Section: 03

Student: Sheetal Singh

Challenge 1

Q1:

-- Write a code to insert order header

CREATE SEQUENCE SalesOrderNumb

START WITH 1

INCREMENT BY 1;

GO

use [AdventureWorksLT2012]

DECLARE @OrderDate datetime;

DECLARE @DueDate datetime;

DECLARE @CustomerID int = 1;

declare @OrderID int;

declare @OrderIDvalue int;

DECLARE @ShipMethod nvarchar(30) ='CARGO TRANSPORT 5';

SET IDENTITY\_INSERT [SalesLT].[SalesOrderHeader] ON

set @OrderID = NEXT VALUE FOR SalesOrderNumb ;

set @OrderIDvalue=@OrderID;

INSERT INTO [SalesLT].[SalesOrderHeader]

([SalesOrderID]

,[OrderDate]

,[DueDate]

,CustomerID

,[ShipMethod])

VALUES

(@OrderID

,GETDATE()

,DATEADD (day , 7 , GETDATE() )

,@CustomerID

,@ShipMethod)

PRINT @OrderID ;

Q2:

-- Write code to insert order detail

DECLARE @SalesOrderID int

DECLARE @ProductID int = 760;

DECLARE @Quantity int = 1;

DECLARE @UnitPrice money = 782.99;

SET IDENTITY\_INSERT [SalesLT].[SalesOrderHeader] ON

SET @SalesOrderID = @OrderIDvalue;

IF EXISTS (SELECT \* FROM SalesLT.SalesOrderHeader WHERE SalesOrderID = @SalesOrderID)

BEGIN

INSERT INTO SalesLT.SalesOrderDetail (SalesOrderID, OrderQty, ProductID, UnitPrice)

VALUES

(@SalesOrderID, @Quantity, @ProductID, @UnitPrice)

select \* from SalesLT.SalesOrderDetail

END

ELSE

BEGIN

PRINT 'The order does not exist'

END

Q2(2)

Write code to insert order detail(Test code with 0)

DECLARE @SalesOrderID int

DECLARE @ProductID int = 760;

DECLARE @Quantity int = 1;

DECLARE @UnitPrice money = 782.99;

SET IDENTITY\_INSERT [SalesLT].[SalesOrderHeader] ON

SET @SalesOrderID = 0;

IF EXISTS (SELECT \* FROM SalesLT.SalesOrderHeader WHERE SalesOrderID = @SalesOrderID)

BEGIN

INSERT INTO SalesLT.SalesOrderDetail (SalesOrderID, OrderQty, ProductID, UnitPrice)

VALUES

(@SalesOrderID, @Quantity, @ProductID, @UnitPrice)

select \* from SalesLT.SalesOrderDetail

END

ELSE

BEGIN

PRINT 'The order does not exist'

END

Challenge 2

Q1:

-- Write a while loop to update bike prices

DECLARE @MarketAverage money = 2000;

DECLARE @CustomerMax money = 5000;

DECLARE @MaxListPrice money;

DECLARE @AverageListPrice money;

SELECT @AverageListPrice = AVG([ListPrice]), @MaxListPrice= MAX([ListPrice])

FROM [SalesLT].[Product]

WHERE [ProductCategoryID]IN

(SELECT DISTINCT [ProductCategoryID]

FROM [SalesLT].[vGetAllCategories]

WHERE [ParentProductCategoryName] = 'Bikes');

WHILE @AverageListPrice < @MarketAverage

BEGIN

UPDATE [SalesLT].[Product]

SET [ListPrice] = [ListPrice] \* 1.1

WHERE [ProductCategoryID] IN

(SELECT DISTINCT [ProductCategoryID]

FROM [SalesLT].[vGetAllCategories]

WHERE [ParentProductCategoryName] = 'Bikes');

SELECT @AverageListPrice = AVG([ListPrice]), @MaxListPrice = MAX([ListPrice])

FROM [SalesLT].[Product]

WHERE [ProductCategoryID] IN

(SELECT DISTINCT [ProductCategoryID]

FROM [SalesLT].[vGetAllCategories]

WHERE [ParentProductCategoryName] = 'Bikes');

IF @MaxListPrice >= @CustomerMax

BREAK

ELSE

CONTINUE

END

PRINT 'New average bike price:' + CONVERT(varchar, @AverageListPrice);

PRINT 'New maximum bike price:' + CONVERT(varchar, @MaxListPrice);