**Section : 03**

**Name : Sushma Suttakote Parameshappa**

**Challenge 1: Creating scripts to insert sales orders**

**Que 1:**

-- Write code to insert an order header

--Creating the sequence

CREATE SEQUENCE SalesLT.SalesOrderNumber AS INT

START WITH 1

INCREMENT BY 1

GO

-- Script to Insert

SET IDENTITY\_INSERT AdventureWorksLT2012.SalesLT.SalesOrderHeader ON

DECLARE @SalesOrderId int = NEXT VALUE FOR SalesLT.SalesOrderNumber,

@OrderDate Datetime= GETDATE(),

@DueDate Datetime,

@CustomerId int =1;

INSERT INTO AdventureWorksLT2012.SalesLT.SalesOrderHeader(SalesOrderID,OrderDate,DueDate,CustomerID,ShipMethod)

VALUES(@SalesOrderId,@OrderDate,@OrderDate+7,@CustomerId,'CARGO TRANSPORT 5');

PRINT N' Sales Order ID : ' + STR(@SalesOrderId)

GO

**Que 2:**

--Code to insert an order detail

DECLARE @SalesOrderIdUser int = IDENT\_CURRENT('AdventureWorksLT2012.SalesLT.SalesOrderHeader'),

@ProductId int=760,

@Quantity int= 1,

@UnitPrice float= 782.99

IF EXISTS (select \* FROM AdventureWorksLT2012.SalesLT.SalesOrderHeader where SalesOrderID=@SalesOrderIdUser)

insert into AdventureWorksLT2012.SalesLT.SalesOrderDetail(SalesOrderID,ProductID,OrderQty,UnitPrice)

values(@SalesOrderIdUser,@ProductID,@Quantity,@UnitPrice)

ELSE

Print N'The order does not exist'

GO

DECLARE @SalesOrderIdUser int = 0,

@ProductId int=760,

@Quantity int= 1,

@UnitPrice float= 782.99

IF EXISTS (select \* FROM AdventureWorksLT2012.SalesLT.SalesOrderHeader where SalesOrderID=@SalesOrderIdUser)

insert into AdventureWorksLT2012.SalesLT.SalesOrderDetail(SalesOrderID,ProductID,OrderQty,UnitPrice)

values(@SalesOrderIdUser,@ProductID,@Quantity,@UnitPrice)

ELSE

Print N'The order does not exist'

GO

**Challenge 2: Updating Bike Prices**

**Que 1:**

--Write a while loop to update bike prices

DECLARE @MarketAverage FLOAT= 2000,

@MaximunAcceptablePrice FLOAT= 5000,

@AverageSellingPrice FLOAT,

@MaximumSellingPrice FLOAT;

WHILE(@AverageSellingPrice < @MarketAverage)

BEGIN

UPDATE AdventureWorksLT2012.SalesLT.Product

SET ListPrice = 1.1 \* ListPrice

SELECT @AverageSellingPrice = AVG(ListPrice)

FROM AdventureWorksLT2012.SalesLT.Product AS A

JOIN AdventureWorksLT2012.SalesLT.vGetAllCategories AS B

ON A.ProductCategoryID = B.ProductCategoryID

WHERE B.ParentProductCategoryName = 'Bikes'

SELECT @MaximumSellingPrice = MAX(ListPrice)

FROM AdventureWorksLT2012.SalesLT.Product AS A

JOIN AdventureWorksLT2012.SalesLT.vGetAllCategories AS B

ON A.ProductCategoryID = B.ProductCategoryID

WHERE B.ParentProductCategoryName = 'Bikes'

IF(@MaximumSellingPrice >= @MaximunAcceptablePrice)

BREAK

ELSE

CONTINUE

END