

```
--Ho Phuc Lam - LOP TMDT
--Bai tap tuan 05-06 ham va thu tuc
```

```
/*
```

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

```
*/
```

```
use Northwind
Go
```

```
--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
```

```
create proc TongHangTonKho @country nvarchar(15)
as
```

```
    select p.SupplierID, sum(p.UnitsInStock) as SumOfUnitInStock
    from Products p
    join Suppliers s on s.SupplierID = p.SupplierID
    where s.Country = @country
    group by p.SupplierID
```

```
Go
```

```
--Thuc thi
```

```
exec TongHangTonKho 'USA'
Go
```

```
--2) Viết thủ tục có tham số a, b dạng input để giải phương trình bậc nhất  $ax+b=0$ .
```

```
create proc ptBacNhat
```

```
    @a int, @b int
```

```
as
```

```
    begin
```

```

declare @x float;

--kiem tra he so a
if @a = 0
begin
    if @b = 0
        print N'Phương trình có vô số nghiệm.';
    else
        print N'Phương trình vô nghiệm.';
    end
else
begin
    set @x = -@b / @a;
    print N'Phương trình có nghiệm: x = ' + convert(nvarchar(10), @x);
end
end

Go

--thuc thi
exec ptBacNhat 2,5
Go

```

```

/*
--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
*/
create proc ListOfTableTemp
as
begin

```

```

create table #Mytable(
    CustID nchar(5),
    CustName nvarchar(40),
    AvgDiff float
);

insert into #Mytable (CustID, CustName, AvgDiff)
values ('C01', 'NumberOne', '12.12'),
       ('C02', 'NumberTwo', '70.70');

select * from #Mytable;

drop table #Mytable;

```

end

Go

--thuc thi

exec ListOfTableTemp

Go

/\*  
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.  
\*/

```

create proc TongThu
    @MaNV int,
    @Total float output

```

as

```

begin
    select @Total = sum(od.UnitPrice*od.Quantity)
    from Orders o

```

```
join [Order Details] od on o.OrderID = od.OrderID
where o.EmployeeID = @MaNV
```

```
if @Total is null
begin
    set @Total = 0;
    return 0;
end
else
    return 1;
```

```
end
```

```
Go
```

```
--thuc thi
```

```
declare @Total float;
```

```
declare @Binary int;
```

```
exec @Binary = TongThu @MaNV = 11, @Total = @Total output;
```

```
print N'Tổng giá trị các hóa đơn: ' + convert(nvarchar(20), @Total);
```

```
if @Binary = 0
```

```
    print N'Trạng thái trả về: ' + convert(nvarchar(20), @Binary) + N'. Thất bại!';
```

```
if @Binary = 1
```

```
    print N'Trạng thái trả về: ' + convert(nvarchar(20), @Binary) + N'. Thành công!';
```

```
GO
```

```
/*cau 5: Tạo thủ tục hiển thị tên và số tiền mua của khách hàng mua nhiều hàng nhất theo năm đã cho.*/
```

```
go
```

```
CREATE PROC SP_TOP @nam int
```

```
AS
```

```
SELECT C.CompanyName, SUM(OD.UnitPrice*OD.Quantity) AS ToTal
```

```
FROM [dbo].[Orders] O JOIN [dbo].[Customers] C ON C.CustomerID=O.CustomerID
```

```

JOIN [dbo].[Order Details] OD ON OD.OrderID=O.OrderID
WHERE YEAR(O.OrderDate)=@nam
GROUP BY C.CompanyName
having SUM(OD.UnitPrice*OD.Quantity)=
(
    SELECT TOP 1 SUM(OD.UnitPrice*OD.Quantity) AS ToTal
    FROM [dbo].[Orders] O JOIN [dbo].[Customers] C ON C.CustomerID=O.CustomerID
    JOIN [dbo].[Order Details] OD ON OD.OrderID=O.OrderID
    WHERE YEAR(O.OrderDate)=@nam
    GROUP BY C.CompanyName
    ORDER BY SUM(OD.UnitPrice*OD.Quantity) DESC
)
ORDER BY SUM(OD.UnitPrice*OD.Quantity) DESC
---
EXEC SP_TOP 1997
Go

```

/\*  
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.  
\*/

```

create proc Sp_InsertProduct
    @proName nvarchar(40),
    @supID int,
    @cateID int,
    @discontinue bit
as
begin
    insert into Products( ProductName, SupplierID, CategoryID, Discontinued)
    values (@proName, @supID, @cateID, @discontinue);
    return 1;

```

```
    end  
    return 0;  
Go
```

```
--thuc thi  
declare @result int;  
exec @result = Sp_InsertProduct @proName = 'NewIPhone',@supID = '1', @cateID = '2',@discontinue =  
'0'  
if @result = 1  
    print N'Thêm thông tin sản phẩm thành công!'  
else  
    print N'Lỗi!!! Thêm thất bại!'  
Go
```

```
/*  
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 hoá 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.  
*/
```

```
create proc XoaHD  
    @odID int  
as  
    begin  
        delete from [Order Details] where OrderID = @odID;  
        delete from Orders where OrderID = @odID;  
        print N'Xóa thành công'  
        return 1;  
    end  
    print N'Xóa thất bại'
```

```
        return 0;
Go
```

```
--thuc thi
declare @result int
exec @result = XoaHD @odID = 0
if @result = 0 print N'XoaHD thất bại'
if @result = 1 print N'XoaHD thành công'
Go
```

```
/*
```

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.

```
*/
```

```
create proc Sp_Update_Product
    @ProID int
as
begin
    if exists (select 1 from Products where ProductID = @ProID)
        begin
            update Products
            set UnitPrice = UnitPrice * 1.10
            where ProductID = @ProID
            print N'Đã tăng giá sản phẩm id = '+convert(nvarchar,@ProID)+' lên 10%'
        end
    else
        begin
            print N'Không có sản phẩm id = '+convert(nvarchar, @ProID)
        end
    end
end
```

Go

--thực thi

exec Sp\_Update\_Product @ProID = 1;

Go

/\*

Bài tập 16/07/2024

Tạo thủ tục ListProduct chứa danh sách những sản phẩm nhóm Beverages và Seafood có tổng số lượng bán trong mỗi năm trên 30 sản phẩm, thông tin gồm CategoryName, ProductName, Year, SumOfOrderQuantity

Sửa lại: tạo thủ tục có tham số đầu vào là tên nhóm nào đó.

Sửa lại: Tạo thủ tục có tham số đầu vào là tên nhóm nào đó; sử dụng tham số Output để trả về Tổng lượng bán trong 1 năm

\*/

create proc p\_ViewListProduct\_1

as

```
    select CategoryName, ProductName, Year(o.OrderDate) as Years, sum(od.Quantity) as
SumOfOrderQuantity
    from [Order Details] od
    join Orders o on od.OrderID = o.OrderID
    join Products p on p.ProductID = od.ProductID
    join Categories c on p.CategoryID = c.CategoryID
    where c.CategoryName like '%Beverages%' or c.CategoryName like '%Seafood%'
    group by CategoryName, ProductName, Year(o.OrderDate)
    having sum(od.Quantity) > '30'
```

Go

--thực thi

exec p\_ViewListProduct\_1

Go



--Sửa lại: tạo thủ tục có tham số đầu vào là tên nhóm nào đó.

```
create proc p_ViewListProduct_2
    @substring nvarchar(30)
as
    select CategoryName, ProductName, Year(o.OrderDate) as Years, sum(od.Quantity) as
SumOfOrderQuantity
    from [Order Details] od
    join Orders o on od.OrderID = o.OrderID
    join Products p on p.ProductID = od.ProductID
    join Categories c on p.CategoryID = c.CategoryID
    where c.CategoryName like '%' + @substring + '%'
    group by CategoryName, ProductName, Year(o.OrderDate)
    having sum(od.Quantity) > '30'
Go
--Thuc thi
exec p_ViewListProduct_2 @substring = 'Seafood';
Go
```

--Sửa lại: Tạo thủ tục có tham số đầu vào là tên nhóm nào đó; sử dụng tham số Output để trả về Tổng lượng bán trong 1 năm

```
create proc p_ViewListProduct_3
    @substring nvarchar(30),
    @total int output
as
    --truy vấn
    declare @sumTotal int
    select CategoryName, ProductName, Year(o.OrderDate) as Years, sum(od.Quantity) as
SumOfOrderQuantity
    from [Order Details] od
    join Orders o on od.OrderID = o.OrderID
    join Products p on p.ProductID = od.ProductID
    join Categories c on p.CategoryID = c.CategoryID
```

```

where c.CategoryName like '%' + @substring + '%'
group by CategoryName, ProductName, Year(o.OrderDate)
having sum(od.Quantity) > '30'

--tính lượng bán ra
select @sumTotal = sum(od.Quantity)
from [Order Details] od
join Orders o on od.OrderID = o.OrderID
join Products p on p.ProductID = od.ProductID
join Categories c on p.CategoryID = c.CategoryID
where c.CategoryName like '%' + @substring + '%'

set @total = @sumTotal
Go

-- Thực thi
declare @sumtotal INT;
exec p_ViewListProduct_3 @substring = 'Seafood', @total = @sumtotal OUTPUT;
-- Xem giá trị tổng lượng bán
select @sumtotal as TotalSales;
Go

```