

BÀI TẬP TUẦN 5-6

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:
 - Scalar function
 - Table valued Function
 - Multi Statement table valued Function
 - Stored Procedure
 - 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='31', nếu @tongsoHD>50 thì in ra chuỗi "Sản phẩm 31 có trên 50 đơn hàng", ngược lại in ra chuỗi "Sản phẩm 31 có ít đơn đặt hàng"
- 2) Viết một đoạn Batch với tham số @makh và @n, chứa số hóa đơn @n của khách hàng @makh, tham số @nam chứa năm lập hóa đơn (ví dụ @nam=1996), nếu @n>0 thì in ra chuỗi:"Khách hàng có @n hóa đơn trong năm 1996", ngược lại nếu @n=0 thì in ra chuỗi "Khách hàng không có hóa đơn nào trong năm 1996"
- 3) Viết một batch tính số tiền giảm cho những hóa đơn (OrderID) có tổng tiền>500, thông tin gồm OrderID, TongTien=sum(UnitPrice*Quantity), Tiền giảm, với Tiền giảm được tính như sau:
 - Những hóa đơn có TongTien <500 thì không giảm,
 - TongTien từ 500 đến <5000 thì giảm 5% của TongTien
 - TongTien từ 5000 đến <10000 thì giảm 10% của TongTien
 - TongTien từ 10000 trở lên thì giảm 15% của TongTien*(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 SupplierID, ProductID,Quantity, với giá trị truyền cho các biến @MaNCC, @MaSP (vd: @MaNCC =5, @MaSP =11, thì chương trình sẽ gán giá trị tương ứng của field Quantity cho biến @SoLuongCC, nếu @SoLuongCC trả về giá trị là null thì in ra chuỗi "Nha cung cap 5 khong cung cap san pham 11", ngược lại (vd: @SoLuongCC =12) thì in chuỗi "Nha cung cap 5 cung cap san pham 11 với số lượng là 12"
- 5) Viết một batch thực hiện tăng đơn giá (UnitPrice) trung bình của đơn hàng

(Orders) theo điều kiện sau: Khi trung bình của đơn giá trong đơn hàng < 50 thì cập nhật tăng đơn giá của đơn hàng lên 10%, nếu sau khi cập nhật mà đơn giá lớn nhất của đơn hàng >300 thì dừng.

```

/*
LẬP TRÌNH
- Biến: ý nghĩa, khai báo, cách dùng
- Cú pháp điều khiển: IF, CASE..
- Vòng lặp: Do, While, for
- Hàm (function); thủ tục (Procedure)

*/
-- Batch - Ví dụ 1: Viết lệnh khai biến chứa mã nhân viên. Viết câu lệnh hiển thị
các hóa đơn được lập
-- bởi nhân viên này
Declare @manv int
Set @manv = 2
Select * from [dbo].[Orders]
where [EmployeeID] = @manv
--Ví dụ 2: đếm số hóa đơn được lập cho 1 khách hàng nào đó. Nếu số hóa đơn được lập
>5
-- Xuất thông báo 'Số hóa đơn đã lập cho KH xx >5'
-- Ngược lại: 'Số hóa đơn đã lập cho KH xx <=5'
go
Declare @maKH nchar (5), @sohd int
Set @maKH = 'VINET'
Select @sohd= COUNT([OrderID]) from [dbo].[Orders]
where [CustomerID] = @maKH
If (@sohd >5)
    print 'Số hóa đơn đã lập cho KH:' + @maKH + ' >5'
else
    print 'Số hóa đơn đã lập cho KH:' + @maKH + ' <=5'

```

II)

III) **Function:** (Nhớ kiểm tra từng hàm sau khi làm xong, bằng cách gọi hàm và truyền tham số)

– **Scalar Function:**

1) Viết hàm tên **CountOfProducts** (dạng scalar function) với tham số @MaNhom, giá trị truyền vào lấy từ field CategoryID, hàm trả về số sản phẩm tương ứng với mã nhóm hàng. Áp dụng hàm đã viết vào câu truy vấn liệt kê danh sách các nhóm hàng cùng với số sản phẩm thuộc mỗi nhóm, thông tin gồm: CategoryID, CategoryName, CountOfProduct.

```

CREATE FUNCTION CountofProducts (@manhom int)
Returns int
as
begin
    declare @sosp int
    Select @sosp = count(ProductID) From [dbo].[Products]
    where [CategoryID] = @manhom
    return @sosp
end
--su dung ham
Select [CategoryID], [CategoryName], dbo.CountofProducts ([CategoryID]) as 'Số sản phẩm'

```

from [dbo].[Categories]

- 2) Viết hàm tên là **InstockProd** (dạng scalar function) với tham số vào là @ProductID và @SupplierID. Hàm trả về số lượng tồn kho (UnitInStock) của sản phẩm @ProductID do nhà cung cấp @SupplierID cung cấp.
- 3) Viết hàm tên **SalesOfEmp** (dạng scalar function) trả về tổng doanh thu bán hàng (SUM(UnitPrice*Quantity)) của một nhân viên trong một tháng tùy ý và trong một năm tùy ý, với tham số vào @EmployeeID, @MonthOrder, @YearOrder

– **Table Valued Functions**

- 4) Viết hàm **SumofOrder** với hai tham số @thang và @nam, trả về danh sách các hóa đơn (OrderID) lập trong tháng và năm được truyền vào từ 2 tham số @thang và @nam, có tổng tiền >5000, thông tin gồm OrderID, OrderDate, **SubTotal**, trong đó **SubTotal = sum(Quantity*UnitPrice)**.
- 5) Viết hàm tên **SumOfProduct** với tham số đầu vào là @MaNCC (**SupplierID**), hàm dùng để tính tổng số lượng (**SumOfQuantity**) và chiết khấu cao nhất (**MaxOfDiscount**) của các sản phẩm do nhà cung cấp @MaNCC cung cấp, thông tin gồm **ProductID**, **SumOfQuantity**, **MaxOfDiscount**
- 6) Viết hàm tên **Discount_Func** tính số tiền giảm trên các hóa đơn (OrderID), thông tin gồm OrderID, Quantity, Discount, trong đó, Discount được tính như sau:
Nếu Quantity < 10 thì Discount = 0,
Nếu 10 <= Quantity < 30 thì Discount = 5% [UnitPrice*Quantity]
Nếu 30 <= Quantity < 50 thì Discount = 10% [UnitPrice*Quantity]
Nếu Quantity >= 50 thì Discount = 15% [UnitPrice*Quantity]

Gợi ý: Sử dụng Case When ...Then ...

```
Select Discount=
Case
  when Quantity < 10 then 0
  when Quantity >= 10 and Quantity < 30 then 5% * (UnitPrice*Quantity)
  when Quantity >= 30 and Quantity < 50 then 10% * (UnitPrice*Quantity)
  else 15% * (UnitPrice*Quantity)
End
From [Order Details]
```

- 7) Viết hàm **TotalOfEmp** với tham số @MonthOrder, @YearOrder để tính tổng doanh thu của các nhân viên trong tháng và năm được truyền vào 2 tham số, thông tin gồm EmployeeID, **Total**, với **Total = Sum(UnitPrice*Quantity)**.

– **Multi statement Table Valued Functions**

- 8) Viết lại các câu 5,6,7 bằng multi-statement table valued function
- 9) Viết hàm tên TonKho trả về kết quả là bảng tồn kho của sản phẩm, với tham số vào là @MaSP, thông tin gồm ProductID, ProductName, UnitsInStock. .
 - Nếu giá trị của tham số truyền vào là @MaSP khác Null thì kết quả là bảng tồn kho của sản phẩm đó.
 - Nếu giá trị truyền vào là Null thì kết quả là bảng tồn kho của tất cả sản phẩm

IV) Stored Procedure: (Nhớ kiểm tra từng thủ tục sau khi làm xong, bằng cách gọi thủ tục và truyền tham số)

- 1) Viết một thủ tục tính tổng số hàng tồn kho (UnitsInStock) của từng nhà cung cấp trong một quốc gia nào đó, thông tin gồm: SupplierID, SumOf UnitsInStock
- 2) Viết thủ tục có tham số a, b dạng input để giải phương trình bậc nhất $ax+b=0$.
- 3) Viết một thủ tục dùng để thực hiện các công việc sau: tạo một bảng tạm tên là #MyTable, chèn hai mẫu tin tùy ý vào bảng này, dùng câu lệnh SELECT ... FROM.. để liệt kê các mẫu tin có trong bảng #MyTable. Cấu trúc của #MyTable như sau CustID nchar(5), CustName nvarchar(40), AvgDiff float.
- 4) Tạo thủ tục đặt tên là TongThu có tham số vào là mã nhân viên, tham số đầu ra là tổng trị giá các hóa đơn nhân viên đó bán được. Sử dụng lệnh RETURN để trả về trạng thái thành công hay thất bại của thủ tục.
- 5) Tạo thủ tục hiển thị tên và số tiền mua của khách hàng mua nhiều hàng nhất theo năm đã cho.
- 6) Viết thủ tục Sp_InsertProduct có tham số dạng input dùng để chèn một mẫu tin vào bảng Products. Yêu cầu: chỉ thêm vào các trường có giá trị not null và các field là khóa ngoại.
- 7) Viết thủ tục XoaHD, dùng để xóa 1 hóa đơn trong bảng Orders khi biết OrderID. Lưu ý trước khi xóa mẫu tin trong bảng Orders thì phải xóa các mẫu tin của hóa đơn đó trong bảng [Order Details]. Nếu không xóa được các dòng trong bảng Orders thì cũng không được phép xóa dòng trong bảng [Order Details] ứng với hóa đơn đó.
- 8) Viết thủ tục Sp_Update_Product có tham số vào là ProductID, dùng để tăng UnitsInStock lên 10% nếu sản phẩm này tồn tại, ngược lại hiện thông báo không có sản phẩm này.