubles the prices again. This loop continues doubling the prices until the maximum price is greater than $500, and then exits the WHILE loop and prints a message



## GOTO



# Stored Procedure

## Using OUTPUT parameter

- Step 1: Create a SP named Sales.uspGetEmployeeSalesYTD with @SalesYTD as OUTPUT parameter



- Step2: Execute above SP



## Using a Return Code

- Step 1: Create a SP named usp\_GetSalesYTD



- Step 2: Handling return code from above SP



## Return Using SELECT

- Step 1: Copy and Run below code



- Step 2: Get result by using:



# Function Demo

## Scalar function

- Step 1: Create a function as below



- Step 2: Using that function



## Inline Table-Valued function

- Step 1: Create a function as below



- Step 2: Using that function



Multi-statement Table-Valued function

- Step 1: Create a function to split string base on delimeter as below



- Step 2: Using above function



# Trigger demo

## DML Trigger for INSERT action

- Step 1: Copy and execute below statement



- Step 2: Try insert a record by using:



## DML Trigger for UPDATE action

- Step 1: Copy and execute below statement



- Step 2: Try update by using:



## DML Trigger for DELETE action

- Step 1: Copy and execute below statement



- Step 2: Try delete a record will be inserted into Employee\_Audit

## DDL Trigger

- Step 1: Copy and execute below statement



- Step 2: Try to drop a table the trigger will be fired.