



20280083_LaiToanThang_DBM_Lab08

```
--1. 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
-- (Ví dụ: EXEC usp_GetOrderId_CustomerId_MaxAndMinTotalQuantity @CustomerId,@MaxOrderId OUTPUT,
-- @MaxTotalAmount OUTPUT, @MinOrderId OUTPUT, @MinTotalAmount OUTPUT)
```

```
SELECT * FROM [Order]

DROP PROCEDURE sp_GetOrderId_MaxAndMinTotalAmount
CREATE PROCEDURE sp_GetOrderId_MaxAndMinTotalAmount
    @CustomerId INT,
    @MaxOrderId INT OUTPUT,
    @MaxTotalAmount DECIMAL(12, 2) OUTPUT,
    @MinOrderId INT OUTPUT,
    @MinTotalAmount DECIMAL(12, 2) OUTPUT
AS
BEGIN
    -- Trả về OrderId và TotalAmount với TotalAmount cao nhất cho CustomerId đầu vào
    SELECT TOP 1 @MaxOrderId = Id, @MaxTotalAmount = TotalAmount
    FROM [Order]
    WHERE CustomerId = @CustomerId
    ORDER BY TotalAmount DESC;

    -- Trả về OrderId và TotalAmount với TotalAmount thấp nhất cho CustomerId đầu vào
    SELECT TOP 1 @MinOrderId = Id, @MinTotalAmount = TotalAmount
    FROM [Order]
    WHERE CustomerId = @CustomerId
    ORDER BY TotalAmount ASC;
END
```

```
-- Chạy stored procedure này với CustomerId là 11 xem kết quả hóa đơn nào mà
-- CustomerId 11 có TotalAmount là lớn nhất và nhỏ nhất
```

```
DECLARE @MinOrderId INT;
DECLARE @MinTotalAmount DECIMAL(12, 2);
DECLARE @MaxOrderId INT;
DECLARE @MaxTotalAmount DECIMAL(12, 2);

EXEC sp_GetOrderId_TotalAmount_MinAndMax @CustomerId = 11,
    @MinOrderId = @MinOrderId OUTPUT,
    @MinTotalAmount = @MinTotalAmount OUTPUT,
    @MaxOrderId = @MaxOrderId OUTPUT,
    @MaxTotalAmount = @MaxTotalAmount OUTPUT;

SELECT 'Min' AS Type, @MinOrderId AS OrderId, @MinTotalAmount AS TotalAmount
UNION ALL
SELECT 'Max' AS Type, @MaxOrderId AS OrderId, @MaxTotalAmount AS TotalAmount;
```

Results		Messages	
	Type	OrderId	TotalAmount
1	Min	291	139.80
2	Max	776	1500.00

--2. 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 đó đã có trong bảng thì báo lỗi tương ứng và ROLL BACK lại

```

CREATE PROCEDURE sp_InsertCustomer
    @FirstName NVARCHAR(50),
    @LastName NVARCHAR(50),
    @City NVARCHAR(50),
    @Country NVARCHAR(50),
    @Phone NVARCHAR(20)
AS
BEGIN
    -- Nếu các input mà rỗng
    IF @FirstName IS NULL OR @FirstName = ''
        OR @LastName IS NULL OR @LastName = ''
        OR @City IS NULL OR @City = ''
        OR @Country IS NULL OR @Country = ''
        OR @Phone IS NULL OR @Phone = ''
    BEGIN
        PRINT N'Vui lòng cung cấp thông tin đầy đủ của khách hàng!';
        RETURN -1;
    END

    -- Nếu Input đó đã có trong bảng
    IF EXISTS(
        SELECT * FROM Customer
        WHERE @FirstName = FirstName
        AND @LastName = LastName
        AND @Phone = Phone
    )
    BEGIN
        PRINT N'Khách hàng đã tồn tại trong hệ thống'
        RETURN -1
    END

    -- Nếu các kiểm tra trên được thực hiện thành công
    BEGIN TRY
        BEGIN TRANSACTION
            -- Thêm thông tin khách hàng vào
            INSERT INTO Customer(FirstName, LastName, City, Country, Phone)
            VALUES (@FirstName, @LastName, @City, @Country, @Phone)
            -- Nếu câu lệnh chèn được thực thi thành công,
            -- transaction sẽ được xác nhận bằng cách sử dụng COMMIT TRANSACTION,
            -- và thông tin khách hàng mới sẽ được thêm vào cơ sở dữ liệu.
            COMMIT TRANSACTION
        END TRY
        -- Trong trường hợp xảy ra lỗi trong quá trình chèn, thủ tục sẽ vào khối CATCH
        BEGIN CATCH
            IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;
            DECLARE @ErrorMessage NVARCHAR(MAX);
            SET @ErrorMessage = ERROR_MESSAGE();

```

```

PRINT N'Có lỗi xảy ra khi thêm khách hàng: ' + @ErrorMessage;
RAISERROR (@ErrorMessage, 16, 1);
RETURN -1
END CATCH
END

```

```

-- Giả sử thêm khách hàng (Chưa có khách hàng này).
-- Nhưng điền thông tin không đủ
-- Chương trình sẽ báo lỗi và ROLL BACK lại

DECLARE @StateInsert INT
EXEC @StateInsert = sp_InsertCustomer 'Thang', 'Lai', NULL, 'TP HCM', '0123456789'
PRINT(@StateInsert)

```

Messages

Vui lòng cung cấp thông tin đầy đủ của khách hàng
-1

```

-- Giả sử thêm khách hàng.
-- Nhưng thông tin đã có trong cơ sở dữ liệu
-- Chương trình sẽ báo lỗi và ROLL BACK lại

DECLARE @StateInsert INT
EXEC @StateInsert = sp_InsertCustomer 'Maria', 'Anders', 'Berlin', 'Germany', '030-0074321'
PRINT(@StateInsert)

```

Messages

Khách hàng đã tồn tại trong hệ thống
-1

```

-- Giả sử thêm khách hàng (Chưa có khách hàng này).
-- Với đầy đủ thông tin như sau

DECLARE @StateInsert INT
EXEC @StateInsert = sp_InsertCustomer 'Thang', 'Lai', 'Viet Nam', 'TP HCM', '0123456789'
PRINT(@StateInsert)

```

Messages

(1 row affected)
0

```
SELECT * FROM [Customer] WHERE FirstName = 'Thang'
```

Results

Messages

	Id	FirstName	LastName	City	Country	Phone
1	92	Thang	Lai	Viet Nam	TP HCM	0123456789

```

--3. Viết Store Procedure cập nhật lại UnitPrice của sản phẩm trong bảng OrderItem.
-- Khi cập nhật lại UnitPrice này thì cũng phải cập nhật lại Total Amount trong bảng Order

```

```
-- tương ứng với Total Amount = SUM (UnitPrice *Quantity)
```

```
CREATE PROCEDURE sp_UpdateUnitPriceAndTotalAmount
    @ProductId INT,
    @NewUnitPrice DECIMAL(12, 2)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;
        -- Cập nhật UnitPrice của sản phẩm trong bảng OrderItem
        UPDATE OrderItem
        SET UnitPrice = @NewUnitPrice
        WHERE ProductId = @ProductId;

        -- Cập nhật lại TotalAmount trong bảng Order tương ứng với TotalAmount = SUM(UnitPrice * Quantity)
        UPDATE [Order]
        SET TotalAmount = (
            SELECT SUM(UnitPrice * Quantity)
            FROM OrderItem
            WHERE OrderId = [Order].Id
        )
        WHERE Id IN (
            SELECT OrderId
            FROM OrderItem
            WHERE ProductId = @ProductId
        );

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION; -- kiểm tra xem có đang tồn tại một giao dịch đang chạy hay không
        DECLARE @ErrorMessage NVARCHAR(MAX);
        SET @ErrorMessage = ERROR_MESSAGE();

        PRINT N'Có lỗi xảy ra khi cập nhật: ' + @ErrorMessage;
        RAISERROR (@ErrorMessage, 16, 1);
        RETURN -1
    END CATCH;
END
```

```
-- Execute stored procedure để update UnitPrice và TotalAmount
EXEC sp_UpdateUnitPriceAndTotalAmount @ProductId = 7, @NewUnitPrice = 35;

-- Kiểm tra kết quả cập nhật bằng cách truy vấn dữ liệu
SELECT * FROM OrderItem WHERE ProductId = 7;

SELECT * FROM [Order]
WHERE Id IN (
    SELECT OrderId FROM OrderItem
    WHERE ProductId = 7
);
```

- TRƯỚC KHI UPDATE
- SAU KHI UPDATE

Results Messages					
	Id	OrderId	ProductId	UnitPrice	Quantity
1	42	15	7	24.00	15
2	363	138	7	24.00	10
3	562	212	7	24.00	16
4	595	224	7	24.00	30
5	688	264	7	30.00	50
6	796	299	7	30.00	10
7	911	344	7	30.00	10
8	948	360	7	30.00	45
9	1018	387	7	30.00	35
10	1172	447	7	30.00	90
11	1367	519	7	30.00	35

	Id	OrderDate	OrderNumber	CustomerId	TotalAmount
1	15	2012-07-22 00:00:00.000	542392	65	624.80
2	138	2012-12-17 00:00:00.000	542515	75	864.00
3	212	2013-02-27 00:00:00.000	542589	84	1688.00
4	224	2013-03-11 00:00:00.000	542601	11	1328.00
5	264	2013-04-18 00:00:00.000	542641	9	3000.00
6	299	2013-05-23 00:00:00.000	542676	84	2812.00
7	344	2013-07-07 00:00:00.000	542721	83	812.50
8	360	2013-07-22 00:00:00.000	542737	71	6475.40
9	387	2013-08-15 00:00:00.000	542764	23	4985.50
10	447	2013-10-06 00:00:00.000	542824	63	4825.00
11	519	2013-12-05 00:00:00.000	542896	56	2310.00
12	543	2013-12-22 00:00:00.000	542920	31	850.00
13	586	2014-01-15 00:00:00.000	542963	56	1007.70
14	611	2014-01-29 00:00:00.000	542988	40	649.00
15	661	2014-02-26 00:00:00.000	543038	66	698.00
16	662	2014-02-26 00:00:00.000	543039	70	670.00
17	693	2014-03-11 00:00:00.000	543070	9	360.00

Results Messages					
	Id	OrderId	ProductId	UnitPrice	Quantity
1	42	15	7	35.00	15
2	363	138	7	35.00	10
3	562	212	7	35.00	16
4	595	224	7	35.00	30
5	688	264	7	35.00	50
6	796	299	7	35.00	10
7	911	344	7	35.00	10
8	948	360	7	35.00	45
9	1018	387	7	35.00	35
10	1172	447	7	35.00	90
11	1367	519	7	35.00	35
12	1421	543	7	35.00	3
13	1529	586	7	35.00	20
14	1602	611	7	35.00	5

	Id	OrderDate	OrderNumber	CustomerId	TotalAmount
1	15	2012-07-22 00:00:00.000	542392	65	789.80
2	138	2012-12-17 00:00:00.000	542515	75	974.00
3	212	2013-02-27 00:00:00.000	542589	84	1864.00
4	224	2013-03-11 00:00:00.000	542601	11	1658.00
5	264	2013-04-18 00:00:00.000	542641	9	3250.00
6	299	2013-05-23 00:00:00.000	542676	84	2862.00
7	344	2013-07-07 00:00:00.000	542721	83	862.50
8	360	2013-07-22 00:00:00.000	542737	71	6700.40
9	387	2013-08-15 00:00:00.000	542764	23	5160.50
10	447	2013-10-06 00:00:00.000	542824	63	5275.00
11	519	2013-12-05 00:00:00.000	542896	56	2485.00
12	543	2013-12-22 00:00:00.000	542920	31	865.00
13	586	2014-01-15 00:00:00.000	542963	56	1107.70
14	611	2014-01-29 00:00:00.000	542988	40	674.00