**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**

**------------------------------**

A blue and white logo

Description automatically generated

**ĐỒ ÁN MÔN HỌC**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**Đề tài**

**QUẢN LÝ GIỎ HÀNG**

**Mục Lục**

**[LỜI MỞ ĐẦU 2](#_Toc16851)**

**[CHƯƠNG 1. QUẢN LÝ CƠ SỞ DỮ LIỆU](#_Toc9828)** [3](#_Toc9828)

[1.1 GIOI THIỆU CƠ SỞ DỮ LIỆU ADVANTUREWORK 3](#_Toc11143)

[Cơ sở dữ liệu Advanturesworks. 3](#_Toc24780)

[1.1.2. Mô tả Cơ sở dữ liệu ‘Quản lý giỏ hàng’ 5](#_Toc7006)

[1.1.3 Thông tin dữ liệu các bảng trong ‘Quản Lý Gio Hàng’ 5](#_Toc9012)

[1.1.4 Sơ đồ mối quan hệ 9](#_Toc29647)

**[CHƯƠNG 2: CÁC THAO TÁC TRÊN CƠ SỞ DỮ LIỆU](#_Toc21033)** [9](#_Toc21033)

[1. Tạo các view 9](#_Toc16536)

[1.1. Tạo 2 view có điều kiện đơn giản (trên 1 bảng, nhiều bảng) 10](#_Toc32707)

[2.1. Tạo view có điều kiện phức tạp 11](#_Toc28167)

[2.3. Tạo view phức tạo cập nhật dữ liệu 12](#_Toc32350)

[2. Tạo thủ tục 13](#_Toc24742)

[2.1 Tạo thủ tục không có tham số 13](#_Toc20954)

[2.2 Tạo một thủ tục có tham số mặc định 13](#_Toc71)

[2.3 Tạo thủ tục có tham số output 14](#_Toc20564)

[2.4. Tạo 2 thủ tục có tham số input: 14](#_Toc27106)

[3. Tạo các Function 15](#_Toc27910)

[3.1 Tạo 2 hàm trả về kiểu vô hướng 15](#_Toc28005)

[3.2 Tạo 2 function trả về table 16](#_Toc7929)

[3.3. Function trả về bảng tự đinh nghĩa 17](#_Toc19345)

[4. Các trigger 18](#_Toc16272)

[4.1. Tạo 1 trigger insert 18](#_Toc11695)

[4.2 Tạo 1 Trigger Update 19](#_Toc7858)

[4.3 Tạo 1 trigger Delete 19](#_Toc3193)

[4.4 Tạo 2 transaction 19](#_Toc26595)

[5. Tạo các user 21](#_Toc29140)

**[CHƯƠNG 3: KẾT LUẬN](#_Toc8225)** [22](#_Toc8225)

[3.1. Kết quả đã làm được: 22](#_Toc3790)

[3.2. Hạn chế (chưa làm được): 22](#_Toc31803)

[3.3. Thuận lợi và khó khăn: 22](#_Toc21766)

[-Thuận lợi 22](#_Toc16158)

[-Khó khăn 22](#_Toc23644)

**[TÀI LIỆU THAM KHẢO](#_Toc1862)** [22](#_Toc1862)

[Website: 22](#_Toc32519)

**LỜI MỞ ĐẦU**

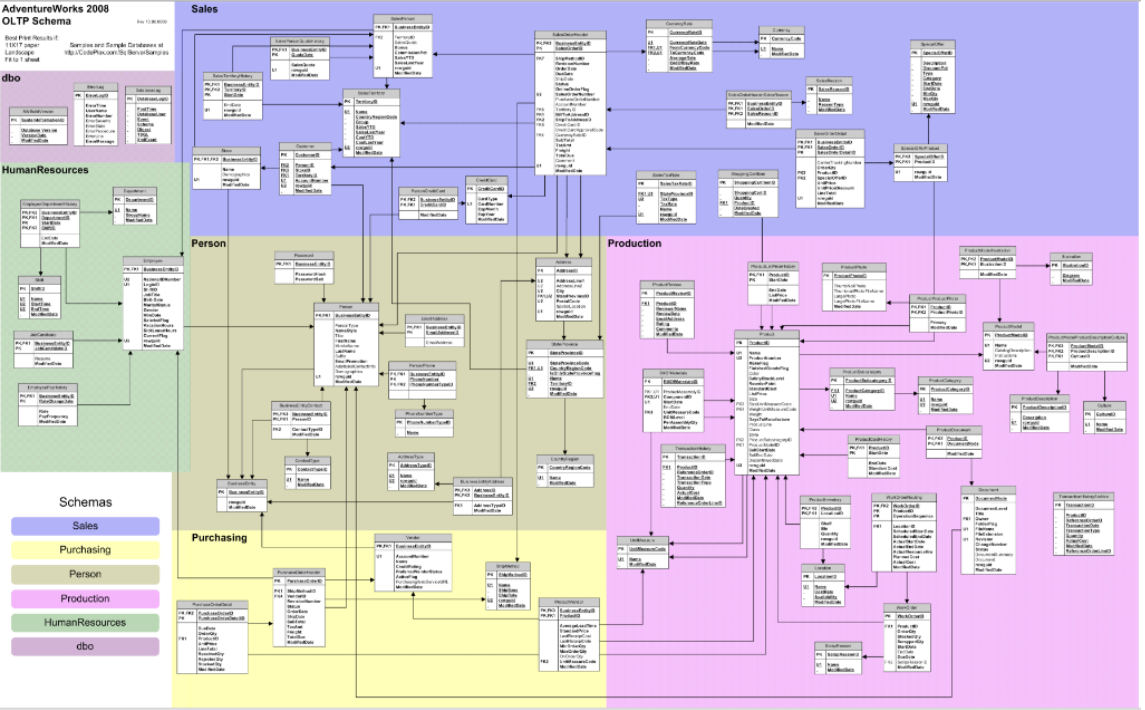
Với sự phát triển nhanh chóng của công nghệ, thế giới đường như ngày càng thu nhỏ nhờ mạng Internet. Để có thể đáp ứng được đòi hỏi trình độ công nghệ thông tin ngày càng cao của thị trường, sinh viên ngành kỹ thuật phần mềm nói riêng và các đối tượng hoạt động trong lĩnh vực công nghệ thông tin nói chung cần phải nắm được các kiến thức về công nghệ để có thể dễ dàng tiếp cận tiếp thu cũng như quản lí các thông tin. Một trong những cách để tổ chức cũng như quản lý dữ liệu là Database Management System(Hệ quản lý cơ sở dữ liệu).

Hệ quản lý cơ sở dữ liệu (*Database Management System - DBMS*) là phần mềm tương tác với người dùng cuối, ứng dụng và chính cơ sở dữ liệu để thu thập và phân tích dữ liệu. Phần mềm DBMS bao gồm các tiện ích cốt lõi được cung cấp để quản trị cơ sở dữ liệu. Tổng cộng của cơ sở dữ liệu, DBMS và các ứng dụng liên quan có thể được gọi là "hệ thống cơ sở dữ liệu". Thông thường thuật ngữ "cơ sở dữ liệu" cũng được sử dụng để nói đến bất kỳ DBMS, hệ thống cơ sở dữ liệu hoặc ứng dụng nào được liên kết với cơ sở dữ liệu.

**CHƯƠNG 1. QUẢN LÝ CƠ SỞ DỮ LIỆU**

* 1. GIOI THIỆU CƠ SỞ DỮ LIỆU ADVANTUREWORK

Cơ sở dữ liệu Advanturesworks.



***Hình 1.1\_ Advanture-work***

Cơ sở dữ liệu AdventureWorks là một cơ sở dữ liệu mẫu được Microsoft pháttriển với mục đích giúp người dùng hiểu rõ hơn về các công nghệ của Microsoft vàhọc cách sử dụng chúng để phát triển các ứng dụng. Cơ sở dữ liệu này được thiết kếcho SQL Server, nhưng cũng có thể được sử dụng với các hệ thống quản lý cơ sở dữliệu khác như Oracle, MySQL, PostgreSQL.

AdventureWorks được lấy bối cảnh trên một công ty hư cấu có tênAdventureWorks Cycles, một công ty sản xuất và kinh doanh xe đạp với quy mô lớn.Cơ sở dữ liệu này chứa các bảng, chế độ xem và thủ tục liên quan đến các hoạt độngquản lý khách hàng, sản phẩm, đặt hàng và các giao dịch khác, được sử dụng để minhhọa cho các đối tượng trong xử lý tác vụ (OLTP).

Cơ sở dữ liệu AdventureWorks cung cấp các chức năng phức tạp như xử lý dữliệu với giao dịch và kết hợp dữ liệu từ nhiều bảng khác nhau trong cơ sở dữ liệu.Ngoài ra, AdventureWorks còn có các tính năng đáng chú ý như sử dụng các ràngbuộc dữ liệu để bảo vệ tính toàn vẹn của dữ liệu, sử dụng các chỉ mục để tối ưu hóahiệu suất truy cập dữ liệu và sử dụng các trigger để tự động thực hiện các hành độngkhi dữ liệu được thêm, sửa hoặc xóa. Các tính năng này giúp AdventureWorks trởthành một cơ sở dữ liệu mẫu rất hữu ích cho các nhà phát triển và quản trị cơ sở dữ liệu.

Các tập dữ liệu trong AdventureWorks có thể được sử dụng để phân tích dữliệu, trực quan hóa và huấn luyện các mô hình dự đoán và phân loại. AdventureWorkscũng được sử dụng trong nhiều khóa học, chứng chỉ và đào tạo về quản trị cơ sở dữliệu, thiết kế cơ sở dữ liệu và phát triển ứng dụng.

**1.1.2. Mô tả Cơ sở dữ liệu ‘Quản lý giỏ hàng’**

**- Mô hình quan hệ**

+Bảng 1**:**

**Sales.SalesOrderHeader**(SalesOrderID,RevisionNumber,OrderDate,DueDate,ShipDa

te,Status,OnlineOrderFlag,SalesOrderNumber,PurchaseOrderNumber,AccountNumber

,CustomerID,SalesPersonID,TerritoryID,BillToAddressID,ShipToAddressID,ShipMet

hodID,CreditCardID,CreditCardApprovalCode,CurrencyRateID,SubTotal,TaxAmt,Fre

ight,TotalDue,Comment,Rowguid,ModifiedDate)

**+**Bảng 2:

**Sales.SalesOrderDetail**(SalesOrderID,SalesOrderDetailID,CarrierTrackingNumber,O

rderQty,ProductID,SpecialOfferID)

+Bảng 3:

**Production.Product**(ProductID,Name,ProductNumber,MakeFlag,FinishedGoodsFlag,

Color,SafetyStockLevel,ReorderPoint,StandardCost,ListPrice,Size,SizeUnitMeasureC

ode,WeightUnitMeasureCode,Weight,DaysToManufacture,ProductLine,Class,Style,Pr

oductSubcategoryID,ProductModelID,SellStartDate,SellEndDate,DiscontinuedDate,R

owguid,ModifiedDate)

+Bảng 4:

**Sales.ShoppingCartItem**(ShoppingCartItemID,ShoppingCartID,Quantity,ProductID,

DateCreated,ModifiedDate)

+Bảng 5:

**Sales.SpecialOfferProduct**(SpecialOfferID,ProductID,Rowguid,ModifiedDate)

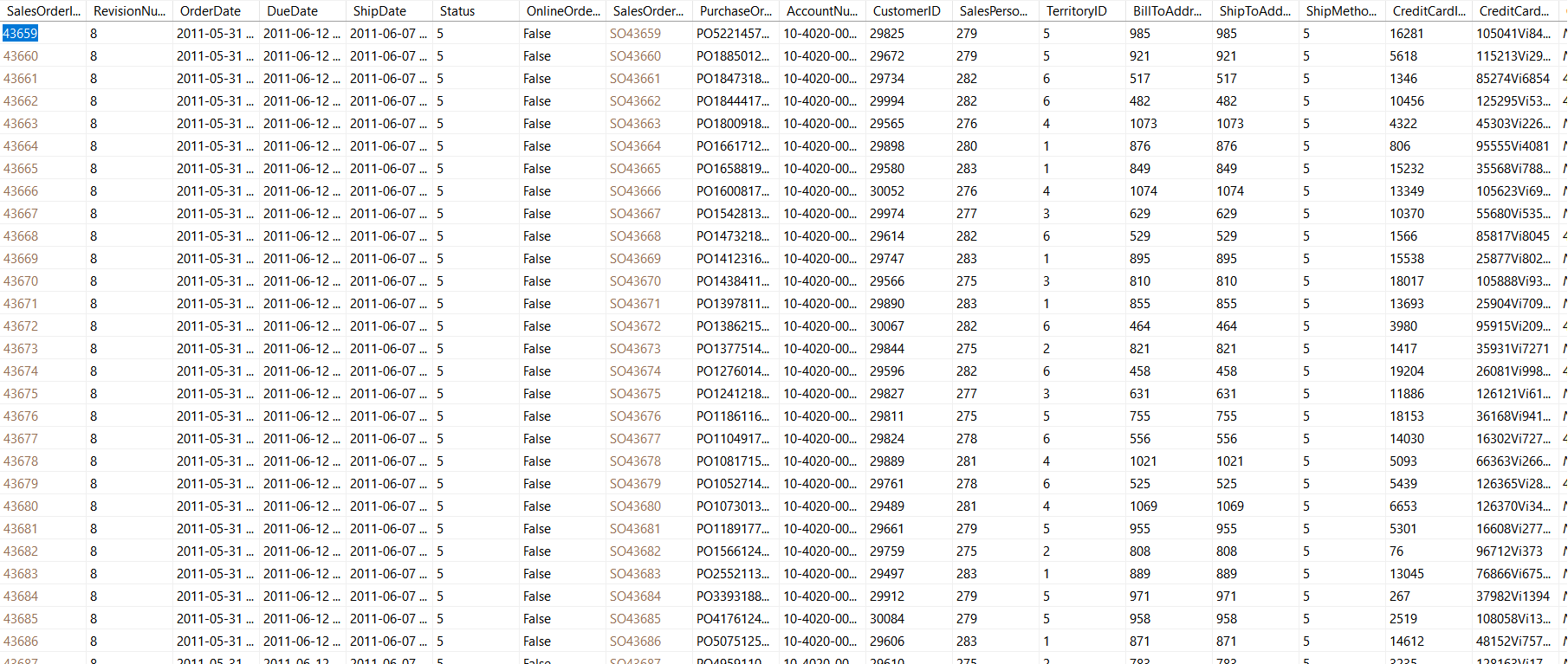
+Bảng 6:

**Sales.SpecialOffer**(SpecialOfferID,Description,DiscountPct,Type,Category,StartDate,

EndDate,MinQty,MaxQty,Rowguid,ModifiedDate)

**1.1.4** Thông tin dữ liệu có trong các bảng trong ‘**Quản Lý Giỏ Hàng’**

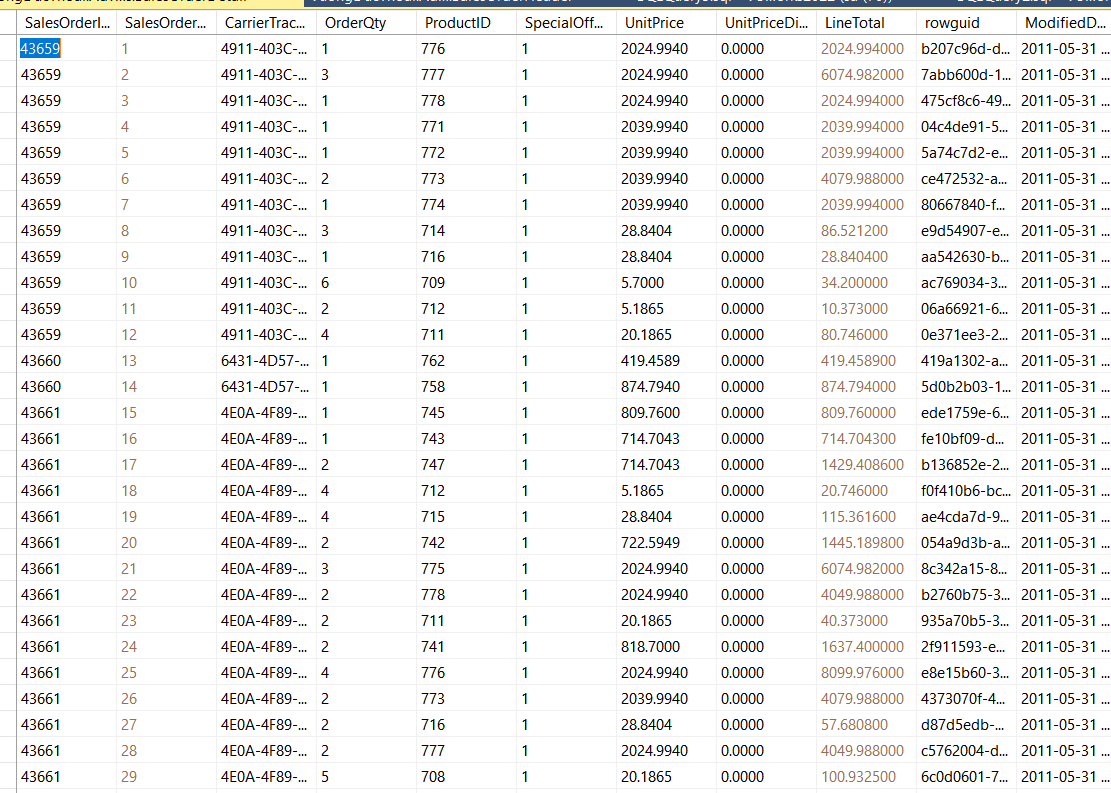
**1.1.4.1 Bảng Sales.SalesOrderHeader**



***Hình 1.2\_Sales.SalesOrderHeader***

Bảng SalesOrderHeader mô tả Thông tin chung về đơn đặt hàng.

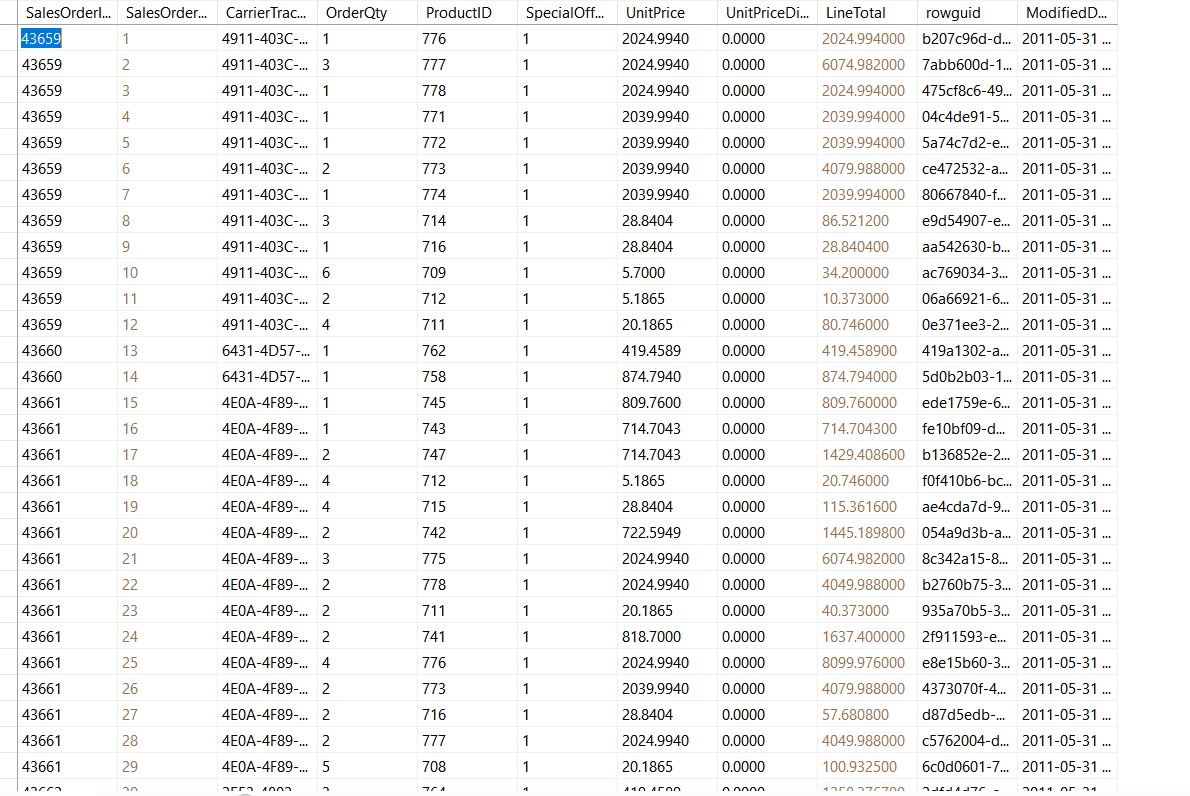
**1.1.4.2 Bảng Sales.SalesOrderDetail**



***Hình 1.3\_Sales.SalesOrderDetail***

Bảng SalesOrderDetail mô tả Các sản phẩm riêng lẻ được liên kết với một đơn đặt hàng cụ thể

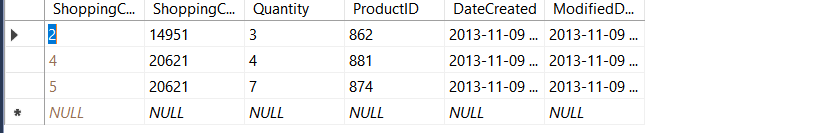
**1.1.4.3 Bảng Production.Product**



***Hình 1.4\_Production.Product***

Bảng Product mô tả sản phẩm đã bán hoặc được sử dụng để sản xuất các sản phẩm đã bán.

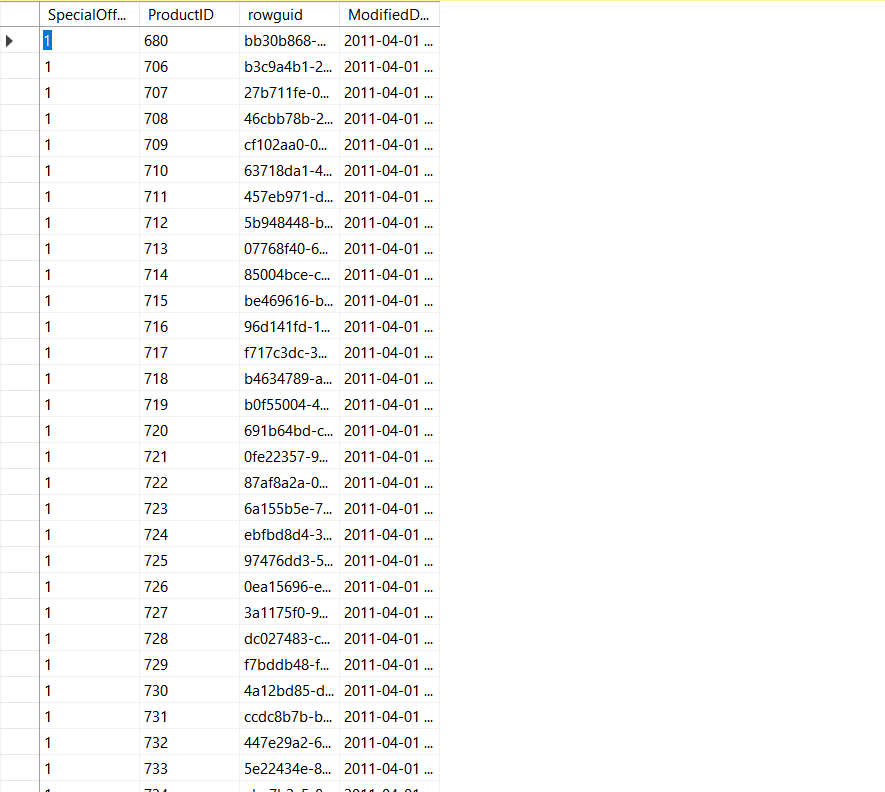
**1.1.4.4 Bảng Sales.ShoppingCartItem**



***Hình 1.5 \_Sales.ShoppingCartItem***

Bảng ShoppingCartItem mô tả các thông tin chi tiết về sản phẩm trong giỏ

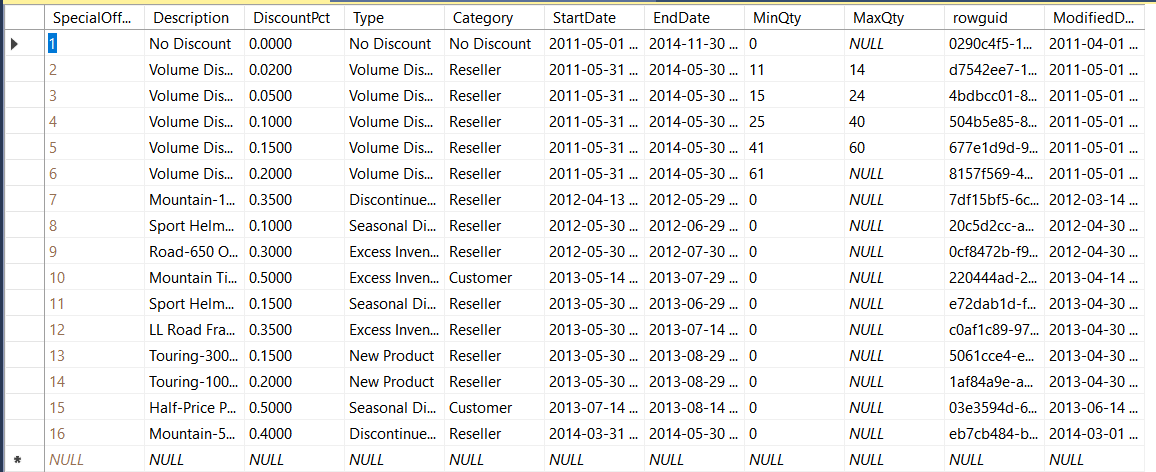
**1.1.4.5 Bảng Sales.SpecialOfferProduct**



***Hình 1.6 \_Sales.SpecialOfferProduct***

Bảng SpecialOfferProduct mô tả Các sản phẩm bản đồ bảng tham khảo chéo để giảm giá ưu đãi đặc biệt.

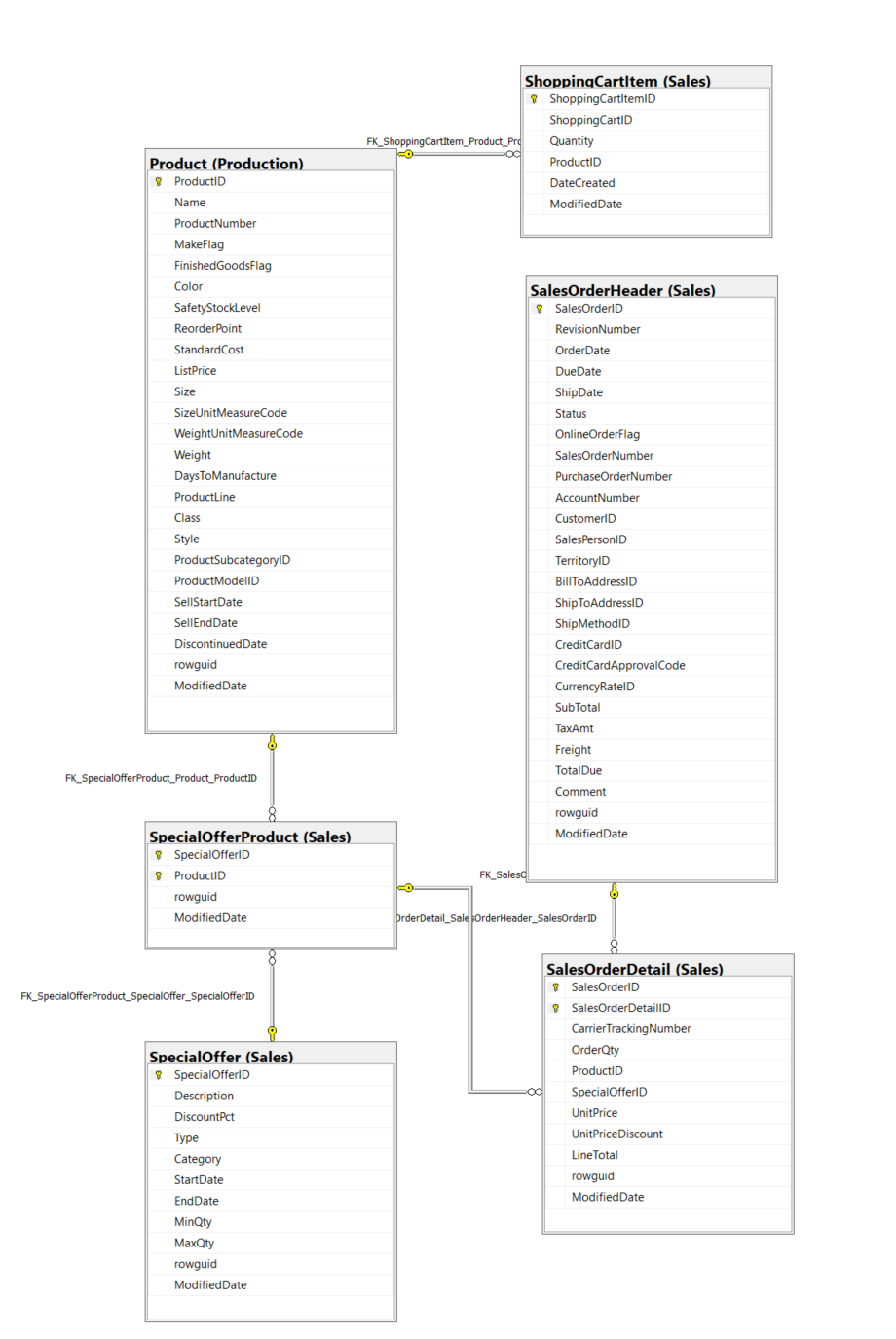
**1.1.3.6 Bảng Sales.SpecialOffer**



***Hình 1.7\_Sales.SpecialOffer***

Bảng Special mô tả tra cứu chiết khấu giảm giá.

**1.1.4 Sơ đồ mối quan hệ**



***Hình 1.8\_ModelRelationship***

**CHƯƠNG 2: CÁC THAO TÁC TRÊN CƠ SỞ DỮ LIỆU**

1. **Tạo các view**

**1.1.** Tạo 2 view có điều kiện đơn giản (trên 1 bảng, nhiều bảng)

**1.1.1.** View xuất ra hàng hóa có giá trị lớn hơn 3000

--**Tao View**

CREATE VIEW [dbo].[view\_HighPriceProducts]

AS

SELECT ProductID, Name, ListPrice

FROM Production.Product

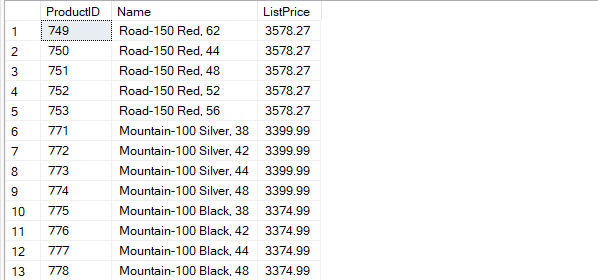
WHERE ListPrice > 3000

GO

--**Thuc thi View**

SELECT \* FROM view\_HighPriceProducts

--**Ket Qua**



***Hình 2.1\_[v\_HighPriceProducts]***

**1.1**.**2.** View xuất ra sản phẩm trong giỏ hàng

--Tao View

CREATE VIEW [dbo].[view\_ProductCartItem]

AS

SELECT Production.Product.ProductID,

Sales.ShoppingCartItem.ShoppingCartItemID, Production.Product.Name,

Production.Product.ListPrice, Sales.ShoppingCartItem.Quantity

FROM Production.Product INNER JOIN

Sales.ShoppingCartItem ON Production.Product.ProductID =

Sales.ShoppingCartItem.ProductID

--Thuc thi View

SELECT \* FROM view\_ProductCartItem

-- Ket Qua



***Hình 2.2 [v\_ProductCartItem]***

**2.1. Tạo view có điều kiện phức tạp**

**2.1.1.**  View xuất ra tồng tiền của tất cả các đơn hàng

-- Tạo View

CREATE VIEW [dbo].[view\_TotalSalesWithPromotion]

AS

SELECT Sales.SalesOrderDetail.SalesOrderID,

Sales.SpecialOfferProduct.ProductID,

CASE WHEN Sales.SpecialOffer.SpecialOfferID = '1' THEN

Sales.SalesOrderDetail.OrderQty \* Sales.SalesOrderDetail.UnitPrice ELSE

Sales.SalesOrderDetail.OrderQty \* Sales.SalesOrderDetail.UnitPrice \* (1 -

Sales.SpecialOffer.DiscountPct)

END AS 'Amount', Sales.SpecialOffer.Description

FROM Sales.SpecialOffer

INNER JOIN Sales.SpecialOfferProduct ON Sales.SpecialOffer.SpecialOfferID =

Sales.SpecialOfferProduct.SpecialOfferID

INNER JOIN Sales.SalesOrderDetail ON Sales.SpecialOfferProduct.SpecialOfferID =

Sales.SalesOrderDetail.SpecialOfferID AND

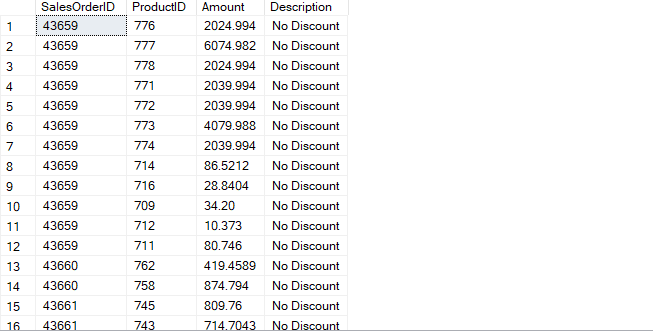
Sales.SpecialOfferProduct.ProductID = Sales.SalesOrderDetail.ProductID

GO

--Thuc thi

SELECT \* FROM view\_TotalSalesWithPromotion

-- Ket Qua



***Hình 2.3 [v\_TotalSalesWithPromotion]***

**2.1.2.** View xuất ra các sản phẩm được giảm giá từ 2012 trở đi và ProductLine là ‘S’)

--Tao View

CREATE VIEW [dbo].[RecentProductPromotions]

AS

SELECT Production.Product.ProductID, Production.Product.Name,

Production.Product.Color, Production.Product.ProductLine,

Sales.SpecialOfferProduct.SpecialOfferID,

Sales.SpecialOfferProduct.ModifiedDate

FROM Production.Product INNER JOIN

Sales.SpecialOfferProduct ON Production.Product.ProductID =

Sales.SpecialOfferProduct.ProductID

WHERE (Production.Product.Color IS NOT NULL) AND

(YEAR(Sales.SpecialOfferProduct.ModifiedDate) > '2012') AND

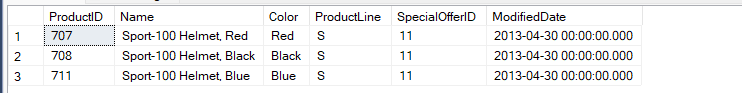
(Production.Product.ProductLine = 'S')

GO

-- Thuc thi

SELECT \* FROM RecentProductPromotions

-- Ket Qua



***Hình 2.4 [v\_RecentProductPromotions]***

**2.3. Tạo view phức tạo cập nhật dữ liệu**

**2.3.1** View cập nhật giá sản phẩm có mã sản phẩm là ‘776’

--Tao View

CREATE VIEW ComplexSalesView AS

SELECT

p.ProductID,

p.Name AS ProductName,

p.ProductNumber,

p.ListPrice,

p.StandardCost,

sod.SalesOrderDetailID,

sod.SalesOrderID,

sod.OrderQty,

sod.UnitPrice,

sod.LineTotal,

soh.OrderDate,

soh.ShipDate,

soh.SalesOrderNumber

FROM

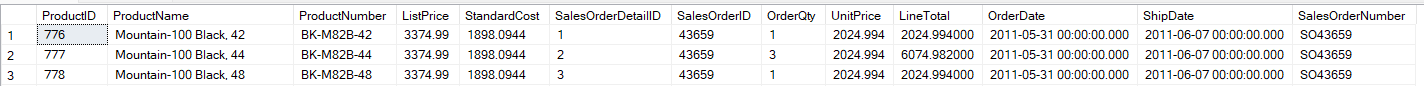
Production.Product p

INNER JOIN Sales.SalesOrderDetail sod ON p.ProductID = sod.ProductID

INNER JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID;

--Ket qua

SELECT \* FROM ComplexSalesView



***Hình 2.5 [v\_ComplexSalesView]***

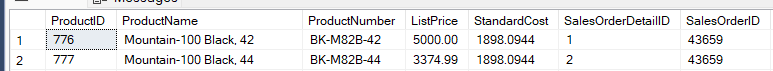
-- Cap nhat View

UPDATE ComplexSalesView

SET ListPrice = 5000

WHERE ProductID = 1001;

--Ket Qua



***Hình 2.6 [u\_CompleSalesView]***

1. **Tạo thủ tục**

**2.1** Tạo thủ tục không có tham số

**2.1.1.** Thủ tục hiển thị danh sách sản phẩm

--Tạo thủ tục

CREATE PROCEDURE GetAllProducts

AS

BEGIN

SELECT ProductID, Name, ProductNumber, ListPrice

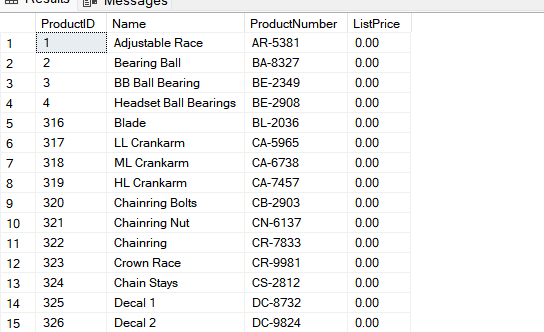
FROM Production.Product;

END;

--Thực thi

EXEC GetAllProducts

--Kết quả



***Hình 2.7 [pc\_GetAllProducts]***

**2.2** Tạo một thủ tục có tham số mặc định

**2.2.1.** Store proceduce hiển thị mã giảm giá (DiscountPct) cho khách hàng có tham số đầu vào là mã khách hàng

--Tạo thủ tục

CREATE PROCEDURE showDiscountCustomer @a nvarchar(50) = N'Customer'

AS

BEGIN

SELECT SpecialOfferID, Description, DiscountPct, Type, Category, StartDate,

EndDate, MinQty, MaxQty, rowguid, ModifiedDate

FROM Sales.SpecialOffer

WHERE (Category = @a)

END

--Kết quả

EXEC showDiscountCustomer

--Thực thi



***Hình 2.8 [pc\_showDiscountCustomer]***

**2.3** Tạo thủ tục có tham số output

**2.3.1.** Store proceduce hiển thị số lượng sản phẩm chưa đặt hàng

--Tạo Proceduce

CREATE PROCEDURE [show\_soLuongSanPhamChuaDatHang]

(

@EId int output

)

AS

BEGIN

SET NOCOUNT ON

SELECT @EId= (SELECT count (ShoppingCartItemID)

FROM Sales.ShoppingCartItem)

END

GO

--Thuc thi

declare @a INT

EXEC show\_soLuongSanPhamChuaDatHang @a OUTPUT

SELECT @a as N'Số Lượng'

--Ket qua



***Hình 2.9 [pc\_soLuongSanPhamChuaDatHang]***

**2.4**. Tạo 2 thủ tục có tham số input:

(Có thể xây dựng sau đó dùng Thủ tục để gọi hàm)

**2.4.1.** Store proceduce hiển thị thông tin của sản phẩm khi nhập mã sản phẩm.

--Tạo proceduce

CREATE PROC [show\_producewithMaSP] @productID int

AS

BEGIN

SELECT\*

FROM[Production].[Product]

WHERE(ProductID = @productID)

END

GO

--Thuc thi

EXEC show\_producewithMaSP @productID='316'

--Ket qua



***Hình 2.10 [pc\_proceducewithMaSP]***

**2.4.2.** Stored proceduce đếm số lượng sản phẩm có trong lượng nhập vào

--Tạo proceduce

CREATE PROCEDURE [CountProductByWeight](

@weight decimal(8,2)

)

AS

BEGIN

SELECT count( ProductID) as N'Số Lượng'

FROM Production.Product

WHERE(Weight = @weight)

END

GO

--Thực thi

exec CountProductByWeight @weight= 2.22

--Kết quả



***Hình 2.11 [Count\_ProductByWeight]***

1. **Tạo các Function**

**3.1** Tạo 2 hàm trả về kiểu vô hướng

**3.1.1** Function tính tổng số lượng sản phẩm trong giỏ hàng dựa vào mã sản phẩm.

--Tạo Function

CREATE FUNCTION dbo.fn\_GetTotalProductQuantityInCart (@ProductID INT)

RETURNS INT

AS

BEGIN

DECLARE @TotalQuantity INT;

SELECT @TotalQuantity = SUM(Quantity)

FROM Sales.ShoppingCartItem

WHERE ProductID = @ProductID;

RETURN ISNULL(@TotalQuantity, 0);

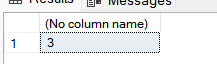
END;

GO

--Thuc thi

SELECT dbo.fn\_GetTotalProductQuantityInCart(862);

--Ket qua



***Hình 2.12 [fn\_GetTotalProductQuantityInCart]***

**3.1.2.** Function tính tổng tiền của đơn hàng được nhập có tham số đầu vào là mã sản phẩm.

--Tạo function

CREATE FUNCTION [dbo].[calculateTotalPrice]

(@salesOrderId int

)

RETURNS numeric(38,6)

AS

BEGIN

RETURN (select SUM (LineTotal) FROM Sales.SalesOrderDetail

WHERE (SalesOrderID = @salesOrderId))

END

GO

--Thực thi

SELECT [dbo].calculateTotalPrice(43664)

--Ket qua



***Hình 2.13 [fn\_calculateTotalPrice]***

**3.2** Tạo 2 function trả về table

**3.2.1.**  Function trả về thông tin bán hàng chi tiết cho một sản phẩm cụ thể

--Tạo function

CREATE FUNCTION dbo.fn\_GetProductSalesDetails (@ProductID INT)

RETURNS TABLE

AS

RETURN

(

SELECT

P.ProductID,

P.Name AS ProductName,

SOD.SalesOrderID,

SOD.OrderQty,

SOD.UnitPrice,

SOD.UnitPriceDiscount,

(SOD.UnitPrice \* SOD.OrderQty) AS TotalPrice,

(SOD.UnitPriceDiscount \* SOD.OrderQty) AS TotalDiscount,

SO.SpecialOfferID,

SO.Description AS SpecialOfferDescription,

SOH.OrderDate,

SOH.DueDate,

SOH.ShipDate,

SOH.SubTotal

FROM

Production.Product P

INNER JOIN

Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID

INNER JOIN

Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID

LEFT JOIN

Sales.SpecialOfferProduct SOP ON SOD.ProductID = SOP.ProductID

LEFT JOIN

Sales.SpecialOffer SO ON SOP.SpecialOfferID = SO.SpecialOfferID

WHERE

P.ProductID = @ProductID

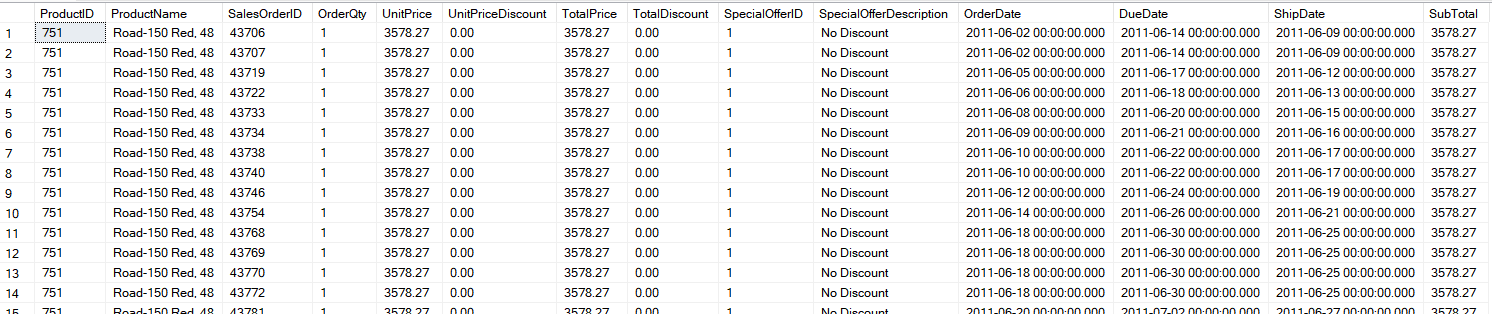
);

GO

--Thực thi

SELECT \* FROM dbo.fn\_GetProductSalesDetails(751)

--Kết quả



***Hình 2.14 [fn\_TotalSalesWithPromotion]***

**3.2.2.** Function xuất thông tin sản phẩm khi vào mã sản phẩm

--Code function

CREATE FUNCTION show\_ProductByProductId (@productId int)

RETURNS TABLE

AS

RETURN

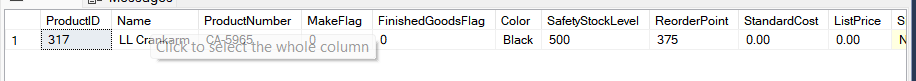
(SELECT \*FROM Production.Product WHERE (ProductID = @productId))

GO

--Thực thi

select \* from [show\_ProductByProductId] ('317')

--Kết quả



***Hình 2.15 [fn\_showProductByProductId]***

**3.3.**  Function trả về bảng tự đinh nghĩa

--Tạo Function

CREATE FUNCTION show\_productByStyle (@Style char )

RETURNS @bangtam TABLE (ProductID int,Name nvarchar(50))

AS

BEGIN

if (@Style = 'M')

INSERT INTO @bangtam(ProductID,Name)

SELECT ProductID, Name

FROM Production.Product

WHERE Style = 'M'

if (@Style = 'W')

INSERT INTO @bangtam(ProductID,Name)

SELECT ProductID, Name

From Production.Product

WHERE Style = 'W'

if (@Style = 'U')

INSERT INTO @bangtam(ProductID,Name)

SELECT ProductID, Name

FROM Production.Product

WHERE Style = 'U'

RETURN

END

GO

--Thực thi

SELECT \* FROM [show\_productByStyle] ('M')

--Kết quả



***Hình 2.16 [fn\_productByStyle]***

1. **Các trigger**

**4.1.** Tạo 1 trigger insert

Trigger kiểm tra việc thêm một thông tin vào bảng **Sales.SpecialOffer** thì

‘Chiết khẩu tối thiểu nhỏ hơn chiết khâu tối đa’

--Code trigger

CREATE TRIGGER tg1

ON Sales.SpecialOffer

FOR INSERT

AS

BEGIN

IF (( SELECT MinQty FROM INSERTED ) > ( SELECT MaxQty FROM

INSERTED ))

Begin

Print N'Chiết khấu tổi thiểu phải nhỏ hơn chiết khấu tối đa'

rollback transaction

END

END

GO

--Thực thi trigger

INSERT INTO Sales.SpecialOffer

VALUES(

N'abc',

DEFAULT,

N'',

N'',

GETDATE(),

GETDATE(),

100,

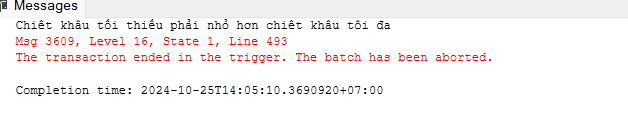
10,

DEFAULT,

DEFAULT

)

--Kết quả



***Hình 2.17\_[trg\_tg1]***

**4.2** Tạo 1 Trigger Update

Kiểm tra nếu số lượng sản phẩm có nhỏ hơn bằng 0 hay không trong giỏ hàng

CREATE TRIGGER tg2

ON Sales.ShoppingCartItem

FOR Update

AS

BEGIN

IF (( SELECT Quantity FROM INSERTED ) <= 0)

Begin

Print N'Số hàng phải lớn hơn 0'

rollback transaction

end

END

**4.3** Tạo 1 trigger Delete

Trigger kiểm tra xóa 1 thông tin trên bảng SpecialOffer

CREATE TRIGGER tg3

ON Sales.SpecialOffer

FOR Delete

AS

BEGIN

IF (( SELECT SpecialOfferID FROM deleted ) like (SELECT

SpecialOfferID FROM Sales.SpecialOfferProduct))

Begin

Print N'Không xóa được'

rollback transaction

end

END

**4.4** Tạo 2 transaction

**4.4**.**1.** Transaction khi thêm dữ liệu vào bảng thì ‘ngày kết thúc không được trước ngày hiện tại’

CREATE TRIGGER [transAddData]

ON Sales.SpecialOffer

FOR INSERT

AS

BEGIN

IF (( SELECT EndDate FROM INSERTED ) < GETDATE())

Begin

Print N'ngày kết thúc không được trước ngày hiện tại'

rollback transaction

end

END

--Thực thi transaction

INSERT INTO Sales.SpecialOffer

VALUES(

N'abc',

DEFAULT,

N'',

N'',

GETDATE(),

GETDATE(),

0,

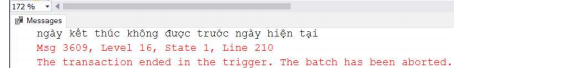
NULL,

DEFAULT,

DEFAULT

)

--Kết quả



***Hình 2.18 [trg\_transAddData]***

**4.4.2.** Transaction sau khi thêm dữ liệu vào bảng thì sẽ hiển thị ‘Số lượng sản phẩm hiện tại’

CREATE TRIGGER [tran2]

ON Production.Product

FOR INSERT

AS

BEGIN

BEGIN TRAN

SELECT (SELECT COUNT(ProductID) FROM

Production.Product)-(SELECT COUNT(ProductID) FROM

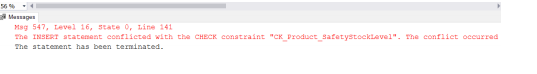
INSERTED)AS N'Số lượng sản phẩm hiện có'

COMMIT TRAN

SELECT COUNT(ProductID) AS N'Số lượng sản phẩm hiện tại' FROM

Production.Product

END



***Hình 2.19 [trg\_trans2]***

**5. Tạo các user**

**-** Tạo 2 user

User 1:

CREATE LOGIN [FormMain] WITH PASSWORD=N' ·åé\_áËß [Ø]N\_ \_DKü;   

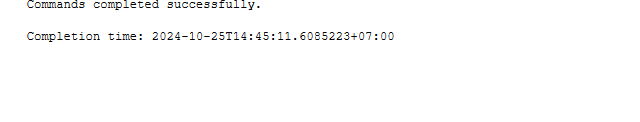
%ßÜÛi<\_±CDòá', DEFAULT\_DATABASE=[AdventureWorks2022],

DEFAULT\_LANGUAGE=[us\_english], CHECK\_EXPIRATION=ON,

CHECK\_POLICY=ON

GO

Create user VietAnh for login FormMain



***Hình 2.20 [user1]***

User 2:

CREATE LOGIN [MainForm] WITH PASSWORD=N' ·åé\_áËß [Ø]N\_ \_DKü;   

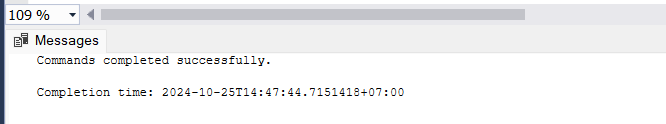
%ßÜÛi<\_±CDòá', DEFAULT\_DATABASE=[AdventureWorks2022],

DEFAULT\_LANGUAGE=[us\_english], CHECK\_EXPIRATION=ON,

CHECK\_POLICY=ON

GO

Create user AnhViet for login MainForm



***Hình 2.22 [trg\_transAddData]***

**CHƯƠNG 3: KẾT LUẬN**

**3.1. Kết quả đã làm được:**

Qua nghiên cứu đề tài “*Quản lý giỏ hàng*” cũng như trong quá trình làm việc nhóm và dưới sự hỗ trợ nhiệt tình của thầy Nguyễn Thế Bảo, nhóm chúng em đã hoàn thành tốt bài tiểu luận môn học Hệ quản trị cơ sở dữ liệu:

+Nắm bắt được những kiến thức cơ bản của hệ quản trị cơ dữ liệu

+Cơ bản nắm được các bước xây dựng một cơ sở quản lý giỏ hàng, thực hiện đúng

quy trình

**3.2. Hạn chế (chưa làm được):**

Đề tài vẫn còn rất nhiều thiếu sót chưa được hoàn thiện do thời gian và trình độ

chuyên môn cũng như kinh nghiệm thực tế của nhóm chưa cao nên có nhiều chỗ đúng

với thực tế hay đảm bảo tính đúng đắn của hệ thống. Chúng em rất mong nhận được sự

chỉ bảo,đánh giá của các thầy cô để có thể phát triển mô hình ngày càng hoàn thiện

hơn trong tương lai.

**3.3. Thuận lợi và khó khăn:**

**-Thuận lợi**

+Được Giảng viên có trình độ chuyên môn sâu nhiều kinh nghiệm tận tình trao đổi

giúp đỡ cho chúng em.

+Có cơ sở vật chất tốt tạo điều cho chúng em có thể hoàn thành tốt đề tài của

minh.Nguồn Internet và thư viện kham khảo với nguồn tài liệu đầy phong phú.

-**Khó khăn**

+Do trình độ chuyên môn chưa cao nên có nhiều phần thiếu xót và chưa làm được

như user đăng nhập,giao diện chưa đượclinh hoạt,…

+Ngôn ngữ và phần mềm soạn thảo còn mới mẻ nên nhiều chức năng chưa được

vận dụng và kiểm soát đượ

**TÀI LIỆU THAM KHẢO**

**Website:**

1. Gioi thiệu về VIEW trong SQL Server - viblo.asia

2. FUNCTION (Hàm) trong SQL Server - QuanTriMang.com

3. Sử dụng trigger trong SQL qua ví dụ cơ bản. (viblo.asia)

4. TRANSACTION trong SQL - QuanTriMang.com

5. Gioi thiệu cơ sở dữ liệu AdventureWorks - hocviendaotao.com