**Section : 03**

**Name : Sushma Suttakote Parameshappa**

**Challenge 1: Retrieve Product Price Information**

**Que 1:**

select A.ProductID,A.Name,A.ListPrice

from AdventureWorksLT2012.SalesLT.Product A

where A.ListPrice > (select AVG(B.UnitPrice) avgprice

from AdventureWorksLT2012.SalesLT.SalesOrderDetail AS B)

**Que 2:**

-- Retrieve products with a list price of $100 or more that have been sold for less than $100

SELECT A.ProductID,

A.Name,

A.ListPrice

FROM

(

SELECT A.ProductID,

A.Name,

A.ListPrice

FROM AdventureWorksLT2012.SalesLT.Product A

WHERE A.ListPrice > 100

) AS A

JOIN AdventureWorksLT2012.SalesLT.SalesOrderDetail AS B

ON A.ProductID = B.ProductID

WHERE B.UnitPrice < 100;

**Que 3:**

-Retrieve the cost, list price and average selling price for each product

SELECT p.ProductID,

p.Name,

p.ListPrice,

p.StandardCost,

B.avgprice

FROM AdventureWorksLT2012.SalesLT.Product AS p

JOIN

(

SELECT ProductID,

AVG(UnitPrice) avgprice

FROM AdventureWorksLT2012.SalesLT.SalesOrderDetail

GROUP BY ProductID

) AS B ON p.ProductID = B.ProductID;

**Que 4:**

--Retrieve the cost, list price and average selling price for each product

SELECT p.ProductID,

p.Name,

p.ListPrice,

p.StandardCost,

B.avgprice

FROM AdventureWorksLT2012.SalesLT.Product AS p

JOIN

(

SELECT ProductID,

AVG(UnitPrice) avgprice

FROM AdventureWorksLT2012.SalesLT.SalesOrderDetail

GROUP BY ProductID

) AS B ON p.ProductID = B.ProductID

WHERE p.StandardCost > B.avgprice;

**Challenge 2: Retrieve Customer Information**

**Que 1:**

**--Retrieve customer information for all sales orders**

CREATE FUNCTION dbo.ufnGetCustomerInformation(@CustomerId AS INT)

RETURNS @TREE TABLE

(

CustomerId INT NOT NULL

,FirstName VARCHAR(25) NOT NULL

,LastName VARCHAR(25) NOT NULL

)

AS

BEGIN

WITH Customer\_Subtree(CustomerID, FirstName, LastName)

AS

(

Select A.CustomerID,A.FirstName,A.LastName from AdventureWorksLT2012.SalesLT.Customer As A

)

INSERT INTO @TREE

SELECT \* FROM Customer\_Subtree;

RETURN

END;

Select B.SalesOrderID,B.TotalDue,C.CustomerID,C.FirstName,C.LastName

from AdventureWorksLT2012.SalesLT.SalesOrderHeader as B

CROSS APPLY ufnGetCustomerInformation(B.CustomerID) AS C;

**Que 2:**

**Retrieve customer address information**

CREATE FUNCTION dbo.ufnGetCustomerInformation(@CustomerId AS INT)

RETURNS @TREE TABLE

(

CustomerId INT NOT NULL

,FirstName VARCHAR(25) NOT NULL

,LastName VARCHAR(25) NOT NULL

)

AS

BEGIN

WITH Customer\_Subtree(CustomerID, FirstName, LastName)

AS

(

Select A.CustomerID,A.FirstName,A.LastName from AdventureWorksLT2012.SalesLT.Customer As A

)

INSERT INTO @TREE

SELECT \* FROM Customer\_Subtree;

RETURN

END;

Select C.CustomerID,C.FirstName,C.LastName,D.AddressLine1,D.City

from AdventureWorksLT2012.SalesLT.CustomerAddress as B

JOIN AdventureWorksLT2012.SalesLT.Address as D

ON B.AddressID=D.AddressID

CROSS APPLY ufnGetCustomerInformation(B.CustomerID) AS C;