



20280083_LaiToanThang_DBM_Lab06

```
USE Northwind;
--1. Viết hàm truyền vào một CustomerId và xuất ra tổng giá tiền (Total Amount) của các hóa đơn từ khách hàng đó.
--Sau đó dùng hàm này xuất ra tổng giá tiền từ các hóa đơn của tất cả khách hàng

DROP FUNCTION dbo.GetTotalAmountByCustomerId

CREATE FUNCTION dbo.GetTotalAmountByCustomerId
(
    @CustomerId INT
)
RETURNS DECIMAL(12, 2)
AS
BEGIN
    DECLARE @TotalAmount DECIMAL(12, 2);

    SELECT @TotalAmount = SUM(O.TotalAmount)
    FROM [Order] O
    WHERE O.CustomerId = @CustomerId;

    RETURN @TotalAmount;
END;

SELECT *,
    dbo.GetTotalAmountByCustomerId(Id) AS TotalAmount
FROM Customer
```

	Id	FirstName	LastName	City	Country	Phone	TotalAmount
1	1	Maria	Anders	Berlin	Germany	030-0074321	4596.20
2	2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729	1402.95
3	3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932	7515.35
4	4	Thomas	Hardy	London	UK	(171) 555-7788	13806.50
5	5	Christina	Berglund	Luleå	Sweden	0921-12 34 65	26968.15
6	6	Hanna	Moos	Mannheim	Germany	0621-08460	3239.80
7	7	Frédérique	Citeaux	Strasbourg	France	88.60.15.31	19088.00
8	8	Martín	Sommer	Madrid	Spain	(91) 555 22 82	5297.80
9	9	Laurence	Lebihan	Marseille	France	91.24.45.40	23850.95
10	10	Elizabeth	Lincoln	Tsawassen	Canada	(604) 555-4729	22607.70
11	11	Victoria	Ashworth	London	UK	(171) 555-1212	6089.90
12	12	Patricio	Simpson	Buenos Aires	Argentina	(1) 135-5555	1814.80
13	13	Francisco	Chang	México D.F.	Mexico	(5) 555-3392	100.80
14	14	Yang	Wang	Bern	Switzerland	0452-076545	12886.30
15	15	Pedro	Afonso	Sao Paulo	Brazil	(11) 555-7647	3810.75
16	16	Elizabeth	Brown	London	UK	(171) 555-2282	1719.10
17	17	Sven	Ottlieb	Aachen	Germany	0241-039123	3763.21
18	18	Janine	Labruno	Nantes	France	40.67.88.88	1615.90
19	19	Ann	Devon	London	UK	(171) 555-0297	15033.66

(91 rows affected)

```
--2. Viết hàm truyền vào hai số và xuất ra danh sách các sản phẩm có UnitPrice nằm trong khoảng hai số đó.
CREATE FUNCTION GetProductByUnitPriceRange
(
    @MinPrice DECIMAL(12, 2),
    @MaxPrice DECIMAL(12, 2)
)
```

```

RETURNS TABLE
AS
RETURN
(
    SELECT *
    FROM Product
    WHERE UnitPrice >= @MinPrice
    AND UnitPrice <= @MaxPrice
);

SELECT * FROM GetProductByUnitPriceRange(10, 20)

```

	Id	ProductName	SupplierId	UnitPrice	Package	IsDiscontinued
1	1	Chai	1	18.00	10 boxes x 20 bags	0
2	2	Chang	1	19.00	24 - 12 oz bottles	0
3	3	Aniseed Syrup	1	10.00	12 - 550 ml bottles	0
4	15	Genen Shouyu	6	15.50	24 - 250 ml bottles	0
5	16	Pavlova	7	17.45	32 - 500 g boxes	0
6	21	Sir Rodney's Scones	8	10.00	24 pkgs. x 4 pieces	0
7	25	NuNuCa Nuß-Nougat-Creme	11	14.00	20 - 450 g glasses	0
8	31	Gorgonzola Telino	14	12.50	12 - 100 g pkgs	0
9	34	Sasquatch Ale	16	14.00	24 - 12 oz bottles	0
10	35	Steeleye Stout	16	18.00	24 - 12 oz bottles	0
11	36	Inlagd Sill	17	19.00	24 - 250 g jars	0
12	39	Chartreuse verte	18	18.00	750 cc per bottle	0
13	40	Boston Crab Meat	19	18.40	24 - 4 oz tins	0
14	42	Singaporean Hokkien Fried Mee	20	14.00	32 - 1 kg pkgs.	1
15	44	Gula Malacca	20	19.45	20 - 2 kg bags	0
16	46	Spegesild	21	12.00	4 - 450 g glasses	0
17	48	Chocolade	22	12.75	10 pkgs.	0
18	49	Maxilaku	23	20.00	24 - 50 g pkgs.	0
19	50	Valkoinen suklaa	23	16.25	12 - 100 g bars	0
20	57	Ravioli Angelo	26	19.50	24 - 250 g pkgs.	0
21	58	Escargots de Bourgogne	27	13.25	24 pieces	0
22	66	Louisiana Hot Spiced Okra	2	17.00	24 - 8 oz jars	0

(29 rows affected)

```

--3. Viết hàm truyền vào một danh sách các tháng 'June;July;August;September' và xuất ra thông tin của các hóa đơn
-- có trong những tháng đó. Viết cả hai hàm dưới dạng inline và multi statement sau đó cho biết thời gian thực thi của mỗi hàm,
-- so sánh và đánh giá

```

```

-- INLINE FUNCTION

CREATE FUNCTION GetOrdersByMonths_Inline(@MonthsList VARCHAR(MAX))
RETURNS TABLE
AS
RETURN
(
    SELECT *
    FROM [Order]
    WHERE DATENAME(MONTH, OrderDate) IN (
        SELECT value
        FROM STRING_SPLIT(@MonthsList, ';')
    )
)

SELECT * FROM GetOrdersByMonths_Inline('June;July;August;September')

```

	Id	OrderDate	OrderNumber	CustomerId	TotalAmount
1	23	2012-08-01 00:00:00.000	542400	87	1376.00
2	24	2012-08-01 00:00:00.000	542401	75	48.00
3	25	2012-08-02 00:00:00.000	542402	65	1456.00
4	26	2012-08-05 00:00:00.000	542403	63	2142.40
5	27	2012-08-06 00:00:00.000	542404	85	538.60
6	28	2012-08-07 00:00:00.000	542405	49	307.20
7	29	2012-08-08 00:00:00.000	542406	80	420.00
8	30	2012-08-09 00:00:00.000	542407	52	1200.80
9	31	2012-08-12 00:00:00.000	542408	5	1488.80
10	32	2012-08-13 00:00:00.000	542409	44	468.00
11	33	2012-08-14 00:00:00.000	542410	5	613.20
12	34	2012-08-14 00:00:00.000	542411	69	86.50
13	35	2012-08-15 00:00:00.000	542412	69	155.40
14	36	2012-08-16 00:00:00.000	542413	46	1414.80
15	37	2012-08-19 00:00:00.000	542414	44	1452.00
16	38	2012-08-20 00:00:00.000	542415	63	2179.20
17	39	2012-08-21 00:00:00.000	542416	63	3016.00
18	40	2012-08-22 00:00:00.000	542417	67	924.00
19	41	2012-08-23 00:00:00.000	542418	66	89.00
20	42	2012-08-26 00:00:00.000	542419	11	479.40
21	43	2012-08-27 00:00:00.000	542420	15	2169.00
22	44	2012-08-27 00:00:00.000	542421	61	552.80

(203 rows affected)

```
-- MULTI STATEMENT FUNCTION
DROP FUNCTION GetOrdersByMonths_MultiStatement

CREATE FUNCTION GetOrdersByMonths_MultiStatement(@MonthList VARCHAR(MAX))
RETURNS @ResultTable
TABLE(
    Id INT,
    OrderDate DATE,
    OrderNumber NVARCHAR(10),
    CustomerId INT,
    TotalAmount DECIMAL(12,2)
)
AS
BEGIN
    SET @MonthList = LOWER(@MonthList)

    INSERT INTO @ResultTable
    SELECT *
    FROM [Order]
    WHERE CHARINDEX(LTRIM(RTRIM(LOWER(DATENAME(mm, OrderDate)))), LOWER(@MonthList)) > 0
    RETURN
END

SELECT *
FROM GetOrdersByMonths_MultiStatement('June;July;August;September')
```

	Id	OrderDate	OrderNumber	CustomerId	TotalAmount
1	1	2012-07-04	542378	85	440.00
2	2	2012-07-05	542379	79	1863.40
3	3	2012-07-08	542380	34	1813.00
4	4	2012-07-08	542381	84	670.80
5	5	2012-07-09	542382	76	3730.00
6	6	2012-07-10	542383	34	1444.80
7	7	2012-07-11	542384	14	625.20
8	8	2012-07-12	542385	68	2490.50
9	9	2012-07-15	542386	88	517.80
10	10	2012-07-16	542387	35	1119.90
11	11	2012-07-17	542388	20	2018.60
12	12	2012-07-18	542389	13	100.80
13	13	2012-07-19	542390	56	1746.20
14	14	2012-07-19	542391	61	448.00
15	15	2012-07-22	542392	65	624.80
16	16	2012-07-23	542393	20	2464.80
17	17	2012-07-24	542394	24	724.50
18	18	2012-07-25	542395	7	1176.00
19	19	2012-07-26	542396	87	364.80
20	20	2012-07-29	542397	25	4031.00
21	21	2012-07-30	542398	33	1101.20
22	22	2012-07-31	542399	89	676.00
23	23	2012-08-01	542400	87	1376.00
24	24	2012-08-01	542401	75	48.00
25	25	2012-08-02	542402	65	1456.00
26	26	2012-08-05	542403	63	2142.40
27	27	2012-08-06	542404	85	538.60

(203 rows affected)

```
SET STATISTICS TIME ON
SELECT * FROM GetOrdersByMonths_Inline('June;July;August;September');
SELECT * FROM GetOrdersByMonths_MultiStatement('June;July;August;September');
SET STATISTICS TIME OFF
```

(203 rows affected)

```
SQL Server Execution Times:
    CPU time = 0 ms,  elapsed time = 2 ms.
SQL Server parse and compile time:
    CPU time = 0 ms,  elapsed time = 0 ms.
```

(203 rows affected)

```
SQL Server Execution Times:
    CPU time = 5 ms,  elapsed time = 39 ms.
```



Đánh giá:

Hàm Inline có thể nhanh hơn vì nó được tối ưu hóa để thực hiện thi trực tiếp trong câu lệnh.

Hàm Multi-Statement phải tạo bảng biến đổi và chèn dữ liệu trước khi return kết quả

```
--4. Viết hàm kiểm tra mỗi hóa đơn không có quá 5 sản phẩm (kiểm tra trong bảng OrderItem).
--Nếu insert quá 5 sản phẩm cho một hóa đơn thì báo lỗi và không cho insert.
SELECT * FROM OrderItem
```

```

DROP FUNCTION ufn_CheckNumberProductOfOrder

CREATE FUNCTION ufn_CheckNumberProductOfOrder(@OrderId INT)
RETURNS BIT
AS
BEGIN
    DECLARE @Status BIT, @Count INT;
    SET @Count = (SELECT COUNT(ProductId) FROM OrderItem WHERE OrderId = @OrderId);
    IF( @Count <= 5)
        SET @Status = 1;
    ELSE
        SET @Status = 0;

    RETURN @Status;
END

```

```

DROP TRIGGER IF EXISTS CheckNumberProductOfOrderTrigger;

CREATE TRIGGER CheckNumberProductOfOrderTrigger
ON OrderItem
AFTER INSERT
AS
BEGIN
    DECLARE @OrderId INT;
    SELECT @OrderId = OrderId from inserted;
    IF EXISTS (SELECT 1 FROM inserted WHERE dbo.ufn_CheckNumberProductOfOrder(@OrderId) = 0)
    BEGIN
        RAISERROR('CAN NOT INSERT, NUMBER OF PRODUCT IS AT MOST 5 ITEMS', 16, 1);
        ROLLBACK TRANSACTION
    END
END

INSERT INTO OrderItem VALUES(26 , 12, 7, 3);

```

```

Msg 50000, Level 16, State 1, Procedure CheckNumberProductOfOrderTrigger, Line 11 [Batch Start Line 165]
CAN NOT INSERT, NUMBER OF PRODUCT IS AT MOST 5 ITEMS
Msg 3609, Level 16, State 1, Line 166
The transaction ended in the trigger. The batch has been aborted.

```