Лабораторная работа 5

Задане 1

Создайте scalar-valued функцию, которая будет принимать в качестве входного параметра id заказа (Purchasing.PurchaseOrderHeader.PurchaseOrderID) и возвращать сумму по заказу из детализированного списка заказов (Purchasing.PurchaseOrderDetail.LineTotal).

```sql
CREATE FUNCTION [Purchasing].GetSumOfOrders(@PurchaseOrderID INT)
RETURNS MONEY
AS
BEGIN
    DECLARE @result MONEY
    SELECT @result = SUM([AdventureWorks2012].[Purchasing].
            [PurchaseOrderDetail].[LineTotal])
    FROM [AdventureWorks2012].[Purchasing].[PurchaseOrderDetail]
    WHERE [PurchaseOrderDetail].[PurchaseOrderID] = @PurchaseOrderID
    RETURN @result
END;
GO


PRINT [Purchasing].GetSumOfOrders(1);
```

```
AdventureWorks2012> PRINT [Purchasing].GetSumOfOrders(1)
[2020-11-12 23:15:02] [S0001] 201.04
[2020-11-12 23:15:02] completed in 5 ms
```

Задане 2

Создайте inline table-valued функцию, которая будет принимать в качестве входных параметров id заказчика (Sales.Customer.CustomerID) и количество строк, которые необходимо вывести.

Функция должна возвращать определенное количество самых прибыльных заказов (по TotalDue) из Sales.SalesOrderHeader для каждого заказчика.
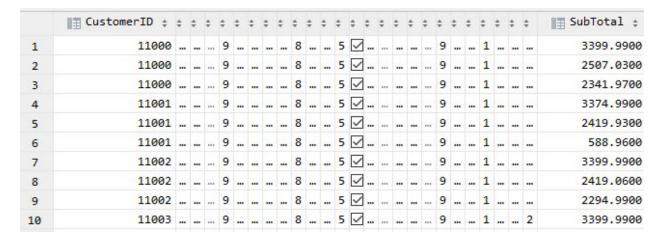
```sql
CREATE FUNCTION [Sales].GetBestOrdersByCustomerID(@CustomerID INT, @CountOfRow INT)
RETURNS TABLE AS RETURN (
    SELECT TOP(@CountOfRow)
        [SalesOrderID],
        [RevisionNumber],
        [OrderDate],
        [DueDate],
        [ShipDate],
        [Status],
        [OnlineOrderFlag],
        [SalesOrderNumber],
        [PurchaseOrderNumber],
        [AccountNumber],
        [CustomerID],
        [SalesPersonID],
        [TerritoryID],
        [BillToAddressID],
        [ShipToAddressID],
        [ShipMethodID],
        [CreditCardID],
        [CreditCardApprovalCode],
        [CurrencyRateID],
        [SubTotal],
        [TaxAmt],
        [Freight],
        [TotalDue],
        [Comment],
        [rowguid],
        [ModifiedDate]
```

```
  FROM [AdventureWorks2012].[Sales].[SalesOrderHeader]

  WHERE [CustomerID] = @CustomerID

  ORDER BY [TotalDue] DESC

);

GO
```
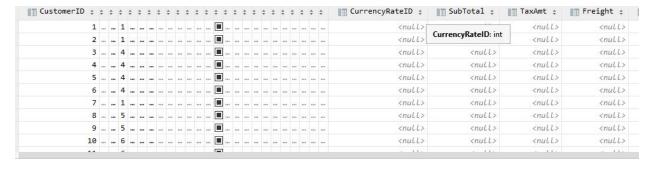
Задане 3

Вызовите функцию для каждого заказчика, применив оператор CROSS APPLY. Вызовите функцию для каждого заказчика, применив оператор OUTER APPLY.

```
SELECT * FROM [AdventureWorks2012].[Sales].[Customer] CROSS APPLY
[AdventureWorks2012].[Sales].[GetBestOrdersByCustomerID]([CustomerID], 3);
```

	CustomerID																										SubTotal
1	11000	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	3399.9900
2	11000	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	2507.0300
3	11000	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	2341.9700
4	11001	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	3374.9900
5	11001	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	2419.9300
6	11001	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	588.9600
7	11002	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	3399.9900
8	11002	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	2419.0600
9	11002	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	…	2294.9900
10	11003	…	…	…	9	…	…	…	…	8	…	…	5	☑	…	…	…	…	…	9	…	…	1	…	…	2	3399.9900

```
SELECT * FROM [AdventureWorks2012].[Sales].[Customer] OUTER APPLY
[AdventureWorks2012].[Sales].[GetBestOrdersByCustomerID]([CustomerID], 3);
```

	CustomerID																							CurrencyRateID	SubTotal	TaxAmt	Freight
	1	…	…	1	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	2	…	…	1	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null> CurrencyRateID: int	<null>	<null>	<null>
	3	…	…	4	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	4	…	…	4	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	5	…	…	4	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	6	…	…	4	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	7	…	…	1	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	8	…	…	5	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	9	…	…	5	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>
	10	…	…	6	…	…	…	…	…	…	…	…	…	■	…	…	…	…	…	…	…	…	…	<null>	<null>	<null>	<null>

Задане 4

Измените созданную inline table-valued функцию, сделав ее multistatement table-valued

(предварительно сохранив для проверки код создания inline table-valued функции).

```sql
CREATE FUNCTION [Sales].GetBestOrdersByCustomerID(@CustomerID INT, @CountOfRow INT)
RETURNS @result TABLE(
  [SalesOrderID] INT NOT NULL,
  [RevisionNumber] TINYINT NOT NULL,
  [OrderDate] DATETIME NOT NULL,
  [DueDate] DATETIME NOT NULL,
  [ShipDate] DATETIME NULL,
  [Status] TINYINT NOT NULL,
  [OnlineOrderFlag] dbo.Flag NOT NULL,
  [SalesOrderNumber] NVARCHAR(23),
  [PurchaseOrderNumber] dbo.OrderNumber NULL,
  [AccountNumber] dbo.AccountNumber NULL,
  [CustomerID] INT NOT NULL,
  [SalesPersonID] INT NULL,
  [TerritoryID] INT NULL,
  [BillToAddressID] INT NOT NULL,
  [ShipToAddressID] INT NOT NULL,
  [ShipMethodID] INT NOT NULL,
  [CreditCardID] INT NULL,
  [CreditCardApprovalCode] VARCHAR(15) NULL,
  [CurrencyRateID] INT NULL,
  [SubTotal] MONEY NOT NULL ,
  [TaxAmt] MONEY NOT NULL,
  [Freight] MONEY NOT NULL,
  [TotalDue] INT NOT NULL,
  [Comment] NVARCHAR(128) NULL,
  [rowguid] UNIQUEIDENTIFIER ROWGUIDCOL  NOT NULL,
  [ModifiedDate] DATETIME NOT NULL
) AS BEGIN
  INSERT INTO @result
  SELECT TOP(@CountOfRow)
    [SalesOrderID],
    [RevisionNumber],
    [OrderDate],
    [DueDate],
    [ShipDate],
    [Status],
    [OnlineOrderFlag],
    [SalesOrderNumber],
    [PurchaseOrderNumber],
    [AccountNumber],
```

```sql
        [CustomerID],
        [SalesPersonID],
        [TerritoryID],
        [BillToAddressID],
        [ShipToAddressID],
        [ShipMethodID],
        [CreditCardID],
        [CreditCardApprovalCode],
        [CurrencyRateID],
        [SubTotal],
        [TaxAmt],
        [Freight],
        [TotalDue],
        [Comment],
        [rowguid],
        [ModifiedDate]
    FROM [AdventureWorks2012].[Sales].[SalesOrderHeader]
    WHERE [CustomerID] = @CustomerID
    ORDER BY [TotalDue] DESC
    RETURN
END;
```