



20280083_LaiToanThang_DBM_Lab07

```
--1. Trigger:
--a. Viết trigger khi xóa một OrderId thì xóa luôn các thông tin của Order đó trong bảng OrderItem.
--Nếu có Foreign Key Constraint xảy ra không cho xóa thì hãy xóa Foreign Key Constraint đó đi rồi thực thi.

CREATE TRIGGER [dbo].[Trigger_OrderIdDelete]
ON [dbo].[OrderItem]
FOR DELETE
AS
BEGIN
    DECLARE @DeleteOrderId INT
    SELECT @DeleteOrderId = OrderId FROM deleted

    DELETE FROM OrderItem
    WHERE OrderId = @DeleteOrderId

    PRINT 'Cac thong tin cua Order co OrderId la ' + LTRIM(STR(@DeleteOrderId)) + ' trong bang OrderItem da bi xoa'

    IF EXISTS (
        SELECT 1
        FROM sys.foreign_keys
        WHERE name = 'FK_ORDERITE_REFERENCE_Order' AND OBJECT_NAME(parent_object_id) = 'OrderItem'
    )
    BEGIN
        ALTER TABLE OrderItem DROP CONSTRAINT FK_ORDERITE_REFERENCE_Order
        PRINT 'Foreign Key Constraint FK_ORDERITE_REFERENCE_Order đã bị xóa'
    END
END
```

```
-- Trước khi delete OrderId
SELECT *
FROM OrderItem
WHERE OrderId = 21;
```

	Id	OrderId	ProductId	UnitPrice	Quantity
1	110	42	3	8.00	30
2	111	42	64	26.60	9

```
-- Liệt kê tất cả các trigger đang tồn tại trong cơ sở dữ liệu
SELECT name, type_desc
FROM sys.triggers
WHERE parent_class_desc = 'OBJECT_OR_COLUMN';

-- Kiểm tra kết quả trigger
DELETE FROM [OrderItem]
WHERE OrderId = 21;
```

(0 rows affected)

Cac thong tin cua Order co OrderId la 42 trong bang OrderItem da bi xoa

(2 rows affected)

```
--b. Viết trigger khi xóa hóa đơn của khách hàng Id = 1 thì báo lỗi không cho xóa sau đó ROLL BACK lại.
--Lưu ý: Đưa trigger này lên làm Trigger đầu tiên thực thi xóa dữ liệu trên bảng Order

-- Cách 1:
CREATE TRIGGER [dbo].[Trigger_DeleteExceptCustomerId] -- Tạo trigger
ON [dbo].[Order] -- trên bảng Order trong schema dbo
AFTER DELETE -- trigger này được thực hiện sau khi có thao tác delete
AS
BEGIN
    SET NOCOUNT ON; -- khi thực thi, câu lệnh DELETE không gửi lại thông báo số hàng bị tác động (tức là số hàng bị xóa) cho ứng dụng hoặc

    DECLARE @DeleteCustomerId INT -- lưu trữ giá trị CustomerId của đơn hàng bị xóa.
    SELECT @DeleteCustomerId = CustomerId FROM deleted; -- Lấy giá trị CustomerId của đơn hàng bị xóa từ bảng "deleted"

    -- Kiểm tra nếu CustomerId là 1
    IF (@DeleteCustomerId = 1)
    BEGIN
        -- Nếu CustomerId là 1, phát sinh lỗi và hủy bỏ thao tác xóa
        RAISERROR ('Không thể xóa hóa đơn của khách hàng có Id = 1', 16, 1);
        ROLLBACK TRANSACTION; -- Hủy bỏ thao tác xóa hiện tại
        RETURN; -- Kết thúc trigger.
    END;

    -- Xóa các thông tin của đơn hàng trong bảng OrderItem liên quan đến đơn hàng bị xóa
    DELETE FROM OrderItem WHERE OrderId IN (SELECT OrderId FROM deleted);

    -- Kiểm tra và xóa Foreign Key Constraint nếu tồn tại
    IF OBJECT_ID('FK_ORDERITE_REFERENCE_ORDER', 'F') IS NOT NULL
    BEGIN
        ALTER TABLE OrderItem DROP CONSTRAINT FK_ORDERITE_REFERENCE_ORDER; -- Nếu tồn tại, xóa Foreign Key Constraint "FK_ORDERITE_REFERENC
        PRINT 'Foreign Key Constraint FK_ORDERITE_REFERENCE_ORDER đã bị xóa'; -- Xuất thông báo xóa Foreign Key Constraint
    END;
END;
```

```
-- Kiểm tra kết quả
DELETE FROM [Order] WHERE CustomerId = 1
```

Msg 50000, Level 16, State 1, Procedure Trigger_DeleteExceptCustomerId, Line 15 [Batch Start Line 85]
 Không thể xóa hóa đơn của khách hàng có Id = 1
 Msg 3609, Level 16, State 1, Line 86
 The transaction ended in the trigger. The batch has been aborted.

```
-- Cách 2:
CREATE TRIGGER [dbo].[Trigger_CustomerIdOrder1Delete]
ON [dbo].[Order]
FOR DELETE
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        DECLARE @DeleteCustomerId INT
        SELECT @DeleteCustomerId = CustomerId FROM deleted

        IF (@DeleteCustomerId = 1)
        BEGIN
            RAISERROR('Không thể xóa hóa đơn của khách hàng có Id = 1', 16, 1);
            ROLLBACK TRANSACTION;
            RETURN;
        END

        -- Xóa các thông tin của Order trong bảng OrderItem liên quan đến đơn hàng bị xóa
        DELETE FROM OrderItem WHERE OrderId IN (SELECT OrderId FROM deleted);

        -- Xóa Foreign Key Constraint nếu tồn tại
        IF EXISTS (
            SELECT 1
            FROM sys.foreign_keys
            WHERE name = 'FK_ORDERITE_REFERENCE_ORDER' AND OBJECT_NAME(parent_object_id) = 'OrderItem'
        )
        BEGIN
            ALTER TABLE OrderItem DROP CONSTRAINT FK_ORDERITE_REFERENCE_ORDER;
        END
    END TRY
    BEGIN CATCH
        -- Xử lý lỗi
    END CATCH
END;
```

```

        PRINT 'Foreign Key Constraint FK_ORDERITE_REFERENCE_ORDER đã bị xóa';
    END
END TRY
BEGIN CATCH
    DECLARE @ErrorMessage NVARCHAR(MAX), @ErrorSeverity INT, @ErrorState INT;
    SELECT
        @ErrorMessage = ERROR_MESSAGE(),
        @ErrorSeverity = ERROR_SEVERITY(),
        @ErrorState = ERROR_STATE();

    -- Ghi lại thông báo lỗi
    RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
END CATCH;
END;

```

```

-- Kiểm tra kết quả
DELETE FROM [Order] WHERE CustomerId = 1

```

Msg 50000, Level 16, State 1, Procedure Trigger_CustomerIdOrderDelete, Line 41 [Batch Start Line 87]
 Không thể xóa hóa đơn của khách hàng có Id = 1
 Msg 3616, Level 16, State 1, Line 88
 An error was raised during trigger execution. The batch has been aborted and the user transaction, if any, has been rolled back.

```

--c. Viết trigger không cho phép cập nhật Phone là NULL hay trong Phone có chữ cái ở bảng Supplier.
--Nếu có thì báo lỗi và ROLL BACK lại

```

```

CREATE TRIGGER [dbo].[Trigger_PhoneUpdate]
ON [dbo].[Supplier]
FOR UPDATE
AS
BEGIN
    -- Khai báo biến lưu trữ giá trị mới của trường "Phone"
    DECLARE @UpdatedPhone NVARCHAR(MAX)

    -- Kiểm tra xem trường "Phone" có được cập nhật trong thao tác UPDATE hay không
    IF UPDATE(Phone)
    BEGIN
        -- Lấy giá trị mới của trường "Phone" từ bảng "inserted"
        SELECT @UpdatedPhone = Phone FROM inserted

        -- Kiểm tra nếu giá trị mới của trường "Phone" là NULL
        IF @UpdatedPhone IS NULL
        BEGIN
            -- Nếu là NULL, phát sinh lỗi và hủy bỏ thao tác UPDATE
            RAISERROR('Phone không thể để trống', 16, 1)
            ROLLBACK TRANSACTION
        END

        -- Kiểm tra nếu giá trị mới của trường "Phone" chứa ký tự chữ cái
        IF @UpdatedPhone LIKE '%[a-zA-Z]%'
        BEGIN
            -- Nếu chứa chữ cái, phát sinh lỗi và hủy bỏ thao tác UPDATE
            RAISERROR('Phone không được chứa chữ cái', 16, 1)
            ROLLBACK TRANSACTION
        END
    END
END

```

```

-- Kiểm tra trigger
UPDATE Supplier SET Phone = '0-XXX-XXX-XX' WHERE Id = 4

```

Messages

Msg 50000, Level 16, State 1, Procedure Trigger_PhoneUpdate, Line 27 [Batch Start Line 191]
 Phone không được chứa chữ cái
 Msg 3609, Level 16, State 1, Line 193
 The transaction ended in the trigger. The batch has been aborted.

```
--2. Cursor:
-- Viết một function với input vào Country và xuất ra danh sách các Id và Company Name ở thành phố đó theo dạng sau
--INPUT : 'USA'
--OUTPUT : Companies in USA are : New Orleans Cajun Delights(ID:2) ; Grandma Kelly's Homestead(ID:3) ...

CREATE FUNCTION dbo.ufn_ListCompanyByCountry(@CountryName NVARCHAR(MAX))
RETURNS NVARCHAR(MAX)
AS
BEGIN
    -- Khởi tạo biến @CompanyList lưu trữ danh sách công ty
    DECLARE @CompanyList NVARCHAR(MAX) ='Companies in ' + @CountryName + ' are: ';

    -- Khai báo biến lưu trữ Id và tên công ty
    DECLARE @Id INT
    DECLARE @CompanyName NVARCHAR(MAX)

    -- Khai báo con trỏ Cursor để lặp qua các công ty trong đất nước
    DECLARE CompanyCursor CURSOR READ_ONLY
    FOR
    SELECT Id, CompanyName
    FROM Supplier
    WHERE LOWER(Country) LIKE LOWER(@CountryName)

    -- Mở con trỏ Cursor
    OPEN CompanyCursor

    -- Lấy dữ liệu đầu tiên từ Cursor
    FETCH NEXT FROM CompanyCursor INTO @Id, @CompanyName

    -- Lặp qua từng công ty
    WHILE @@FETCH_STATUS = 0
    BEGIN
        -- Cập nhật danh sách công ty
        SET @CompanyList = @CompanyList + @CompanyName + '(ID:' + LTRIM(STR(@Id)) + ') ';

        -- Lấy dữ liệu công ty tiếp theo từ Cursor
        FETCH NEXT FROM CompanyCursor INTO @Id, @CompanyName
    END

    -- Đóng và giải phóng Cursor
    CLOSE CompanyCursor
    DEALLOCATE CompanyCursor

    -- Trả về danh sách công ty
    RETURN @CompanyList
END

SELECT dbo.ufn_ListCompanyByCountry('USA')
```

Results		Messages
	(No column name)	
1	Companies in USA are: New Orleans Cajun Delights(ID:2) ; Grandma Kelly's Homestead(ID:3) ; Bigfoot Breweries(ID:16) ; New England Seafood Cannery(ID:19) ;	

```
--3. Transaction:
-- Viết các dòng lệnh cập nhật Quantity của các sản phẩm trong bảng OrderItem mà có OrderID được đặt từ khách hàng USA.
--Quantity được cập nhật bằng cách input vào một @DFactor sau đó Quantity được tính theo công thức Quantity = Quantity / @DFactor.
--Ngoài ra còn xuất ra cho biết số lượng hóa đơn đã được cập nhật. (Sử dụng TRANSACTION để đảm bảo nếu có lỗi xảy ra thì ROLL BACK lại)

-- Sử dụng TRY-CATCH cho phép bạn xử lý các ngoại lệ một cách linh hoạt và thực hiện các hành động thích hợp dựa trên loại lỗi gặp phải.
BEGIN TRY
    BEGIN TRANSACTION UpdateQuantityTrans

    -- Vô hiệu hóa thông báo số hàng bị tác động
    SET NOCOUNT ON

    -- Khai báo biến lưu trữ số lượng records đã cập nhật
    DECLARE @NumberOfUpdateRecords INT = 0
    DECLARE @DFactor INT = 1;

    -- Cập nhật số lượng của các đơn hàng trong bảng OrderItem
    UPDATE OI
    SET Quantity = Quantity/@DFactor
    FROM OrderItem OI
    INNER JOIN [Order] O ON O.Id = OI.OrderId
```

```

INNER JOIN Customer C ON C.Id = O.CustomerId
WHERE C.Country LIKE '%USA%' -- theo điều kiện rằng khách hàng có đất nước là "USA"

-- Lấy số lượng records đã cập nhật
SET @NumberOfUpdateRecords = @@ROWCOUNT

-- In thông báo về số lượng record đã cập nhật thành công
PRINT 'Cap nhat thanh cong ' + LTRIM(STR(@NumberOfUpdateRecords)) + ' dong (hoa don) trong bang OrderItem'

COMMIT TRANSACTION UpdateQuantityTrans
END TRY
BEGIN CATCH
-- Nếu xảy ra lỗi, rollback giao dịch
ROLLBACK TRAN UpdateQuantityTrans

-- In thông báo lỗi và chi tiết lỗi
PRINT 'Cap nhat that bai. Xem chi tiet: ' + ERROR_MESSAGE()
END CATCH

```



Messages

Cap nhat thanh cong 352 dong (hoa don) trong bang OrderItem

```

--4. Temp Table:
-- Viết TRANSACTION với Input là hai quốc gia.
--Sau đó xuất thông tin là quốc gia nào có số sản phẩm cung cấp (thông qua SupplierId) nhiều hơn.
--Cho biết luôn số lượng số sản phẩm cung cấp của mỗi quốc gia. Sử dụng cả hai dạng bảng tạm (# và @)

BEGIN TRY
    BEGIN TRANSACTION CompareTwoCountryTrans;

    -- Tắt thông báo số hàng bị tác động
    SET NOCOUNT ON;

    -- Khai báo và khởi tạo biến cho hai quốc gia
    DECLARE @Country1 NVARCHAR(MAX) = 'USA';
    DECLARE @Country2 NVARCHAR(MAX) = 'Germany';

    -- Tạo bảng tạm (#SupplyInfo1) dùng bảng vật lý
    CREATE TABLE #SupplyInfo1 (
        ProductName NVARCHAR(MAX),
        Country NVARCHAR(MAX)
    );

    -- Tạo biến bảng (@SupplyInfo2)
    DECLARE @SupplyInfo2 TABLE (
        ProductName NVARCHAR(MAX),
        Country NVARCHAR(MAX)
    );

    -- Thêm thông tin cung cấp cho Quốc gia 1 vào bảng tạm
    INSERT INTO #SupplyInfo1
    SELECT P.ProductName, S.Country
    FROM Product P
    INNER JOIN Supplier S ON S.Id = P.SupplierId
    WHERE S.Country = @Country1;

    -- Thêm thông tin cung cấp cho Quốc gia 2 vào biến bảng
    INSERT INTO @SupplyInfo2
    SELECT P.ProductName, S.Country
    FROM Product P
    INNER JOIN Supplier S ON S.Id = P.SupplierId
    WHERE S.Country = @Country2;

    -- Lấy số lượng sản phẩm cung cấp duy nhất của mỗi quốc gia
    DECLARE @NumSupplyProduct1 INT = (SELECT COUNT(DISTINCT ProductName) FROM #SupplyInfo1);
    DECLARE @NumSupplyProduct2 INT = (SELECT COUNT(DISTINCT ProductName) FROM @SupplyInfo2);

    -- In số lượng sản phẩm cung cấp của mỗi quốc gia
    PRINT 'Quoc gia: ' + @Country1 + ' cung cap ' + CAST(@NumSupplyProduct1 AS NVARCHAR(10)) + ' san pham';
    PRINT 'Quoc gia: ' + @Country2 + ' cung cap ' + CAST(@NumSupplyProduct2 AS NVARCHAR(10)) + ' san pham';

    PRINT '';
    -- So sánh số lượng sản phẩm cung cấp giữa hai quốc gia và in kết quả

```

```

PRINT
CASE
    WHEN @NumSupplyProduct1 = @NumSupplyProduct2 THEN 'So luong san pham cung cap tai quoc gia ' + @Country1 + ' bang voi quoc gia ' + @Country2
    WHEN @NumSupplyProduct1 > @NumSupplyProduct2 THEN 'So luong san pham cung cap tai quoc gia ' + @Country1 + ' nhieu hon quoc gia ' + @Country2
    ELSE 'So luong san pham cung cap tai quoc gia ' + @Country2 + ' nhieu hon quoc gia ' + @Country1
END;

-- Xóa bảng tạm
DROP TABLE #SupplyInfo1;

COMMIT TRANSACTION CompareTwoCountryTrans;
END TRY
BEGIN CATCH
    -- Nếu xảy ra lỗi ở, chuyển ngay lập tức đến khối CATCH. Transaction sẽ được roll back và một thông báo lỗi sẽ được in ra để chỉ ra lỗi
    ROLLBACK TRANSACTION CompareTwoCountryTrans;
    PRINT 'Co loi xay ra. Xem chi tiet';
    PRINT ERROR_MESSAGE();
END CATCH;

```

Messages

Quoc gia: USA cung cap 12 san pham

Quoc gia: Germany cung cap 9 san pham

So luong san pham cung cap tai quoc gia USA nhieu hon quoc gia Germany