

BÀI TẬP TUẦN 05

--Module 4. Batch, Stored Procedure, Function

--Tuần 5

--II) 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)

```
create proc Tongthu @thang int, @nam int
as
begin
    select c.CustomerID, Sum(TotalDue) as SumOfTotalDue
    from [Sales].[SalesOrderHeader] h JOIN Sales.Customer c
        on h.CustomerID = c.CustomerID
    where @thang = month([OrderDate]) and @nam = year([OrderDate])
    group by c.CustomerID
end
go
```

--tìm hiểu dữ liệu

```
select c.CustomerID, orderdate, Sum(TotalDue) as SumOfTotalDue
from [Sales].[SalesOrderHeader] h JOIN Sales.Customer c
    on h.CustomerID = c.CustomerID
group by c.CustomerID, OrderDate
--tháng 2, năm 2006
--tháng 7, năm 2009
```

--thực thi

exec Tongthu 2,2006

exec Tongthu 7,2009

go

--2) Viết một thủ tục dùng để xem doanh thu từ đầu năm cho đến ngày hiện tại
--(SalesYTD) của một nhân viên bất kỳ, với một tham số đầu vào và một tham số
--đầu ra. Tham số @SalesPerson nhận giá trị đầu vào theo chỉ định khi gọi thủ tục,
--tham số @SalesYTD được sử dụng để chứa giá trị trả về của thủ tục.

```
create proc DoanhThuTuDauNamCuaNhanVien @SalesPerson int , @SalesYTD money out
as
begin
    select @SalesYTD = [SalesYTD]
    from [Sales].[SalesPerson]
    where @SalesPerson = [BusinessEntityID]
end
go
```

--tìm hiểu dữ liệu

```
select [BusinessEntityID], SalesYTD
from [Sales].[SalesPerson]
--[BusinessEntityID] =277
--[BusinessEntityID] =291
```

--gọi thủ tục bằng batch

declare @SalesYTD money

exec DoanhThuTuDauNamCuaNhanVien 277, @SalesYTD out

select @SalesYTD as [Doanh thu nam] --in trị đầu ra

go

--3) Viết một thủ tục trả về một danh sách ProductID, ListPrice của các sản phẩm có
--giá bán không vượt quá một giá trị chỉ định (tham số input @MaxPrice).

```
create proc [Gia khong vuot chi dinh] @MaxPrice money
as
begin
    select [ProductID], [ListPrice]
    from [Production].[Product]
    where [ListPrice] <= @MaxPrice
end
go
```

```
--tìm hiểu dữ liệu
select [ProductID], [ListPrice]
from [Production].[Product]
-- ListPrice = 0
--ListPrice = 4000
```

```
--thực thi
exec [Gia khong vuot chi dinh] 4000
go
```

--4) Viết thủ tục tên NewBonus cập nhật lại tiền thưởng (Bonus) cho 1 nhân viên bán
--hàng (SalesPerson), dựa trên tổng doanh thu của nhân viên đó. Mức thưởng mới
--bằng mức thưởng hiện tại cộng 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

```
--Kiểm tra dữ liệu hiện tại:
select p.BusinessEntityID, p.Bonus, sum(SubTotal) as SumOfTotal
from [Sales].[SalesOrderHeader] h JOIN [Sales].[SalesPerson] p
    on h.SalesPersonID = p.BusinessEntityID
group by p.BusinessEntityID, p.Bonus
--p.BusinessEntityID=281 có Bonus = 3550.00 và SumOfTotal= 6427005.554
```

```
--tạo thủ tục
create proc NewBonus @SalesPersonID int
as
begin
    update [Sales].[SalesPerson]
    set [Bonus] = Bonus + (select sum(SubTotal)
        from [Sales].[SalesOrderHeader] h
        JOIN [Sales].[SalesPerson] p
        on h.SalesPersonID = p.BusinessEntityID
        where BusinessEntityID = @SalesPersonID
    ) * 0.01
    where BusinessEntityID = @SalesPersonID
end
go
```

```
--Thực thi
exec NewBonus 281
go
```

```
--Kiểm tra dữ liệu sau khi cập nhật
select p.BusinessEntityID, p.Bonus, sum(SubTotal) as SumOfTotal
from [Sales].[SalesOrderHeader] h JOIN [Sales].[SalesPerson] p
    on h.SalesPersonID = p.BusinessEntityID
group by p.BusinessEntityID, p.Bonus
--p.BusinessEntityID=281 sau khi cập nhật có Bonus= 67820.0555
```

```
--5) Viết một thủ tục dùng để xem thông tin của nhóm sản phẩm (ProductCategory)
--có tổng số lượng (OrderQty) đặt hàng cao nhất trong một năm tùy ý (tham số
--input), thông tin gồm: ProductCategoryID, Name, SumOfQty. Dữ liệu từ bảng
--ProductCategory, ProductSubCategory, Product và SalesOrderDetail.
--(Lưu ý: dùng Sub Query)
```

```
create proc [SP duoc dat hang nhieu nhat trong nam] @nam int
as
begin
    select c.ProductCategoryID, c.Name, sum(OrderQty) as SumOfQty
    from Production.ProductCategory c JOIN Production.ProductSubcategory s
    on c.ProductCategoryID = s.ProductCategoryID
    JOIN Production.Product p on s.ProductSubcategoryID = p.ProductSubcategoryID
    JOIN Sales.SalesOrderDetail od on od.ProductID = p.ProductID
    JOIN Sales.SalesOrderHeader h on h.SalesOrderID = od.SalesOrderID
    where year(OrderDate) = @nam
    group by c.ProductCategoryID, c.Name
    having sum(OrderQty) >= all(
        select sum(OrderQty)
        from Production.ProductCategory c JOIN Production.ProductSubcategory s
        on c.ProductCategoryID = s.ProductCategoryID
        JOIN Production.Product p on s.ProductSubcategoryID =
p.ProductSubcategoryID
        JOIN Sales.SalesOrderDetail od on od.ProductID = p.ProductID
        JOIN Sales.SalesOrderHeader h on h.SalesOrderID = od.SalesOrderID
        where year(OrderDate) = @nam
        group by c.ProductCategoryID, c.Name
    )
end
go

--tìm hiểu dữ liệu

select c.ProductCategoryID, c.Name, sum(OrderQty), year(orderdate)
from Production.ProductCategory c JOIN Production.ProductSubcategory
s
on c.ProductCategoryID = s.ProductCategoryID
JOIN Production.Product p on s.ProductSubcategoryID =
p.ProductSubcategoryID
JOIN Sales.SalesOrderDetail od on od.ProductID = p.ProductID
JOIN Sales.SalesOrderHeader h on h.SalesOrderID = od.SalesOrderID
group by c.ProductCategoryID, c.Name, year(orderdate)
-- 2007 có số sum(OrderQty) lớn nhất là 37042

--thực thi
exec [SP duoc dat hang nhieu nhat trong nam] 2007
go
```

```

--6) 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, @TongTG money out
as
    begin
        set @TongTG =
            (
                select sum(TotalDue)
                from [Sales].[SalesOrderHeader] h JOIN [Sales].[SalesPerson] p
                 on h.SalesPersonID = p.BusinessEntityID
                where p.BusinessEntityID = @maNV
                group by p.BusinessEntityID
            )
        if @TongTG >0 return 0
        else return 1
    end
go

--tìm hiểu dữ liệu
select p.BusinessEntityID, sum(TotalDue)
from [Sales].[SalesOrderHeader] h JOIN [Sales].[SalesPerson] p
 on h.SalesPersonID = p.BusinessEntityID
group by p.BusinessEntityID
--BusinessEntityID = 284
--BusinessEntityID = 260

--thực thi bằng bath
declare @TongTG money, @TriTraVe int
exec @TriTraVe=Tongthu 284, @TongTG out
select @TongTG as [Tong Tri Gia],
       @TriTraVe as TriTraVe
if @TriTraVe = 0
    print N'Thủ tục thành công'
else
    print N'Thủ tục không tính được'
go

```

--7) Tạo thủ tục hiển thị tên và số tiền mua của cửa hàng mua nhiều hàng nhất theo
--năm đã cho.

```
create proc [Cua hang mua nhieu hang nhat] @nam int
as
begin
    select TOP 1 s.Name, sum(OrderQty*UnitPrice), year(orderdate) as Years
    from [Sales].[SalesOrderHeader] h JOIN [Sales].[SalesOrderDetail] od
    on h.SalesOrderID = od.SalesOrderID
    JOIN [Sales].[Store] s on s.[SalesPersonID] = h.SalesPersonID
    where year(orderdate) = @nam
    group by s.Name, year(orderdate)
    order by year(orderdate) asc, sum(OrderQty*UnitPrice) desc
end
go

--Kiểm tra dữ liệu
select s.Name, sum(OrderQty*UnitPrice), year(orderdate)
from [Sales].[SalesOrderHeader] h JOIN [Sales].[SalesOrderDetail] od
on h.SalesOrderID = od.SalesOrderID
JOIN [Sales].[Store] s on s.[SalesPersonID] = h.SalesPersonID
group by s.Name, year(orderdate)
order by year(orderdate) asc, sum(OrderQty*UnitPrice) desc
-- Năm 2006 , Cửa hàng Friendly Bike Shop có tổng mua lớn nhất
4279680.3954
--Năm 2007, Cửa hàng Sports Products Store có tổng mua lớn nhất
6452990.3166

--thực thi
exec [Cua hang mua nhieu hang nhat] 2007
```

```

--III) Function
-- Scalar Function

--1) Viết hàm tên CountOfEmployees (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= CountOfEmployees([DepartmentID]).
--(Dữ liệu lấy từ bảng
--[HumanResources].[EmployeeDepartmentHistory] và
--[HumanResources].[Department])

create function CountOfEmployees(@mapb smallint)
returns int
as
begin
    declare @soNV int
    select @soNV = count([BusinessEntityID])
    from [HumanResources].[Department] d JOIN [HumanResources].[EmployeeDepartmentHistory] h
        on d.DepartmentID = h.DepartmentID
    where @mapb = d.DepartmentID
    return @soNV
end

--thuc thi
declare @mapb smallint
set @mapb = 2
select dbo.CountOfEmployees(@mapb) as [So Nhan Vien Moi Phong Ban]
print 'Phong ban ma ' + convert(varchar(5),@mapb) + ' co ' +
convert(varchar(5),dbo.CountOfEmployees(@mapb)) + ' nhan vien'

--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])
create function InventoryProd (@ProductID int, @LocationID smallint)
returns int
as
begin
    declare @soLuongTonKho int
    select @soLuongTonKho = [Quantity]
    from [Production].[ProductInventory]
    where @ProductID = [ProductID] and @LocationID = [LocationID]
    return @soLuongTonKho
end

--thuc thi
declare @ProductID int , @LocationID smallint
set @ProductID = 1
set @LocationID = 50
select dbo.InventoryProd(@ProductID, @LocationID) as [Tong so luong ton kho]
print 'Tong so luong ton kho cua ProductID ' + convert(varchar(5),@ProductID) + ' va LocationID ' +
convert(varchar(5),@LocationID) + ' la ' + convert(varchar(10), dbo.InventoryProd(@ProductID,
@LocationID))

```

```

--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
--@EmpID, @MonthOrder, @YearOrder
--(Thông tin lấy từ bảng [Sales].[SalesOrderHeader])
create function SubTotalOfEmp (@EmpID int, @MonthOrder datetime, @YearOrder datetime)
returns money
as
    begin
        declare @tongDoanhThu money
        select @tongDoanhThu = [TotalDue]
        from [Sales].[SalesOrderHeader]
        where @EmpID = [SalesPersonID] and @MonthOrder = MONTH([DueDate]) and @YearOrder =
YEAR([DueDate])
        return @tongDoanhThu
    end

--thuc thi
declare @EmpID int, @MonthOrder datetime, @YearOrder datetime
set @EmpID = 279
set @MonthOrder = 07
set @YearOrder = 2005
select dbo.SubTotalOfEmp(@EmpID , @MonthOrder , @YearOrder ) as [Tong doanh thu]
print 'Tong doanh thu cua nhan vien ' + convert(varchar(5), @EmpID) + ' trong thang ' +
convert(varchar(2), convert(int, @MonthOrder)) + ' nam ' +
convert(varchar(4), convert(int, @YearOrder)) + ' la ' +
convert(varchar(10), dbo.SubTotalOfEmp(@EmpID , @MonthOrder , @YearOrder ))

```