

BÀI TẬP TUẦN 06

```
--Tuần 6
--In-line Table Valued Functions:
--4) Viết hàm SumOfOrder với hai tham số @thang và @nam trả về danh sách các
--hóa đơn (SalesOrderID) lập 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).
create function SumOfOrder(@thang datetime, @nam datetime)
returns table
as
    return
        (select h.[SalesOrderID], [OrderDate], sum(OrderQty*UnitPrice) as SubTotal
        from [Sales].[SalesOrderHeader] h JOIN [Sales].[SalesOrderDetail] d
            on h.SalesOrderID = d.SalesOrderID
        where @thang = MONTH([OrderDate]) and @nam = YEAR([OrderDate])
        group by h.[SalesOrderID], [OrderDate]
        having sum(OrderQty*UnitPrice) > 70000)

go

--thuc thi
declare @thang datetime, @nam datetime
set @thang = 8
set @nam = 2005
select * from SumOfOrder(@thang , @nam )
go

--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
create function NewBonus2 (@SalesPersonID int )
returns table
as
    return
        select [BusinessEntityID] , (Bonus + ( select sum(SubTotal)
                                                    from
[Sales].[SalesOrderHeader] h JOIN [Sales].[SalesPerson] p
                                                    on
h.SalesPersonID = p.BusinessEntityID
                                                    where
BusinessEntityID = @SalesPersonID
                                                    )
            * 0.01
        ) as NewBonus
        from [Sales].[SalesPerson]

go

--thuc thi
declare @SalesPersonID int
set @SalesPersonID = 278
select * from NewBonus2(@SalesPersonID)
go
```

```

--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])
create function SumOfProduct(@MaNCC int)
returns table
as
    return (
        select [ProductID] ,sum([OrderQty]) as SumOfQty, sum([SubTotal]) as SumOfSubTotal
        from [Purchasing].[Vendor] v JOIN [Purchasing].[PurchaseOrderHeader] h on
v.BusinessEntityID = h.VendorID
            JOIN [Purchasing].[PurchaseOrderDetail] d on d.PurchaseOrderID =
h.PurchaseOrderID
        where @MaNCC = VendorID
        group by ProductID
    )
go

--thuc thi
declare @MaNCC int
set @MaNCC = 1566
select * from dbo.SumOfProduct(@MaNCC)
go

--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:
--Nếu [SubTotal]<1000 thì Discount=0
--Nếu 1000<=[SubTotal]<5000 thì Discount = 5%[SubTotal]
--Nếu 5000<=[SubTotal]<10000 thì Discount = 10%[SubTotal]
--Nếu [SubTotal]>=10000 thì Discount = 15%[SubTotal]
create function Discount_Func ()
returns table
as
    return
    (
        select [SalesOrderID], [SubTotal],
            (
                case
                when SubTotal<1000 then 0
                when SubTotal>=1000 and SubTotal<5000 then [SubTotal]*0.05
                when SubTotal>=5000 and SubTotal<10000 then [SubTotal]*0.1
                else SubTotal*0.15
                end
            ) as Discount
        from [Sales].[SalesOrderHeader]
    )
go

--thuc thi
select * from dbo.Discount_Func()
go

```

```

--8) Viết hàm TotalOfEmp với tham số @MonthOrder, @YearOrder để tính tổng
--doanh thu của các nhân viên bán hàng (SalesPerson) trong tháng và năm được
--truyền vào 2 tham số, thông tin gồm [SalesPersonID], Total, với
--Total=Sum([SubTotal])
create function TotalOfEmp (@MonthOrder datetime, @YearOrder datetime)
returns table
as
    return(
        select [SalesPersonID], sum([SubTotal]) as Total
        from [Sales].[SalesOrderHeader]
        where @MonthOrder = MONTH(OrderDate) and @YearOrder = YEAR(OrderDate)
        group by [SalesPersonID]
    )
go

--thuc thi
declare @nam datetime, @thang datetime
set @nam = 2007
set @thang = 12
select * from TotalOfEmp(@thang, @nam)
go

--9) Viết lại các câu 5,6,7,8 bằng Multi-statement table valued function
-----
create function NewBonus3 (@SalesPersonID int )
returns @NewBonus2 table
([BusinessEntityID] int, NewBonus money)
as
    begin
        insert @NewBonus2
        select [BusinessEntityID] , (Bonus + ( select sum(SubTotal)
                                                from
[Sales].[SalesOrderHeader] h JOIN [Sales].[SalesPerson] p
                                                on
h.SalesPersonID = p.BusinessEntityID
                                                where
BusinessEntityID = @SalesPersonID
                                                )
        * 0.01
        )
        from [Sales].[SalesPerson]
    return
    end
go

--thuc thi
declare @SalesPersonID int
set @SalesPersonID = 279
select * from NewBonus3(@SalesPersonID)
go

```

```

-----
--6) Làm lại: 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
create function SumOfProduct2(@MaNCC int)
returns @fn_SumOfProduct table
(ProductID int not null, SumOfQty smallint, SumOfSubTotal money )
as
begin
    insert @fn_SumOfProduct
    select [ProductID] , sum([OrderQty]) , sum([SubTotal])
    from [Purchasing].[Vendor] v JOIN [Purchasing].[PurchaseOrderHeader] h on
v.BusinessEntityID = h.VendorID
        JOIN [Purchasing].[PurchaseOrderDetail] d on d.PurchaseOrderID =
h.PurchaseOrderID
    where @MaNCC = VendorID
    group by ProductID
    return
end
go

--thi hanh
declare @MaNCC int
set @MaNCC = 1566
select * from dbo.SumOfProduct2(@MaNCC)
go

-----
--7) Làm lại: 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:
--Nếu [SubTotal]<1000 thì Discount=0
--Nếu 1000<=[SubTotal]<5000 thì Discount = 5%[SubTotal]
--Nếu 5000<=[SubTotal]<10000 thì Discount = 10%[SubTotal]
--Nếu [SubTotal]>=10000 thì Discount = 15%[SubTotal]

create function Discount_Func2 ()
returns @fn_Discount_Func2 table
(SalesOrderID int not null, Subtotal money, Discount money )
as
begin
    insert @fn_Discount_Func2
    select [SalesOrderID], [SubTotal],
        (
            case
            when SubTotal<1000 then 0
            when SubTotal>=1000 and SubTotal<5000 then [SubTotal]*0.05
            when SubTotal>=5000 and SubTotal<10000 then [SubTotal]*0.1
            else SubTotal*0.15
            end
        )
    from [Sales].[SalesOrderHeader]
    return
end
go

--thuc thi
select * from dbo.Discount_Func2()
go
-----

```

```

--8) Lam lai: Viết hàm TotalOfEmp với tham số @MonthOrder, @YearOrder để tính tổng
--doanh thu của các nhân viên bán hàng (SalesPerson) trong tháng và năm được
--truyền vào 2 tham số, thông tin gồm [SalesPersonID], Total, với
--Total=Sum([SubTotal])
create function TotalOfEmp2 (@MonthOrder datetime, @YearOrder datetime)
returns @TotalOfEmp2 table
(SalesPersonID int , Total money)
as
begin
    insert @TotalOfEmp2
    select [SalesPersonID], sum([SubTotal])
    from [Sales].[SalesOrderHeader]
    where @MonthOrder = MONTH(OrderDate) and @YearOrder = YEAR(OrderDate)
    group by [SalesPersonID]
    return
end
go

--thuc thi
declare @nam datetime, @thang datetime
set @nam = 2007
set @thang = 12
select * from TotalOfEmp2(@thang, @nam)
go

-----
--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.
create function SalaryOfEmp(@MaNV int)
returns @SalaryOfEmp table
(BusinessEntityID int not null, FName nvarchar(50), LName nvarchar(50), Salary money )
as
begin
    insert @SalaryOfEmp
    select p.BusinessEntityID, [FirstName], [LastName], Rate
    from [Person].[Person] p JOIN [HumanResources].[EmployeePayHistory] h
    on p.BusinessEntityID = h.BusinessEntityID
    where @MaNV = p.BusinessEntityID OR @MaNV is null
    return
end
go

--thuc thi
declare @MaNV int
set @MaNV = 1
if @MaNV = null
    select * from SalaryOfEmp(null)
else
    select * from SalaryOfEmp(@MaNV)
go

```