

Câu 1

```

8 create procedure usp_GetOrderId_CustomerID_MaxAndMinTotalQuantity
9     @CustomerId int,
10    @MaxOrderId int output,
11    @MaxTotalAmount decimal(12,2) output,
12    @MinOrderId int output,
13    @MinTotalAmount decimal(12,2) output
14 as
15 begin
16     with OrderInfo(Id, CustomerId, TotalAmount)
17     as
18     (
19         select Id, CustomerId, TotalAmount
20         from [Order]
21         where CustomerId = @CustomerId
22     )
23     select @MaxOrderId = (select top 1 Id from OrderInfo Order by TotalAmount DESC),
24            @MaxTotalAmount = MAX(TotalAmount),
25            @MinOrderId = (select top 1 Id from OrderInfo Order by TotalAmount ASC),
26            @MinTotalAmount = MIN(TotalAmount)
27     from OrderInfo
28 end
29
30 declare @CustomerId2 int
31 declare @MaxOrderId2 int
32 declare @MaxTotalAmount2 decimal (12,2)
33 declare @MinOrderId2 int
34 declare @MintotalAmount2 decimal(12,2)
35 SET @CustomerId2 = 21
36 exec usp_GetOrderId_CustomerID_MaxAndMinTotalQuantity @CustomerId2,
37    @MaxOrderId2 output, @MaxTotalAmount2 output, @MinOrderId2 output, @MintotalAmount2 output
38 select @CustomerId2 as CustomerID,
39        @MaxOrderId2 as MaxOrderID, @MinOrderId2 as MinOrderID,
40        @MaxTotalAmount2 as Maxtotalamount, @MintotalAmount2 as Mintotalamount
41

```

Câu 2

```

47 create procedure usp_InsertNewCustomer
48     @CustomerId int,
49     @FirstName nvarchar(50),
50     @LastName nvarchar(50),
51     @City nvarchar(50),
52     @Country nvarchar(30),
53     @Phone nvarchar(30)
54 as
55 begin
56     if(exists(select * from Customer where FirstName = @Firstname and LastName = @Lastname and City = @City and Country = @Country and Phone = @Phone))
57     begin
58         print 'customer already exist'
59         return -1
60     end
61     if(len(@FirstName) = 0 or len(@LastName) = 0 or len(@City)=0 or len(@Country)=0 or len(@Phone)=0)
62     begin
63         print 'Customer description can not be blank'
64         return -1
65     end
66
67     begin try
68         begin transaction
69         set IDENTITY_INSERT Customer on
70         insert into [dbo].[Customer]([Id],[FirstName],[LastName],[City],[Country],[Phone])
71         values (@CustomerId, @FirstName, @LastName, @City, @Country, @Phone)
72         set IDENTITY_INSERT Customer off
73         print 'Insert successful'
74         commit transaction
75     end try
76
77     begin catch
78         if @@TRANCOUNT > 0 rollback transaction
79         declare @ERR nvarchar(MAX)
80         set @ERR = ERROR_MESSAGE()
81         print 'An error occurred after insert data to the Customer table: '
82         raiserror(@ERR, 16,1)
83         return -1
84     end catch
85 end
86
87 declare @StateInsert1 int
88 exec @StateInsert1 = usp_InsertNewCustomer 92, 'Paul', 'Henriot', 'Reims', 'France', '26.47.15.10'
89 print @StateInsert1
90
91
92 declare @StateInsert2 int
93 exec @StateInsert2 = usp_InsertNewCustomer 92, 'Bob', 'Nixon', 'Portland', 'USA', '(071) 34 52 64 28'
94 print @StateInsert2

```

100 %

Messages

customer already exist

-1

customer already exist

-1

|

Câu 3

```

100 create procedure usp_UpdatePriceOrderItem
101     @OrderItemId int,
102     @UnitPrice decimal(12,2)
103 as
104 begin
105     if(not exists(select * from OrderItem where Id = @OrderItemId))
106     begin
107         print 'Order Item does not exist'
108         return -1
109     end
110
111     declare @OrderId int
112     declare @TotalAmount decimal(12,2)
113
114     set @OrderId =(select OrderId from OrderItem where Id = @OrderItemId)
115     begin try
116         begin transaction
117             update OrderItem set UnitPrice = @UnitPrice where Id = @OrderItemId
118
119             set @TotalAmount = (select top 1 sum(UnitPrice*Quantity) over (partition by OrderId)
120                                 from [OrderItem]
121                                 where OrderId = @OrderId)
122             update [Order] set TotalAmount = @TotalAmount where Id = @OrderId
123             print 'Update successful'
124             commit transaction
125         end try
126         begin catch
127             if @@TRANCOUNT > 0 rollback transaction
128             declare @ERR nvarchar(MAX)
129             set @ERR = ERROR_MESSAGE()
130             print 'Update failed. See details: '
131             raiserror(@ERR,16,1)
132             return -1
133         end catch
134     end


```

```

125     end try
126     begin catch
127         if @@TRANCOUNT > 0 rollback transaction
128         declare @ERR nvarchar(MAX)
129         set @ERR = ERROR_MESSAGE()
130         print 'Update failed. See detaail: '
131         raiserror(@ERR,16,1)
132         return -1
133     end catch
134 end
135
136 declare @StateInsert1 int
137 exec @StateInsert1 = usp_UpdatePriceOrderItem 21 ,23.00
138
139 print @StateInsert1
140
141
142
143 declare @StateInsert2 int
144 exec @StateInsert2 = usp_UpdatePriceOrderItem 2145,17.00
145
146 print @StateInsert2
147
148 |
149 select * from OrderItem where OrderId = 8
150 select * from [Order] where Id = 8

```

00 % <

 Messages

(1 row(s) affected)

(1 row(s) affected)

Update successful

0