**Example 28.1**ALTER DATABASE SCOPED CONFIGURATION SET

BATCH\_MODE\_MEMORY\_GRANT\_FEEDBACK = ON;

**Example 28.2**

USE AdventureWorks;

SELECT product.Name, COUNT(history.ProductID) AS Cnt,

SUM(history.Quantity) AS Sum,

AVG(history.ActualCost) AS Avg

FROM Production.TransactionHistory AS history

JOIN Production.Product AS product

ON product.ProductID = history.ProductID

GROUP BY history.ProductID, product.Name;

**Example 28.3**

USE AdventureWorks;

CREATE NONCLUSTERED COLUMNSTORE INDEX csi\_history

ON Production.TransactionHistory(ProductID,Quantity,ActualCost);

**Example 28.4**

USE AdventureWorks;

GO

CREATE FUNCTION GetLastShipped()

RETURNS @CustomerOrder TABLE

(SaleOrderID INT NOT NULL,

CustomerID INT NOT NULL,

OrderDate DATETIME NOT NULL,

OrderQty INT NOT NULL)

AS

BEGIN

INSERT @CustomerOrder

SELECT a.SalesOrderID, a.CustomerID, a.OrderDate, b.OrderQty

FROM Sales.SalesOrderHeader a

INNER JOIN Sales.SalesOrderDetail b

ON a.SalesOrderID = b.SalesOrderID

INNER JOIN Production.Product c

ON b.ProductID = c.ProductID

WHERE a.OrderDate = ( Select Max(SH1.OrderDate)

FROM Sales.SalesOrderHeader As SH1

WHERE SH1.CustomerID = A.CustomerId)

RETURN

END

**Example 28.5**

USE AdventureWorks;

SELECT C = COUNT\_BIG(\*)

FROM GetLastShipped() C

**Example 28.6**

ALTER DATABASE SCOPED CONFIGURATION SET

INTERLEAVED\_EXECUTION\_TV = ON;

**Example 28.7**

EXEC sp\_configure 'show advanced options', 1;

GO

RECONFIGURE WITH OVERRIDE;

GO

EXEC sp\_configure 'max degree of parallelism', 4;

GO

RECONFIGURE WITH OVERRIDE;

GO

**Example 28.8**

USE sample;

CREATE NONCLUSTERED INDEX nCLI\_IFactSales

ON dbo.FactInternetSales

(OrderDateKey, CustomerKey, SalesAmount) ;

**Example 28.9**

USE sample;

SELECT c.CommuteDistance,

d.CalendarYear,

SUM(f.SalesAmount) TotalSalesByCommuteDistance

FROM dbo.FactInternetSales as f

INNER JOIN dbo.DimCustomer as c ON

f.CustomerKey = c.CustomerKey

INNER JOIN dbo.DimDate d ON

d.DateKey = f.OrderDateKey

GROUP BY c.CommuteDistance, d.CalendarYear;

**Example 28.10**

USE AdventureWorksDW;

SET STATISTICS TIME ON;

SELECT count(DISTINCT(SalesOrderNumber))

FROM FactInternetSales;

SELECT APPROX\_COUNT\_DISTINCT(SalesOrderNumber)

FROM FactInternetSales;

**Example 28.11**

ALTER DATABASE AdventureWorks SET COMPATIBILITY\_LEVEL = 140;  
GO  
USE AdventureWorks;  
GO  
CREATE FUNCTION dbo.CustomerRate14 (@CustomerID INT)  
 RETURNS CHAR(10) AS  
BEGIN;  
 DECLARE @sales DECIMAL (18,2);  
 DECLARE @category CHAR(10);  
 SET @sales = (SELECT SUM(Subtotal)   
 FROM Sales.SalesOrderHeader WHERE CustomerID = @CustomerID);  
IF @sales < 500000  
 SET @category = 'REGULAR';  
 ELSE IF @sales < 1000000  
 SET @category = 'GOLD';  
 ELSE  
 SET @category = 'PLATINUM';  
 RETURN @category;  
END

**Example 28.12**

ALTER DATABASE AdventureWorks SET COMPATIBILITY\_LEVEL = 140;  
GO  
USE AdventureWorks;  
GO  
CREATE FUNCTION dbo.CustomerRate14 (@CustomerID INT)  
 RETURNS CHAR(10) AS  
BEGIN;  
 DECLARE @sales DECIMAL (18,2);  
 DECLARE @category CHAR(10);  
 SET @sales = (SELECT SUM(Subtotal)   
 FROM Sales.SalesOrderHeader WHERE CustomerID = @CustomerID);  
IF @sales < 500000  
 SET @category = 'REGULAR';  
 ELSE IF @sales < 1000000  
 SET @category = 'GOLD';  
 ELSE  
 SET @category = 'PLATINUM';  
 RETURN @category;  
END