Module 2. Thao tác dữ liệu  
  
Mục tiêu: (Phần này SV tự thực hành ở nhà, nộp bài làm trong buổi học kế tiếp)  
 Ôn lại các cấu trúc câu lệnh select có lệnh nhóm, subquery  
 Thực hiện các loại merge Query, Pivot query  
I) Câu lệnh SELECT sử dụng các hàm thống kê với các mệnh đề Group by và  
Having  
1) Liệt kê danh sách các hóa đơn (SalesOrderID) lặp trong tháng 6 năm 2008 có  
tổng tiền >70000, thông tin gồm SalesOrderID, Orderdate, SubTotal, trong đó  
SubTotal =sum(OrderQty\*UnitPrice).  
  
2) Đếm tổng số khách hàng và tổng tiền của những khách hàng thuộc các quốc gia  
có mã vùng là US (lấy thông tin từ các bảng SalesTerritory, Sales.Customer,  
Sales.SalesOrderHeader, Sales.SalesOrderDetail). Thông tin bao gồm  
TerritoryID, tổng số khách hàng (countofCus), tổng tiền (Subtotal) với Subtotal

= SUM(OrderQty\*UnitPrice)  
3) Tính tổng trị giá của những hóa đơn với Mã theo dõi giao hàng  
(CarrierTrackingNumber) có 3 ký tự đầu là 4BD, thông tin bao gồm  
SalesOrderID, CarrierTrackingNumber, SubTotal=sum(OrderQty\*UnitPrice)  
4) Liệt kê các sản phẩm (product) có đơn giá (unitPrice)<25 và số lượng bán trung  
bình >5, thông tin gồm ProductID, name, AverageofQty  
5) Liệt kê các công việc (JobTitle) có tổng số nhân viên >20 người, thông tin gồm  
JobTitle, countofPerson=count(\*)  
6) Tính tổng số lượng và tổng trị giá của các sản phẩm do các nhà cung cấp có tên  
  
kết thúc bằng ‘Bicycles’ và tổng trị giá >800000, thông tin gồm  
BusinessEntityID, Vendor\_name, ProductID, sumofQty, SubTotal  
(sử dụng các bảng [Purchasing].[Vendor] [Purchasing].[PurchaseOrderHeader] và  
[Purchasing].[PurchaseOrderDetail])  
  
7) Liệt kê các sản phẩm có trên 500 đơn đặt hàng trong quí 1 năm 2008 và có tổng  
trị giá >10000, thông tin gồm ProductID, Product\_name, countofOrderID và  
Subtotal  
  
  
8) Liệt kê danh sách các khách hàng có trên 25 hóa đơn đặt hàng từ năm 2007 đến  
2008, thông tin gồm mã khách (PersonID) , họ tên (FirstName +' '+ LastName  
as fullname), Số hóa đơn (CountOfOrders).  
9) Liệt kê những sản phẩm có tên bắt đầu với ‘Bike’ và ‘Sport’ có tổng số lượng  
bán trong mỗi mỗi năm trên 500 sản phẩm, thông tin gồm ProductID, Name,  
CountofOrderQty, year. (dữ liệu lấy từ các bảng Sales.SalesOrderHeader,  
Sales.SalesOrderDetail, and Production.Product)  
10) Liệt kê những phòng ban có lương (Rate: lương theo giờ) trung bình >30, thông  
tin gồm Mã phòng ban (DepartmentID), tên phòng ban (name), Lương trung bình  
(AvgofRate). Dữ liệu từ các bảng [HumanResources].[Department],

[HumanResources].[EmployeeDepartmentHistory],  
[HumanResources].[EmployeePayHistory].  
II) Subquery  
1) Liệt kê các sản phẩm gồm các thông tin product names và product ID có trên  
100 đơn đặt hàng trong tháng 7 năm 2008  
  
2) Liệt kê các sản phẩm (ProductID, name) có số hóa đơn đặt hàng nhiều nhất trong  
tháng 7/2008  
3) Hiển thị thông tin của khách hàng có số đơn đặt hàng nhiều nhất, thông tin gồm:  
CustomerID, Name, CountofOrder  
4) Liệt kê các sản phẩm (ProductID, Name) thuộc mô hình sản phẩm áo dài tay với  
tên bắt đầu với “Long-Sleeve Logo Jersey”, dùng phép IN và EXISTS, (sử dụng  
bảng Production.Product và Production.ProductModel  
  
  
5) Tìm các mô hình sản phẩm (ProductModelID) mà giá niêm yết (list price) tối đa  
cao hơn giá trung bình của tất cả các mô hình.  
6) Liệt kê các sản phẩm gồm các thông tin ProductID, Name, có tổng số lượng đặt  
hàng >5000 (dùng In, exists)  
7) Liệt kê những sản phẩm (ProductID, UnitPrice) có đơn giá (UnitPrice) cao nhất  
trong bảng Sales.SalesOrderDetail  
8) Liệt kê các sản phầm không có đơn đặt hàng nào thông tin gồm ProductID,  
Name, dùng 3 cách Not in, not exists và left join.  
9) Liệt kê các nhân viên không lập hóa đơn từ sau ngày 1/5/2008, thông tin gồm  
EmployeeID, FirstName, LastName (dữ liệu từ 2 bảng HR.Employees và  
Sales.Orders)  
10) Liệt kê danh sách các khách hàng (customerID, name) có hóa đơn dặt hàng trong  
năm 2007 nhưng có hóa đơn đặt hàng trong năm 2008.  
III) Các loại truy vấn Union, Merge, Pivot  
1) Dùng UNION hiển thị thông tin ModifiedDate của Person.Person và HireDate

của HumanResources.Employee.  
2) Sử dụng Merge Query, thực hiện các yêu cầu sau:  
a) Tạo bảng một bảng mới tên MyOrder, gồm các thuộc tính PurchaseOrderID,  
EmployeeID, VendorID, ShipMethodID, TotalDue, RevisionNumber, status.  
b) Dùng lệnh Dùng lệnh Select into…chèn dữ liệu vào bảng, lấy dữ liệu từ bảng  
Purchasing.PurchaseOrderHeader.  
c) Xóa tất cả các dòng dữ liệu có PurchaseOrderID khác 1 or 2  
d) Cập nhật giá trị trong cột ToTalDue =1000 cho các dòng có trong bảng  
MyOrder.  
e) Dùng chức năng Merge để cập nhập dữ liệu cho cột ToTalDue của bảng  
MyOrder bằng ToTaldue trong bảng Purchasing.PurchaseOrderHeader cho  
những dòng có dữ liệu trùng nhau trong 2 bảng, ngược lại thì chèn các dòng  
khác nhau vào bảng MyOrder.  
3) Tạo một bảng mới tên UpdatedInventory chứa tổng số lượng tồn kho  
(QtyInventory) của những sản phẩm có ProductID>=900, theo cấu trúc sau:  
CREATE TABLE UpdatedInventory  
  
  
(  
ProductID int not null,  
QtyInventory int,  
CONSTRAINT PK\_Inventory PRIMARY KEY (ProductID)  
)  
 Chèn dữ liệu cho bảng UpdatedInventory lấy dữ liệu từ bảng  
ProductInventory.  
 Tạo một Merger query dùng để cập nhật số lượng tồn kho (InventoryQty) dựa  
trên dữ liệu nguồn là tổng số lượng (SumofQty) sản phẩm đã bán trong tháng 6  
năm 2008. Nếu những sản phẩm này tồn tại trong bảng UpdatedInventory và  
InventoryQty> SumofQty thì cập nhật lại số lượng tồn InventoryQty =  
InventoryQty - SumofQty, ngược lại thì xóa sản phẩm đó.

 Sau khi thực hiện Merger query:  
4) Pivot Query  
a) Tạo bảng dbo.orders có cấu trúc sau  
CREATE TABLE dbo.Orders  
(  
orderid INT NOT NULL,  
orderdate DATE NOT NULL,  
empid INT NOT NULL,  
custid VARCHAR(5) NOT NULL,  
qty INT NOT NULL,  
CONSTRAINT PK\_Orders PRIMARY KEY(orderid)  
)  
  
Chèn dữ liệu vào bảng  
(30001,  
(10001,  
(10005,  
(40001,  
(10006,  
(20001,  
(40005,  
(20002,  
(30003,  
(30004,  
(30007,  
  
'20070802',  
'20071224',  
'20071224',

'20080109',  
'20080118',  
'20080212',  
'20090212',  
'20090216',  
'20090418',  
'20070418',  
'20090907',  
  
3,  
2,  
1,  
2,  
1,  
2,  
3,  
1,  
2,  
3,  
3,  
  
'A',  
'A',  
'B',  
'A',  
'C',  
'B',  
'A',  
'C',  
'B',

'C',  
'D',  
  
10),  
12),  
20),  
40),  
14),  
12),  
10),  
20),  
15),  
22),  
30)  
  
  
b) Tính tổng Qty cho mỗi nhân viên. Thông tin gồm empid, custid  
c) Tạo bảng Pivot có dạng sau  
  
Gợi ý:  
SELECT empid, A, B, C, D  
FROM (SELECT empid, custid, qty  
FROM dbo.Orders) AS D  
PIVOT(SUM(qty) FOR custid IN(A, B, C, D)) AS P;  
  
5) Tạo 1 query lấy dữ liệu từ bảng dbo.Orders trả về số hóa đơn đã lập của nhân  
viên employee trong mỗi năm.  
 Tạo bảng pivot hiển thị số đơn đặt hàng được thực hiện bởi nhân viên có mã 164,  
198, 223, 231, and 233 nhóm theo từng mã người bán (vendor ID).  
6) Tạo một Pivot query thống kê tổng số lượng của những sản phẩm có mã

([ProductID]) >900 đã bán trong các năm 2006, 2007, 2008. Dữ liệu lấy từ bảng  
[Sales].[SalesOrderDetail] và [Sales].[SalesOrderHeader]. Kết quả có dạng  
  
  
Module 3. Hiện thực các ràng buộc toàn vẹn dữ liệu -  
  
Mục tiêu:  
 Thực hiện các ràng buộc toàn vẹn dữ liệu: Primary key, foreign key, domain,  
check, default.  
 Tìm hiểu cascading constraint trong thao tác update và delete  
  
1) Tạo hai bảng mới trong cơ sở dữ liệu AdventureWorks2008 theo cấu trúc sau:  
  
create table MyDepartment  
(  
DepID smallint not null primary key,  
DepName nvarchar(50),  
GrpName nvarchar(50)  
)  
create table MyEmployee (  
EmpID int not null primary key,  
FrstName nvarchar(50),  
MidName nvarchar(50),  
LstName nvarchar(50),  
DepID smallint not null foreign key references  
MyDepartment(DepID)  
)  
2) Dùng lệnh insert select from  
chèn

dữ  
  
liệu  
  
cho  
  
bảng  
  
MyDepartment,  
  
lấy  
  
dữ  
  
liệu  
  
từ  
  
bảng  
  
[HumanResources].[Department].  
  
3) Tương tự câu 2, chèn 20 dòng dữ liệu cho bảng MyEmployee lấy dữ liệu từ 2  
bảng  
[Person].[Person]và  
  
[HumanResources].[EmployeeDepartmentHistory]  
4) Dùng lệnh delete xóa 1 record trong bảng MyDepartment với DepID=1, có thực  
hiện được không? Vì sao?  
5) Thêm một default constraint vào field DepID trong bảng MyEmployee, với

giá trị mặc định là 1.  
  
  
6) Nhập thêm một record mới trong bảng MyEmployee, theo cú pháp sau:  
insert  
  
into  
  
MyEmployee  
  
(EmpID,  
  
FrstName,  
  
MidName,  
  
LstName) values(1, 'Nguyen',’Nhat’,'Nam'), quan sát giá trị  
trong field depID của record mới thêm.  
7) Xóa foreign key constraint trong bảng MyEmployee, thiết lập lại khóa ngoại  
DepID tham chiếu đến DepID của bảng MyDepartment với thuộc tính on delete  
set default.  
8) Xóa một record trong bảng MyDepartment có DepID=7, quan sát kết quả trong  
hai bảng MyEmployee và Mydepartment  
9) Xóa foreign key trong bảng MyEmployee hiệu chỉnh ràng buộc khóa ngoại  
DepID trong bảng MyEmployee, thiết lập thuộc tính on delete cascading và on  
update cascading  
10) Thực hiện xóa một record trong bảng MyDepartment với DepID =3, có thực  
hiện được không?  
11) Thêm ràng buộc check vào bảng MyDepartment tại field GrpName, chỉ cho phép  
nhận thêm những Department thuộc group Manufacturing

12) Thêm ràng buộc check vào bảng [HumanResources].[Employee], tại cột  
Birthday, chỉ cho phép nhập thêm nhân viên mới có tuổi từ 18 đến 60  
  
Module 4. View  
  
Mục tiêu:  
 Tạo view, thao tác trên view  
 Tìm hiểu các thuộc tính của view  
  
1) Tạo view dbo.vw\_Products hiển thị danh sách các sản phẩm từ bảng  
Production.Product và bảng Production.ProductCostHistory. Thông tin bao gồm  
ProductID, Name, Color, Size, Style, StandardCost, EndDate, StartDate  
  
  
2) Tạo view List\_Product\_view chứa danh sách các sản phẩm có trên 500 đơn đặt  
hàng trong quí 1 năm 2008 và có tổng trị giá >10000, thông tin gồm ProductID,  
Product\_name, countofOrderID và Subtotal.  
  
3) Tạo view dbo.vw\_CustomerTotals hiển thị tổng tiền bán được (total sales) từ cột  
TotalDue của mỗi khách hàng (customer) theo tháng và theo năm. Thông tin gồm  
CustomerID, YEAR(OrderDate) AS OrderYear, MONTH(OrderDate) AS  
OrderMonth, SUM(TotalDue).  
  
4) Tạo view trả về tổng số lượng sản phẩm (total quantity) bán được của mỗi nhân  
viên theo từng năm. Thông tin gồm SalesPersonID, OrderYear, sumOfOrderQty  
  
5) Tạo view ListCustomer\_view chứa danh sách các khách hàng có trên 25 hóa đơn  
đặt hàng từ năm 2007 đến 2008, thông tin gồm mã khách (PersonID), họ tên  
(FirstName +' '+ LastName as fullname), Số hóa đơn (CountOfOrders).

6) Tạo view ListProduct\_view chứa danh sách những sản phẩm có tên bắt đầu với  
‘Bike’ và ‘Sport’ có tổng số lượng bán trong mỗi mỗi năm trên 500 sản phẩm,  
thông tin gồm ProductID, Name, CountofOrderQty, year. (dữ liệu lấy từ các  
bảng  
  
Sales.SalesOrderHeader,  
  
Sales.SalesOrderDetail,  
  
and  
  
Production.Product)  
  
7) Tạo view List\_department\_View chứa danh sách các phòng ban có lương (Rate:  
lương theo giờ) trung bình >30, thông tin gồm Mã phòng ban (DepartmentID),  
tên phòng ban (name), Lương trung bình (AvgofRate). Dữ liệu từ các bảng  
[HumanResources].[Department],  
[HumanResources].[EmployeeDepartmentHistory],  
[HumanResources].[EmployeePayHistory].  
  
8) Tạo view Sales.vw\_OrderSummary với từ khóa WITH ENCRYPTION gồm  
orderYear (năm của ngày lập), OrderMonth (tháng của ngày lập), OrderTotal  
(tổng tiền). Sau đó xem thông tin và trợ giúp về mã lệnh của view này  
  
9) Tạo view Production.vwProducts với từ khóa WITH SCHEMABINDING  
gồm ProductID, Name, StartDate,EndDate,ListPrice của bảng Product và bảng  
ProductCostHistory. Xem thông tin của View. Xóa cột ListPrice của bảng  
Product. Có xóa được không? Vì sao?

10) Tạo view view\_Department với từ khóa WITH CHECK OPTION chỉ chứa các  
phòng thuộc nhóm có tên (GroupName) là “Manufacturing” và “Quality  
Assurance”, thông tin gồm: DepartmentID, Name, GroupName.  
a. Chèn thêm một phòng ban mới thuộc nhóm không thuộc hai nhóm  
“Manufacturing” và “Quality Assurance” thông qua view vừa tạo. Có  
chèn được không? Giải thích  
b. Chèn thêm một phòng mới thuộc nhóm “Manufacturing” và một phòng  
thuộc nhóm “Quality Assurance”.  
c. Dùng câu lệnh select xem kết quả trong bảng Department.  
  
Module 5. Fuction stored procedure  
  
Mục tiêu:  
Hiểu và biết cách lập trình trong SQL  
 Viết các batch  
Tạo và thực thi các loại function và stored procedure  
 Function gồm 3 loại  
o Scalar function  
o Table value Function  
o MultiStatement table value Function  
 Stored Procedure  
o Tham số input và output  
I) Batch  
1) Viết một batch khai báo biến @tongsoHD chứa tổng số hóa đơn của sản phẩm  
có ProductID=’778’, nếu @tongsoHD>500 thì in ra chuỗi “San pham 778 có  
trên 500 đơn hàng”, ngược lại tin ra chuỗi “San pham 778 co it don dat hang”  
2) Viết một đoạn Batch với tham số @makh và @n chứa số hóa đơn của khách  
hàng @makh, tham số @nam chứa năm lập hóa đơn (ví dụ @nam=2008), nếu  
@n>0 thì in ra chuỗi:”Khách hàng có @n hóa đơn trong năm 2008” ngược lại  
nếu @n=0 thì in ra chuỗi “Khách hàng không có hóa đơn nào trong năm 2008”

3) Viết một batch tính số tiền giảm cho những hóa đơn (SalesOrderID) có tổng  
tiền>100000, thông tin gồm [SalesOrderID], Subtotal=sum([LineTotal]),  
Discount (tiền giảm), với Discount được tính như sau:  
• Những hóa đơn có Subtotal<100000 thí không giảm,  
• Subtotal từ 100000 đến <120000 thì giảm 5% của Subtotal  
• Subtotal từ 120000 đến <150000 thì giảm 10% của Subtotal  
• Subtotal từ 150000 trở lên thì giảm 15% của Subtotal  
(Gợi ý: Dùng cấu trúc case when …then …)  
4) Viết một Batch với 3 tham số: @mancc, @masp, @soluongcc, chứa giá trị của  
các field [ProductID],[BusinessEntityID],[OnOrderQty], với giá trị truyền cho  
các biến @mancc, @masp (vd: @mancc=1650, @masp)=4, thì chương trình sẽ  
gán giá trị tương ứng của field [OnOrderQty] cho biến @soluongcc, nếu  
@soluongcc trả về giá trị là null thì in ra chuỗi “Nha cung cap 1650 khong cung  
cap san pham 4”, ngược lại (vd: @soluongcc=5) thì in chuỗi “Nha cung cap 1650  
cung cap san pham 4 với số lượng là 5”  
(Gợi ý: Dữ liệu lấy từ [Purchasing].[ProductVendor])  
5) Viết một batch thực hiện tăng lương giờ (Rate) của nhân viên trong  
[HumanResources].[EmployeePayHistory] theo điều kiện sau: Khi tổng lương  
giờ của tất cả nhân viên Sum(Rate)<6000 thì cập nhật tăng lương giờ lên 10%,  
nếu sau khi cập nhật mà lương giờ cao nhất của nhân viên >150 thì dừng.  
WHILE (SELECT SUM(rate) FROM  
[HumanResources].[EmployeePayHistory])<6000  
BEGIN  
UPDATE [HumanResources].[EmployeePayHistory]  
SET rate = rate\*1.1  
IF (SELECT MAX(rate)FROM  
[HumanResources].[EmployeePayHistory]) > 150  
BREAK

ELSE  
CONTINUE  
END  
  
II) Function  
 Scalar Function  
  
  
1) Viết hàm tên countofEmplyees (dạng scalar function) với tham số @mapb, giá  
trị truyền vào lấy từ field [DepartmentID], hàm trả về số nhân viên trong phòng  
ban tương ứng. Áp dụng hàm đã viết vào câu truy vấn liệt kê danh sách các phòng  
ban với số nhân viên của mỗi phòng ban, thông tin gồm: [DepartmentID], Name,  
countOfEmp với countOfEmp= countofEmplyees([DepartmentID]).  
(Dữ liệu lấy từ bảng [HumanResources].[EmployeeDepartmentHistory]  
và [HumanResources].[Department])  
  
2) Viết hàm tên là InventoryProd (dạng scalar function) với tham số vào là  
@ProductID và @locationID trả về số lượng tồn kho của sản phẩm trong khu  
vực tương ứng với giá trị của tham số  
(Dữ liệu lấy từ bảng[Production].[ProductInventory])  
3) Viết hàm tên SubTotalOfEmp (dạng scalar function) trả về tổng doanh thu của  
một nhân viên trong một tháng tùy ý trong một năm tùy ý, với tham số vào  
@EmplID, @MonthOrder, @YearOrder  
(Thông tin lấy từ bảng [Sales].[SalesOrderHeader])  
 Table Valued Functions  
4) Viết hàm sumofOrder với hai tham số @thang và @nam trả về danh sách các  
hóa đơn (SalesOrderID) lặp trong tháng và năm được truyền vào từ 2 tham số  
@thang và @nam, có tổng tiền >70000, thông tin gồm SalesOrderID, Orderdate,  
SubTotal, trong đó SubTotal =sum(OrderQty\*UnitPrice).

5) Viết hàm tên NewBonus tính lại tiền thưởng (Bonus) cho nhân viên bán hàng  
(SalesPerson), dựa trên tổng doanh thu của mỗi nhân viên, mức thưởng mới bằng  
mức thưởng hiện tại tăng thêm 1% tổng doanh thu, thông tin bao gồm  
[SalesPersonID], NewBonus (thưởng mới), sumofSubTotal. Trong đó:  
 SumofSubTotal =sum(SubTotal),  
 NewBonus = Bonus+ sum(SubTotal)\*0.01  
6) Viết hàm tên SumofProduct với tham số đầu vào là @MaNCC (VendorID),  
  
hàm dùng để tính tổng số lượng (sumOfQty) và tổng trị giá (SumofSubtotal)  
của các sản phẩm do nhà cung cấp @MaNCC cung cấp, thông tin gồm  
ProductID, SumofProduct, SumofSubtotal  
  
  
(sử dụng các bảng [Purchasing].[Vendor] [Purchasing].[PurchaseOrderHeader] và  
[Purchasing].[PurchaseOrderDetail])  
  
7) Viết hàm tên Discount\_func tính số tiền giảm trên các hóa đơn (SalesOrderID),  
thông tin gồm SalesOrderID, [SubTotal], Discount, trong đó, Discount được tính  
như sau:  
[SubTotal]<1000 thì Discount=0  
1000>=[SubTotal]<5000 thì Discount = 5%[SubTotal]  
5000>=[SubTotal]<10000 thì Discount = 10%[SubTotal]  
[SubTotal>=10000 thì Discount = 15%[SubTotal]  
Gợi ý: Sử dụng Case when …then …  
case  
when  
when  
when  
else  
end

SubTotal<1000 then 0  
SubTotal>=1000 and SubTotal<5000 then [SubTotal]\*0.05  
SubTotal>=5000 and SubTotal<10000 then [SubTotal]\*0.1  
SubTotal\*0.15  
  
(Sử dụng dữ liệu từ bảng [Sales].[SalesOrderHeader])  
8) Viết hàm TotalOfEmp với tham số @MonthOrder, @YearOrder để tính tổng  
doanh thu của các nhân viên bán hàng (SalePerson) trong tháng và năm được  
truyền và 2 tham số, thông tin gồm [SalesPersonID], Total, với  
Total=Sum([SubTotal])  
 Multi statement Table Valued Functions  
9) Viết lại các câu 5,6,7,8 bằng multi-statement table valued function  
10) Viết hàm tên SalaryOfEmp trả về kết quả là bảng lương của nhân viên, với tham  
số vào là @MaNV (giá trị của [BusinessEntityID]), thông tin gồm  
BusinessEntityID, Fname, Lname, Salary (giá trị của cột Rate).  
 Nếu giá trị của tham số truyền vào là Mã nhân viên khác Null thì kết quả  
  
là bảng lương của nhân viên đó.  
Ví dụ thực thi hàm: select\*from SalaryOfEmp(288)  
  
Kết quả là  
  
  
 Nếu giá trị truyền vào là Null thì kết quả là bảng lương của tất cả nhân  
viên  
Ví dụ: thực thi hàm select\*from SalaryOfEmp(Null)  
Kết quả là 316 record  
  
(Dữ liệu lấy từ 2 bảng [HumanResources].[EmployeePayHistory] và

[Person].[Person] )  
III)  
  
Stored Procedure  
  
1) Viết một thủ tục tính tổng tiền thu (TotalDue) của mỗi khách hàng trong một  
tháng bất kỳ của một năm bất kỳ (tham số tháng và năm) được nhập từ bàn phím,  
thông tin gồm: CustomerID, SumofTotalDue =Sum(TotalDue)  
2) Tạo thủ tục hiển thị tên và số tiền mua của cửa hàng mua nhiều hàng nhất theo  
năm đã cho.  
3) Viết thủ tục Sp\_InsertProduct có tham số dạng input dùng để chèn một mẫu tin  
vào bảng Production.Product. Yêu cầu: chỉ them vào các trường có giá trị not  
null và các field là khóa ngoại.  
4) Viết thủ tục XoaHD, dùng để xóa 1 hóa đơn trong bảng Sales.SalesOrderHeader  
khi  
  
biết  
  
SalesOrderID.  
  
Lưu  
  
ý  
  
trước  
  
khi  
  
xóa

mẫu  
  
tin  
  
trong  
  
Sales.SalesOrderHeader thì phải xóa các mẫu tin của hoá đơn đó trong  
Sales.SalesOrderDetail. Nếu không xoá được hoá đơn thì cũng không được phép  
xóa Sales.SalesOrderDetail của hóa đơn đó.  
5) Viết thủ tục Sp\_Update\_Product có tham số Productid dùng để tăng listprice lên  
10% nếu sản phẩm này tồn tại, ngược lại hiện thông báo không có sản phẩm này.  
  
  
Module 6. TRIGGER  
  
Mục tiêu: Sinh viên hiểu được khái niệm vai trò của trigger trong CSDL, phân biệt  
được các loại trigger, hiện thực được các loại trigger trên CSDL AdventureWorks  
 Trigger chỉ có thể được kích hoạt một cách tự động bởi một trong các lệnh  
Insert, Update, Delete.  
 Khi một trigger được kích hoạt:  
o Dữ liệu được insert hoặc update sẽ được chứa trong bảng Inserted  
o Dữ liệu bị delete được chứa trong bảng Deleted.  
Inserted và Deleted là 2 bảng tạm chỉ có giá trị bên trong trigger, thường  
dùng để kiểm tra dữ liệu trước khi commit hay roll back  
 Có thể áp dụng trigger cho View  
 Loại trigger:  
o INSTEAD OF trigger: bỏ qua các hành động kích hoạt trigger (các  
thao tác insert, delete, update), thay vào đó nó sẽ thực hiện các câu  
lệnh bên trong trigger. Thường dùng trong View với các chức năng  
sau:

▪ Cập nhật nhiều bảng trong view cùng một lúc  
▪ Tăng điều kiện ràng buộc trên các thuộc tính so với CHECK  
▪ Đánh giá trạng thái của bảng trước và sau khi cập nhật dữ  
liệu.  
▪ Cho phép từ chối một phần các câu lệnh và thực thi phần còn  
lại  
o AFTER trigger (FOR): được thực thi sau khi các câu lệnh SQL thực  
thi thành công. AFTER trigger là trigger mặc định khi dùng từ FOR.  
▪ Không dùng trong view  
  
1. Tạo một Instead of trigger thực hiện trên view. Thực hiện theo các bước sau:  
 Tạo mới 2 bảng M\_employees và M\_Department theo cấu trúc sau:  
create table M\_Department  
(  
DepartmentID int not null primary key,  
Name nvarchar(50),  
GroupName nvarchar(50)  
)  
  
  
create table M\_Employees  
(  
EmployeeID int not null primary key,  
Firtname nvarchar(50),  
MiddleName nvarchar(50),  
LastName nvarchar(50),  
DepartmentID int foreign key references M\_Department(DepartmentID)  
)  
 Tạo một view tên EmpDepart\_view bao gồm các field: EmployeeID, Firtname,  
MiddleName, LastName, e.DepartmentID, Name, groupName, dựa trên 2 bảng

M\_employees và M\_Department.  
 Tạo một trigger tên InsteadOf\_Trigger thực hiện trên view EmpDepart\_view,  
dùng để chèn dữ liệu vào các bảng M\_employees và M\_Department khi chèn  
một record mới thông qua view EmpDepart\_view.  
Dữ liệu test:  
insert EmpDepart\_view values(1, 'Nguyen','Hoang','Huy', 11,'Marketing','Sales')  
Kết quả:  
  
2. Tạo một trigger thực hiện trên bảng MySalesOrders có chức năng thiết lập độ ưu  
tiên của khách hàng (custpriority) khi người dùng thực hiện các thao tác Insert,  
Update và Delete trên bảng MySalesOrders theo điều kiện như sau:  
 Nếu tổng tiền Sum(SubTotal) của khách hàng dưới 10,000 $ thì độ ưu tiên của  
khách hàng (custpriority) là 3  
 Nếu tổng tiền Sum(SubTotal) của khách hàng từ 10,000 $ đến dưới 50000 thì  
độ ưu tiên của khách hàng (custpriority) là 2  
 Nếu tổng tiền Sum(SubTotal) của khách hàng từ 50000$ trở lên thì độ ưu tiên  
của khách hàng (custpriority) là 1  
Các bước thực hiện:  
 Tạo bảng Mcustomers và MsalesOrders theo cấu trúc sau:  
create table Mcustomer  
(  
customerID int not null primary key,  
custpriority int  
  
  
)  
create table MsalesOrders  
(  
SalesOrderID int not null primary key,  
OrderDate date,

SubTotal money,  
customerID int foreign key references Mcustomer(customerID) )  
 Chèn dữ liệu cho bảng Mcustomers, lấy dữ liệu từ bảng Sales.Customers,  
nhưng chỉ lấy CustomerID>30100 và CustomerID<30118, cột custpriority cho  
giá trị null.  
 Chèn dữ liệu cho bảng MsalesOrders, lấy dữ liệu từ bảng SalesOrderHeader,  
chỉ lấy những hóa đơn của khách hàng có trong bảng khách hàng.  
 Viết trigger, có thể áp dụng CTE query để lấy dữ liệu từ 2 bảng inserted và  
deleted.  
 Viết câu lệnh kiểm tra việc thưc thi của trigger vừa tạo bằng cách chèn thêm hoặc  
xóa hoặc update một record trên bảng MsalesOrders  
3. Viết một trigger thực hiện trên bảng Memployees sao cho khi người dùng thực hiện  
chèn thêm một nhân viên mới vào bảng Memployees thì chương trình cập nhật số  
nhân viên trong cột NumOfEmployee của bảng MDepartment. Nếu tổng số nhân  
viên của phòng tương ứng <=200 thì cho phép chèn thêm, ngược lại thì hiển thị  
thông báo “Bộ phận đã đủ nhân viên” và hủy giao tác. Các bước thực hiện:  
 Tạo mới 2 bảng Memployees và MDepartment theo cấu trúc sau:  
create table MDepartment  
(  
DepartmentID int not null primary key,  
Name nvarchar(50),  
NumOfEmployee int  
)  
create table Memployees  
(  
EmployeeID int not null,  
Firtname nvarchar(50),  
MiddleName nvarchar(50),  
LastName nvarchar(50),  
DepartmentID int foreign key references MDepartment(DepartmentID)

constraint pk\_emp\_depart primary key(EmployeeID, DepartmentID)  
)  
  
  
 Chèn dữ liệu cho bảng MDepartment, lấy dữ liệu từ bảng Department, cột  
NumOfEmployee gán giá trị NULL, bảng Memployees lấy từ bảng  
EmployeeDepartmentHistory  
 Viết trigger theo yêu cầu trên và viết câu lệnh hiện thực trigger  
4. Bảng [Purchasing].[Vendor], chứa thông tin của nhà cung cấp, thuộc tính  
CreditRating hiển thị thông tin đánh giá mức tín dụng, có các giá trị:  
1 = Superior  
2 = Excellent  
3 = Above average  
4 = Average  
5 = Below average  
Viết một trigger nhằm đảm bảo khi chèn thêm một record mới vào bảng  
[Purchasing].[PurchaseOrderHeader], nếu Vender có CreditRating=5 thì hiển thị  
thông báo không cho phép chèn và đồng thời hủy giao tác.  
Dữ liệu test  
INSERT INTO Purchasing.PurchaseOrderHeader (RevisionNumber, Status,  
EmployeeID, VendorID, ShipMethodID, OrderDate, ShipDate, SubTotal, TaxAmt,  
Freight) VALUES ( 2 ,3, 261, 1652, 4 ,GETDATE() ,GETDATE() , 44594.55,  
,3567.564, ,1114.8638 );  
5. Viết một trigger thực hiện trên bảng ProductInventory (lưu thông tin số lượng sản  
phẩm trong kho). Khi chèn thêm một đơn đặt hàng vào bảng SalesOrderDetail với  
số lượng xác định trong field  
OrderQty, nếu số lượng trong kho  
Quantity> OrderQty thì cập nhật  
lại

số  
  
lượng  
  
trong  
  
kho  
  
Quantity= Quantity- OrderQty,  
ngược lại nếu Quantity=0 thì xuất  
thông báo “Kho hết hàng” và đồng  
thời hủy giao tác.  
Sơ đồ của các bảng liên quan.  
  
  
6. Tạo trigger cập nhật tiền thưởng (Bonus) cho nhân viên bán hàng SalesPerson, khi  
người dùng chèn thêm một record mới trên bảng SalesOrderHeader, theo quy định  
như sau: Nếu tổng tiền bán được của nhân viên có hóa đơn mới nhập vào bảng  
SalesOrderHeader có giá trị >10000000 thì tăng tiền thưởng lên 10% của mức  
thưởng hiện tại. Cách thực hiện:  
• Tạo hai bảng mới M\_SalesPerson và M\_SalesOrderHeader  
create table M\_SalesPerson  
(  
SalePSID int not null primary key,  
TerritoryID int,  
BonusPS money  
)  
create table M\_SalesOrderHeader  
(  
SalesOrdID int not null primary key,

OrderDate date,  
SubTotalOrd money,  
SalePSID int foreign key references M\_SalesPerson(SalePSID)  
)  
• Chèn dữ liệu cho hai bảng trên lấy từ SalesPerson và SalesOrderHeader chọn  
những field tương ứng với 2 bảng mới tạo.  
• Viết trigger cho thao tác insert trên bảng M\_SalesOrderHeader, khi trigger  
thực thi thì dữ liệu trong bảng M\_SalesPerson được cập nhật.