**DATABASE MANAGEMENT – LAB 07**

Đinh Anh Huy – 18110103

**Yêu cầu:**

1. **Trigger**:
   * 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\_OrderDelete]

ON [dbo].[Order]

FOR DELETE

AS

    DECLARE @DeletedOrderId INT

    SELECT @DeletedOrderId = Id FROM deleted

    DELETE FROM OrderItem WHERE OrderId = @DeletedOrderId

    PRINT 'Cac mat hang trong hoa don OrderId = ' + LTRIM(STR(@DeletedOrderId)) + ' da duoc xoa khoi bang OrderItem.'

-- + Kiểm tra các mặt hàng có trong hoá đơn OrderId = 89

SELECT \* FROM [OrderItem] WHERE OrderId = 89

Graphical user interface, application, table

Description automatically generated

-- + Xoá bỏ Foreign Key Constraint để kiểm nghiệm trigger.

ALTER TABLE [OrderItem] DROP CONSTRAINT FK\_ORDERITE\_REFERENCE\_ORDER

-- + Thực hiện lệnh xoá bỏ hoá đơn OrderId = 89

DELETE FROM [Order] WHERE Id = 89

Graphical user interface, text, application

Description automatically generated- + Truy vấn để kiểm tra lại các mặt hàng trong hoá đơn OrderId = 89 còn tồn tại trong bảng OrderItem không

SELECT \* FROM [OrderItem] WHERE OrderId = 89

**Graphical user interface, application

Description automatically generated**

* + 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

CREATE TRIGGER [dbo].[Trigger\_OrderID1Delete]

ON [dbo].[Order]

FOR DELETE

AS

    DECLARE @DeletedOrderId INT

    SELECT @DeletedOrderId = Id FROM deleted

    IF (@DeletedOrderId = 1)

    BEGIN

        RAISERROR ('OrderID = 1 khong the xoa duoc', 16, 1);

        ROLLBACK TRANSACTION

    END

-- + Ưu tiên Trigger "Trigger\_OrderID1Delete" thực hiện trước Trigger "Trigger\_OrderDelete"

EXEC sp\_settriggerorder @triggername = 'Trigger\_OrderID1Delete', @order='First', @stmttype='DELETE'

-- + Thử xoá dữ liệu của OrderId = 1

DELETE FROM [Order] WHERE Id = 1

**Text

Description automatically generated**

* + 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\_SupplierUpdate]

ON [dbo].[Supplier]

FOR UPDATE

AS

    DECLARE @UpdatedPhone NVARCHAR(30)

    IF UPDATE(Phone)

    BEGIN

        SELECT @UpdatedPhone = Phone FROM inserted

        IF (@UpdatedPhone = NULL) OR (ISNUMERIC(@UpdatedPhone) = 0)

        BEGIN

            RAISERROR ('Phone phai la chu so.', 16, 1);

            ROLLBACK TRANSACTION

        END

    END

-- + Thử update Phone của Supplier Id = 10

UPDATE Supplier SET Phone = '123abc'

WHERE Id = 10

**Text

Description automatically generated**

1. **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 (@CountryDescr NVARCHAR(MAX))

RETURNS NVARCHAR(MAX)

AS

BEGIN

    DECLARE @CompanyList NVARCHAR(MAX) = 'Companies in ' + @CountryDescr + ' are: ';

    DECLARE @Id INT;

    DECLARE @CompanyName NVARCHAR(MAX);

    DECLARE CompanyCursor CURSOR READ\_ONLY

    FOR

    SELECT Id, CompanyName

    FROM Supplier

    WHERE LOWER(Country) LIKE '%' + LTRIM(RTRIM(LOWER(@CountryDescr))) + '%'

    OPEN CompanyCursor

    FETCH NEXT FROM CompanyCursor INTO @Id, @CompanyName

    WHILE @@FETCH\_STATUS = 0

    BEGIN

        SET @CompanyList = @CompanyList + @CompanyName + '(ID:' + LTRIM(STR(@Id)) + '); '

        FETCH NEXT FROM CompanyCursor INTO @Id, @CompanyName

    END

    CLOSE CompanyCursor

    DEALLOCATE CompanyCursor

    RETURN @CompanyList

END

-- + Gọi Function này với INPUT là USA

SELECT dbo.ufn\_ListCompanyByCountry('USA')

Graphical user interface, text, application

Description automatically generated

1. **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).

BEGIN TRY

    BEGIN TRANSACTION UpdateQuantity

        SET NOCOUNT ON;

        DECLARE @NumOfUpdateRecords INT = 0;

        DECLARE @DFactor INT;

        SET @DFactor = 2;

        UPDATE OI SET Quantity = Quantity / @DFactor

        FROM OrderItem OI

        INNER JOIN (

            SELECT O.Id, O.CustomerId, C.Country

            FROM [Order] O INNER JOIN Customer C ON O.CustomerId = C.Id

        ) N ON N.Id = OI.OrderId

        WHERE N.Country LIKE '%USA%'

        SET @NumOfUpdateRecords = @@ROWCOUNT

        PRINT 'Cap Nhat Thanh Cong ' + LTRIM(STR(@NumOfUpdateRecords)) + ' dong trong bang OrderItem.';

    COMMIT TRANSACTION UpdateQuantity

END TRY

BEGIN CATCH

    ROLLBACK TRAN UpdateQuantity

    PRINT 'Cap Nhat That Bai. Xem Chi Tiet : ';

    PRINT ERROR\_MESSAGE();

END CATCH

Text

Description automatically generated

1. **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

    SET NOCOUNT ON;

    DECLARE @Country1 NVARCHAR(50);

    DECLARE @Country2 NVARCHAR(50);

    SET @Country1 = 'USA';

    SET @Country2 = 'Australia';

    CREATE TABLE #OrderInfo1

    (

        Country NVARCHAR(50),

        SupplierId INT

    )

    DECLARE @OrderInfo2 TABLE

    (

        Country NVARCHAR(50),

        SupplierId INT

    )

    INSERT INTO #OrderInfo1

    SELECT S.Country, P.SupplierId

    FROM Product P INNER JOIN Supplier S ON S.Id = P.SupplierId

    WHERE Country = @Country1

    INSERT INTO @OrderInfo2

    SELECT S.Country, P.SupplierId

    FROM Product P INNER JOIN Supplier S ON S.Id = P.SupplierId

    WHERE Country = @Country2

    DECLARE @NumProduct1 INT

    SET @NumProduct1 = (SELECT COUNT(DISTINCT SupplierId) FROM #OrderInfo1)

    DECLARE @NumProduct2 INT

    SET @NumProduct2 = (SELECT COUNT(DISTINCT SupplierId) FROM @OrderInfo2)

    PRINT 'So luong san pham cung cap cua ' + LTRIM(@Country1) + ' : ' + LTRIM(STR(@NumProduct1))

    PRINT 'So luong san pham cung cap cua ' + LTRIM(@Country2) + ' : ' + LTRIM(STR(@NumProduct2))

    PRINT

    CASE

        WHEN @NumProduct1 = @NumProduct2

            THEN 'So luong san pham cua ' + LTRIM(@Country1) + ' bang voi ' + LTRIM(@Country2)

        WHEN @NumProduct1 > @NumProduct2

            THEN 'So luong san pham cua ' + LTRIM(@Country1) + ' lon hon ' + LTRIM(@Country2)

        ELSE 'So luong san pham cua ' + LTRIM(@Country1) + ' nho hon voi ' + LTRIM(@Country2)

    END

    DROP TABLE #OrderInfo1

COMMIT TRANSACTION CompareTwoCountryTrans

END TRY

BEGIN CATCH

    ROLLBACK TRAN CompareTwoCountryTrans

    PRINT 'Co loi xay ra. Xem chi tiet : ';

    PRINT ERROR\_MESSAGE();

END CATCH

Text

Description automatically generated