

```
--Viết một stored procedure với Input là một mã khách hàng CustomerId và Output là một hóa đơn OrderId của khách hàng đó--  
--có Total Amount là nhỏ nhất và một hóa đơn OrderId của khách hàng đó có Total Amount là lớn nhất --
```

```
create procedure usp_GetOrderId_CustomerID_MaxAndMinTotalQuantity  
    @CustomerId int,  
    @MaxOrderId int output,  
    @MaxTotalAmount decimal(12,2) output,  
    @MinOrderId int output,  
    @MinTotalAmount decimal(12,2) output  
as  
begin  
    with OrderInfo(Id, CustomerId, TotalAmount)  
    as  
    (  
        select Id, CustomerId, TotalAmount  
        from [Order]  
        where CustomerId = @CustomerId  
    )  
    select @MaxOrderId = ( select top 1 Id from OrderInfo Order by TotalAmount DESC),  
        @MaxTotalAmount = MAX(TotalAmount),  
        @MinOrderId = (select top 1 Id from OrderInfo Order by TotalAmount ASC),  
        @MinTotalAmount = MIN(TotalAmount)  
    from OrderInfo  
END
```

```
declare @CustomerId1 int  
declare @MaxOrderId1 int  
declare @MaxTotalAmount1 decimal(18,2)  
declare @MinOrderId1 int  
declare @MinTotalAmount1 decimal(18,2)  
SET @CustomerId1 = 8  
exec usp_GetOrderId_CustomerID_MaxAndMinTotalQuantity @CustomerId1,  
    @MaxOrderId1 output, @MaxTotalAmount1 output , @MinOrderId1 output, @MinTotalAmount1 output  
select @CustomerId1 as CustomerID,  
    @MaxOrderId1 as MaxOrderId ,@MinOrderId1 as MinOrderId ,  
    @MaxTotalAmount1 as MaxTotalAmount, @MinTotalAmount1 as MinTotalAmount
```

Commands completed successfully.

Completion time: 2022-06-16T20:27:59.7311301407:00

100 %

Query executed successfully.

DESKTOP-DJT45HY\SQLEXPRESS... DESKTOP-DJT45HY\My Com... Northwind : 00:00:00 : 0 rows

```
--Viết một stored procedure để thêm vào một Customer với Input là firstName, LastName, City, Country, và Phone.--
--Lưu ý nếu các input mà rỗng hoặc Input đã có trong bảng thì báo lỗi tương ứng và ROLL BACK lại--
create procedure usp_NewCustomer
    @CustomerId int,
    @FirstName nvarchar(50),
    @LastName nvarchar(50),
    @City nvarchar(50),
    @Country nvarchar(30),
    @Phone nvarchar(30)
as
begin
    if(exists(select *
        from Customer
        where FirstName = @FirstName and LastName = @LastName and City = @City and Country = @Country and Phone = @Phone))
    begin
        print 'Customer already exist'
        return -1
    end
    if(len(@FirstName) = 0 or len(@LastName) = 0 or len(@City) = 0 or len(@Country) = 0 or len(@Phone) = 0)
    begin
        print 'Customer description can not be blank'
        return -1
    end
    begin try
        begin transaction
            set IDENTITY_INSERT Customer on
```

```
        insert into [dbo].[Customer]([Id],[FirstName],[LastName],[City],[Country],[Phone])
        values (@CustomerId, @FirstName, @LastName, @City, @Country, @Phone)
        set identity insert Customer off

        print 'Insert successful'

        commit transaction
    end try
    begin catch
        if @@TRANCOUNT > 0 rollback transaction
        declare @ERR nvarchar(max)
        set @ERR = ERROR_MESSAGE()
        print 'An error occurred after insert data to the Customer table: '
        raiserror(@ERR, 16, 1)
        return -1
    end catch
end

declare @StateInsert1 int
exec @StateInsert1 = usp_NewCustomer 92, 'Paul', 'Henriot', 'Reims', 'France', '26.47.15.18'
print @StateInsert1

declare @StateInsert2 int
exec @StateInsert2 = usp_NewCustomer 92, 'Bob', 'Nixon', 'Portland', 'USA', '{(871) 34 52 64 28}'
print @StateInsert2

select * from Customer
```

Commands completed successfully.

Completion time: 2022-06-16T20:38:43.6080674+07:00

100 %

Query executed successfully.

DESKTOP-DJT45HV\SQLEXPRESS ... DESKTOP-DJT45HV\My Com... Northwind 00:00:00 0 rows

```

--
create procedure usp_UpdatePriceOrderItem
    @OrderItemId int,
    @UnitPrice decimal(12,2)
as
begin
    if(not exists(select * from OrderItem where Id = @OrderItemId))
    begin
        print 'OrderItem does not exist'
        return -1
    end

    declare @OrderId int
    declare @TotalAmount decimal(12,2)

    set @OrderId = (select OrderId from OrderItem where Id = @OrderItemId)
    begin try
        begin transaction
            update OrderItem set UnitPrice = @UnitPrice where Id = @OrderItemId

            set @TotalAmount = (select top 1 sum(UnitPrice * Quantity) over (partition by OrderId
                from [OrderItem]
                where OrderId = @OrderId)
            update [Order] set TotalAmount = @TotalAmount where Id = @OrderId
            print 'Update successful!'
        commit transaction
    end try
    begin catch
        if @@TRANCOUNT > 0 rollback transaction
        declare @ERR nvarchar(744)
        set @ERR = ERROR_MESSAGE()

```

```

        print 'Update failed. See details: '
        raiserror(@ERR,16,1)
        return -1
    end catch
end

declare @StateInsert2 int
exec @StateInsert2 = usp_UpdatePriceOrderItem 2157 , 16.00
print @StateInsert2

declare @StateInsert1 int
exec @StateInsert1 = usp_UpdatePriceOrderItem 21 , 23.00
print @StateInsert1

select * from OrderItem where OrderId = 8
select * from [Order] where Id = 8

```

Commands completed successfully.

Completion time: 2022-06-16T20:42:02.3873460+07:00

100 %

Query executed successfully.

DESKTOP-DIT45HV\SQLEXPRESS ... DESKTOP-DIT45HV\My Com... Northwind | 00:00:00 | 0 rows