Chương 5

Trigger

Nội dung

- Giới thiệu trigger
- Phân loại trigger
- Tạo các loại trigger
  - INSERT trigger
  - UPDATE trigger
  - DELETE trigger

Định nghĩa

- Trigger là một loại Procedure đặc biệt, nó được định nghĩa để tự động thực thi khi có một câu lệnh Insert, Update, Delete được sử dụng.
- Trigger dùng để ràng buộc các qui tắc quản lý một cách tự động mỗi khi dữ liệu bị hiệu chỉnh.
- Dùng để kiểm soát tính toàn vẹn dữ liệu.
- Trigger tự động thực thi, ta không gọi trigger thi hành một cách trực tiếp được.

Lợi ích khi dùng Trigger

- Một trigger có thể nhận biết, ngăn chặn và huỷ bỏ được những thao tác làm thay đổi trái phép dữ liệu trong cơ sở dữ liệu.
- Các thao tác trên dữ liệu (xoá, cập nhật và bổ sung) có thể được trigger phát hiện ra và tự động thực hiện một loạt các thao tác khác trên cơ sở dữ liệu nhằm đảm bảo tính hợp lệ của dữ liệu.
- Thông qua trigger, ta có thể tạo và kiểm tra được những mối quan hệ phức tạp hơn giữa các bảng trong cơ sở dữ liệu mà bản thân các ràng buộc không thể thực hiện được.

Sử dụng Trigger

- Trigger được định nghĩa trên 1 table cụ thể, nhưng không thể tạo trigger trên temporary table hay system table.
- Không gọi trực tiếp hoặc truyền nhận tham số đối với trigger
- Có thể được kích hoạt bởi nhiều hơn 1 event (sự kiện)
- Được sử dụng hầu hết các phát biểu T_SQL để viết trigger: CREATE, ALTER, DROP, GRANT, REVOKE, DENY ,LOAD, RESTORE, RECONFIGURE, TRUNCATE TABLE ,UPDATE STATISTICS, SELECT INTO

Sử dụng Trigger

- Xử lý hành động trên nhiều dòng.
- Đọc dữ liệu từ các Table khác trong CSDL khác.
- Không ngăn ngừa thay đổi cấu trúc, mà quan tâm đến sự thay đổi hay xóa dữ liệu trong các bảng có quan hệ với nhau.

Sử dụng Trigger

- Các Constraint được kiểm tra trước, sau đó mới tới Trigger.
- Không nên dùng quá nhiều trigger trong một table.
- Không thể tạo trigger trên các đối tượng ở Temporary table.
- Không nên thiết kế Trigger trả về tập kết quả để đảm bảo tính chất chuyển tác giữa các user và lập trình.

Tạo Trigger

```
CREATE TRIGGER trigger_name
ON table [WITH ENCRYPTION]
{FOR | AFTER| INSTEAD OF}
{[INSERT][,][UPDATE][,][DELETE] }
[WITH APPEND] [NOT FOR REPLICATION]
AS
sql_statement [ . . . n ]
```

Xem thông tin về trigger : lưu trong table sysobjects và syscomments

sp_helptext *Trigger_Name*

sp_helptrigger *Table_Name*

sp_depends *Table_Name*

Tạo Trigger

- **AFTER triggers**: sau khi thực hiện delete hay insert một dòng vào Table thì Trigger mới tự động thực thi (gọi là reactive)
- **INSTEAD OF triggers**: kiểm tra trước khi Insert/Delete. Không xây dựng được trên table có áp dụng cascade delete/ update
- **NESTED trigger**: Table 1 có trigger1, Table 2 có trigger 2, nếu thao tác trên Table 1 mà có liên quan đến Table 2 thì Trigger 2 sẽ thực thi còn gọi là lồng Trigger

Tạo Trigger

	AFTER trigger	INSTEAD OF trigger
Được áp dụng trên	Tables	Tables và views
Số trigger trên table/view	Nhiều trigger Trên 1 sự kiện (UPDATE, DELETE, INSERT)	1 Trigger trên 1 sự kiện (UPDATE, DELETE, INSERT)
Ràng buộc tham chiếu	Không giới hạn	Không cho phép

Tạo Trigger

	AFTER trigger	INSTEAD OF trigger
Thực thi	- Thực thi các RBTV • Bật cờ sự kiện •Tạo table inserted và deleted. •Thực thi sự kiện bẫy trigger •Thực thi code trong trigger	Tạo table inserted và deleted. •Thực thi code trong trigger •Thực thi các RBTV bỏ qua sự kiện bẫy trigger
Thứ tự thực thi	Cho phép xác định trigger đầu tiên và cuối cùng	
Table inserted và deleted	Không cho phép column có kiểu dữ liệu text, ntext, image	Cho phép column có kiểu dữ liệu text, ntext, image

Tạo Trigger

- VD

```
CREATE TRIGGER ThemxoaCTHD ON
[Order Details] FOR INSERT, UPDATE
AS
        Raiserror ('Có %d dòng đã được hiệu
        chỉnh',0,1,@@rowcount)
RETURN
```

Các loại Trigger

- INSERT trigger
- UPDATE trigger
- DELETE trigger

Insert Trigger

- Trigger sẽ được thực thi khi có mẫu tin chèn vào bảng, SQL server tạo ra bảng mang tên INSERTED để lưu các mẫu tin chèn, trong Trigger ta có thể tham khảo đến mẫu tin này.
- Các bước thực hiện:
  - Step 1

    INSERT statement to a table with an INSERT trigger defined
  - Step 2

    INSERT Statement Logged
  - Step 3

    Trigger Actions Executed

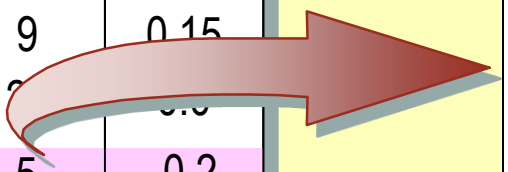
Insert Trigger

TRIGGER Actions Execute

```
Trigger Code:
USE Northwind
CREATE TRIGGER OrdDet_Insert
ON [Order Details]
FOR INSERT
AS
UPDATE P SET
UnitsInStock = (P.UnitsInStock – I.Quantity)
FROM Products AS P INNER JOIN Inserted AS I
ON P.ProductID = I.ProductID
```

Order Details

OrderID	ProductID	UnitPrice	Quantity	Discount
10522	10	31.00	7	0.2
10523	41	9.65	9	0.15
10524	7	30.00	2	0.0
10523	2	19.00	5	0.2



Products

ProductID	UnitsInStock	...	...
1	15		
2	10		
3	65		
4	20		

Insert Trigger

Example:

```
CREATE TRIGGER Trg_NgayLap_NgayGiaoHD
ON Orders AFTER INSERT
AS
DECLARE @NgayLapHD DateTime, @NgayGiao DateTime
SELECT @NgayLapHD=hd.Orderdate, @NgayGiao=hd.RequiredDate
FROM Orders hd INNER JOIN Inserted i ON hd.Orderid=i.orderid
IF @NgayGiao<@NgayLapHD
   BEGIN
         RAISERROR(500103,10,1)
         ROLLBACK TRANSACTION
   END
```

Insert Trigger

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1001	12/23/2003	N	Tp. HCM	12/28/2003	CDCN4
1002	01/01/2004	X	Can Tho	01/05/2004	DHCT
1003	01/01/2004	N	Tp. HCM	03/03/2004	CDCN4

INSERT HoaDon
VALUES (1003,'1/1/2004','N','TP. HCM','03/03/2004','CDCN4')

INSERT Trigger

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1003	01/01/2004	N	Tp. HCM	03/03/2004	CDCN4

Inserted Table

Insert Trigger

Example

```
CREATE TRIGGER ktDongia ON [Order Details]
FOR INSERT AS
IF NOT EXISTS (SELECT * FROM INSERTED i
        INNER  JOIN Orders o
          ON i.OrderId = o.OrderId )
BEGIN
          RAISERROR(60000,16,1,'Orderid','Order
          details','Orderid','Orders')
          ROLLBACK TRANSACTION
END
```

Delete Trigger

- Trigger sẽ được thực thi khi có mẫu tin xóa khỏi bảng, SQL server tạo ra bảng mang tên DELETED để lưu các mẫu tin bị xóa, trong Trigger ta có thể tham khảo đến mẫu tin này.
- Có 3 cách ràng buộc khi sử dụng DELETE trigger.
  - The Cascade method
  - The Restrict method
  - The Nullify method

Delete Trigger

Trigger Actions Execute

Products

ProductID	Discontinued	...	...
1	0		
2	1		
3	0		
4	0		

```
USE Northwind
CREATE TRIGGER Category_Delete
      ON Categories
      FOR DELETE
AS
    UPDATE P SET Discontinued = 1
      FROM Products AS P INNER JOIN deleted AS d
      ON P.CategoryID = d.CategoryID
```

Delete Trigger

Example:

```
CREATE TRIGGER Trg_Xoa_HD
ON Orders AFTER DELETE
AS
SET NOCOUNT ON
IF EXISTS (SELECT * FROM Deleted)
   BEGIN
       DELETE [Order Details] WHERE [Order
details].Orderid
      IN (SELECT hd.Orderid  FROM orders hd
           INNER JOIN Deleted d ON hd.Orderid=d.Orderid)
       RAISERROR('Cac chi tiet HD da bi xoa',10,1)
   END
SET NOCOUNT ON
-------------------------------
DELETE Orders WHERE Orderid=10248
```

Delete Trigger

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1001	12/23/2003	N	Tp. HCM	12/28/2003	CDCN4
1002	01/01/2004	X	Can Tho	01/05/2004	DHCT
1003	01/01/2004	N	Tp. HCM	03/03/2004	CDCN4

1 DELETED HoaDon WHERE MaHD=1003

3 DELETED Trigger

2 Deleted row

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1003	01/01/2004	N	Tp. HCM	03/03/2004	CDCN4

Delete Trigger

VD

```
CREATE TRIGGER NoDelete12
ON Products
FOR DELETE AS
IF (SELECT Productid FROM deleted) = 12
BEGIN
        PRINT 'You cannot delete the Productid =12'
        ROLLBACK TRANSACTION
END
```

Update Trigger

- Mỗi khi có mẫu tin nào đó được cập nhập, giá trị những cột liên quan đến trigger sẽ được kiểm tra trước khi cập nhập. Mẫu tin bị cập nhật sẽ được sao lưu vào bảng insert (chứa giá trị mới) và bảng Delete (chứa giá trị cũ).
- Các bước thực hiện

  Step 1:

  DELETE Statement to a Table with a DELETE Statement Defined

  Step 2

  DELETE Statement Logged

  Step 3

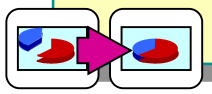
  Trigger Actions Executed

Update Trigger

TRIGGER Actions Execute

```
USE Northwind
GO
CREATE TRIGGER Employee_Update
  ON Employees
  FOR UPDATE
AS
IF UPDATE (EmployeeID)
BEGIN TRANSACTION
  RAISERROR ('Transaction cannot be processed.\
  ***** Employee ID number cannot be modified.', 10, 1)
  ROLLBACK TRANSACTION
```

Transaction cannot be processed.
******* Member number cannot be modified**



Employees

EmployeeID	LastName	FirstName	Title	HireDate
1	Davolio	Nancy	Sales Rep	~~~
2	Fuller	Andrew	Vice Pres.	~~~
3	Leverling	Janet	Sales Rep	~~~
4	Peacock	Margare	Sales Rep	~~~

Update Trigger

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1001	12/23/2003	N	Tp. HCM	12/28/2003	CDCN4
1002	01/01/2004	X	Can Tho	01/05/2004	DHCT
1003	01/01/2004	N	Tp. HCM	03/03/2004	CDCN4

1 UPDATED HoaDon SET MaKH='TH3' WHERE MaHD=1003

3 UPDATED Trigger

2 UPDATE ROW

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1003	01/01/2004	N	Tp. HCM	03/03/2004	TH3

Update Trigger

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1001	12/23/2003	N	Tp. HCM	12/28/2003	CDCN4
1002	01/01/2004	X	Can Tho	01/05/2004	DHCT
1003	01/01/2004	N	Tp. HCM	03/03/2004	th3

3 UPDATED Trigger

1 INSERTED Table

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1003	01/01/2004	N	Tp. HCM	03/03/2004	th3

2 DELETED Table

MaHD	NgayLapHD	LoaiHD	NoiChuyen	NgayGiao	MaKH
1003	01/01/2004	N	Tp. HCM	03/03/2004	CDCN4

Update Trigger

VD

```
CREATE TRIGGER NoUpdateMealcode
ON Products
FOR UPDATE AS
IF UPDATE (Productid)
BEGIN
        PRINT 'You cannot modify Productid codes'
        ROLLBACK TRANSACTION
END
```

Update Trigger

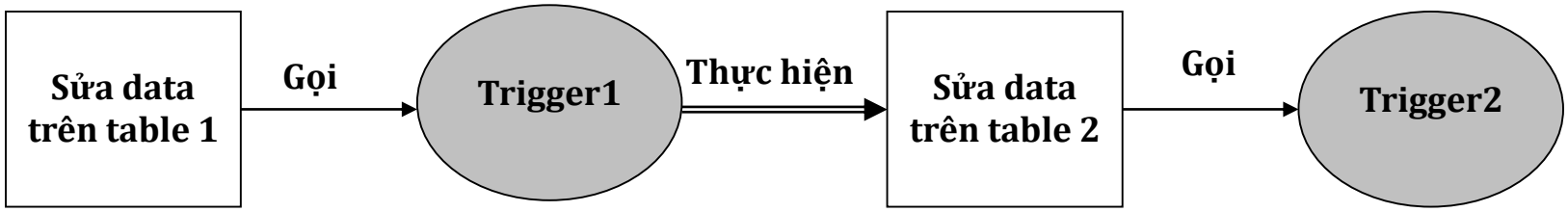
VD

```
CREATE TRIGGER NoUpdateProduct
ON Orders
FOR UPDATE AS
IF (SELECT OrderDate FROM inserted)>= getdate()
BEGIN
        PRINT ‘Ngay lap phai be hon hay bang
              ngay hom nay'
        ROLLBACK TRANSACTION
END
```

Nested Trigger

- Thao tác của một trigger kéo theo việc thi hành một trigger khác, các trigger này được gọi là trigger lồng nhau (nested triggers).
- Có thể lồng tối đa 32 cấp.
- Các trigger được xem như một đơn vị thi hành giao tác (transaction). Do vậy, một trigger trong dãy trigger lồng nhau bị lỗi, thì SQL Server sẽ rollback (quay lui) tất cả các hành động (action) đã được thực hiện bởi các trigger

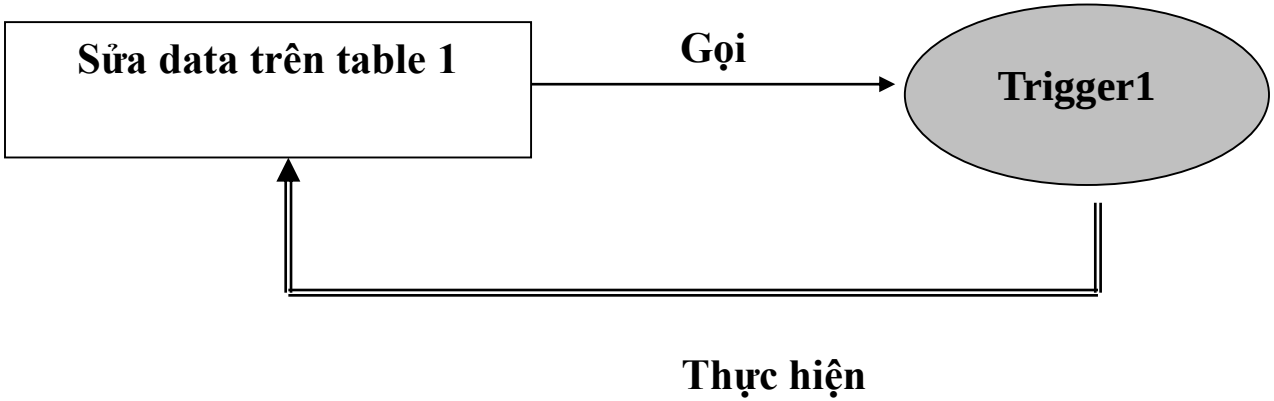
Nested Trigger



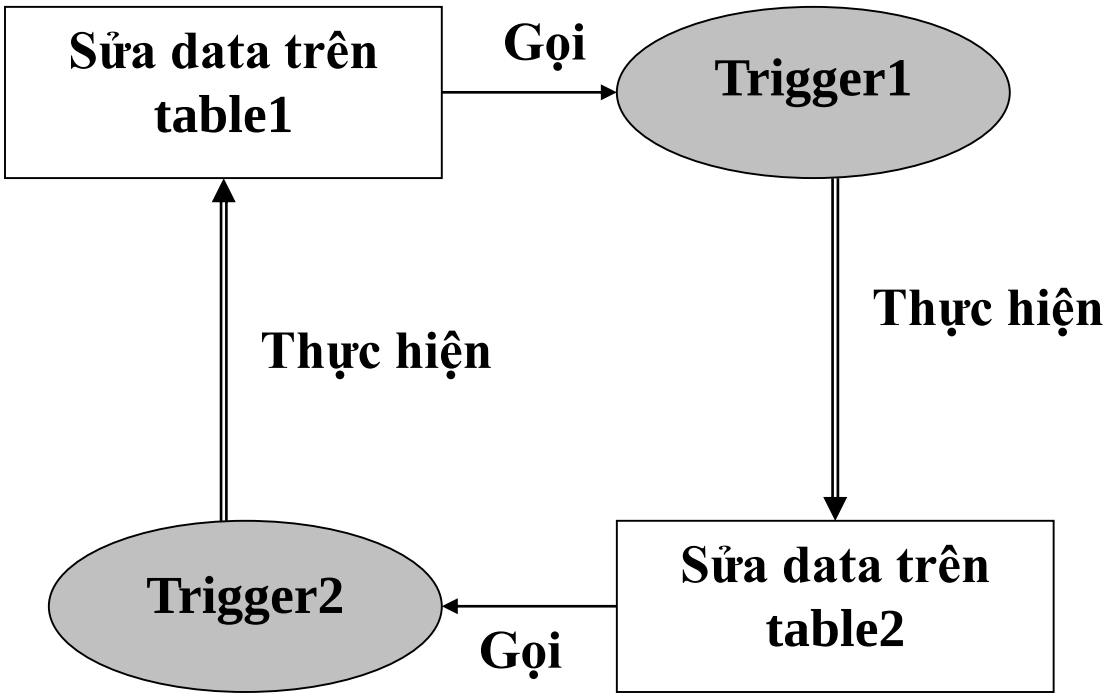
- **Trigger gọi chính nó (recursive trigger) :**

Để tạo trigger dạng này phải bật option của database:

sp_dboption database_name, 'rescusive triggers', True



Nested Trigger



INSTEAD OF Trigger on View

View definition

```
CREATE VIEW service_view
AS
        SELECT o.Productid as ma1, p.Productid as
                ma2, ProductName , orderid
        FROM Products p JOIN [order details] o
                ON  p.productid =o.productid
```

INSTEAD OF Trigger on View

Trigger Definition

```
CREATE TRIGGER del_service
ON service_view
INSTEAD OF DELETE
AS
        DELETE Products WHERE Productid IN
                (SELECT ma1 FROM DELETED)
        DELETE [order details] WHERE productid IN
                (SELECT ma2 FROM DELETED)
```

INSTEAD OF Trigger

- Trigger này chỉ có trong SQL Server 2000, Trigger này sẽ thi hành thay cho các câu lệnh Insert, Delete, Update. Khi tạo trigger kiểu này bạn phải viết lại các lệnh Insert, Delete, Update đối với dữ liệu.
  - Có thể áp dụng cho cả View và Table.
  - Không cho phép áp dụng với các View có lựa chọn With Check Option

INSTEAD OF Trigger

VD: Kiểm tra số lượng sản phẩm tồn kho trước khi tiếp nhận đơn hàng

```
CREATE TRIGGER InsOrdDet ON [Order Details]
INSTEAD OF INSERT
AS
DECLARE @qty int
SELECT @qty=quantity FROM Inserted
IF @qty<= (SELECT UnitsInStock FROM Products P JOIN
    Inserted I ON P.ProductID = I.ProductID)
    INSERT INTO [Order Details]
    SELECT * FROM Inserted
ELSE
    RAISERROR('Not enough products in stock', 16, 1)
```

INSTEAD OF Trigger

- Trường hợp đặc biệt của Views: INSTEAD OF triggers giúp tăng cường khả năng cập nhật

```
CREATE VIEW [Alphabetical list of products]
AS
SELECT Products.*, Categories.CategoryName
FROM Categories INNER JOIN Products
ON Categories.CategoryID = Products.CategoryID
WHERE Products.Discontinued=0
```

INSTEAD OF Trigger

- VD:

```
CREATE TRIGGER InsLP ON [Alphabetical list of
    products]
INSTEAD OF INSERT
AS
IF EXISTS(SELECT * FROM Inserted I JOIN Category C
                ON I.CategoryID=C.CategoryID)
INSERT INTO Products(ProductID, ProductName,
    SupplierID, CategoryID, QuantityPerUnit, UnitPrice,
    UnitsInStock, UnitsOnOrder, ReorderLevel,
    Discontinued)
SELECT ProductID, ProductName, SupplierID,
CategoryID, QuantityPerUnit, UnitPrice, UnitsInStock,
UnitsOnOrder, ReorderLevel, Discontinued
FROM Inserted
```

INSTEAD OF Trigger on View

```
ELSE
BEGIN
    BEGIN TRANSACTION
    INSERT INTO Category(CategoryName)
    SELECT CategoryName FROM Inserted
    INSERT INTO Products(ProductID, ProductName,
      SupplierID, CategoryID, QuantityPerUnit, UnitPrice,
      UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued)
    SELECT ProductID, ProductName, SupplierID,
    @@IDENTITY, QuantityPerUnit, UnitPrice, UnitsInStock,
      UnitsOnOrder, ReorderLevel, Discontinued FROM Inserted
    COMMIT TRANSACTION
END
```

ALTER Trigger

- Trong 1 số trường hợp việc dùng trigger rất nặng nề. VD bạn nhập hàng ngàn dòng trong batch job, trường hợp này tốt nhất bạn hãy vô hiệu (disable) trigger.
- **ALTER TABLE giúp bạn cho phép (enable) hay vô hiệu (disable) trigger theo yêu cầu:**

```
ALTER TABLE table_name
    { ENABLE | DISABLE } TRIGGER
    { ALL | trigger_name [ ,...n ] }
```

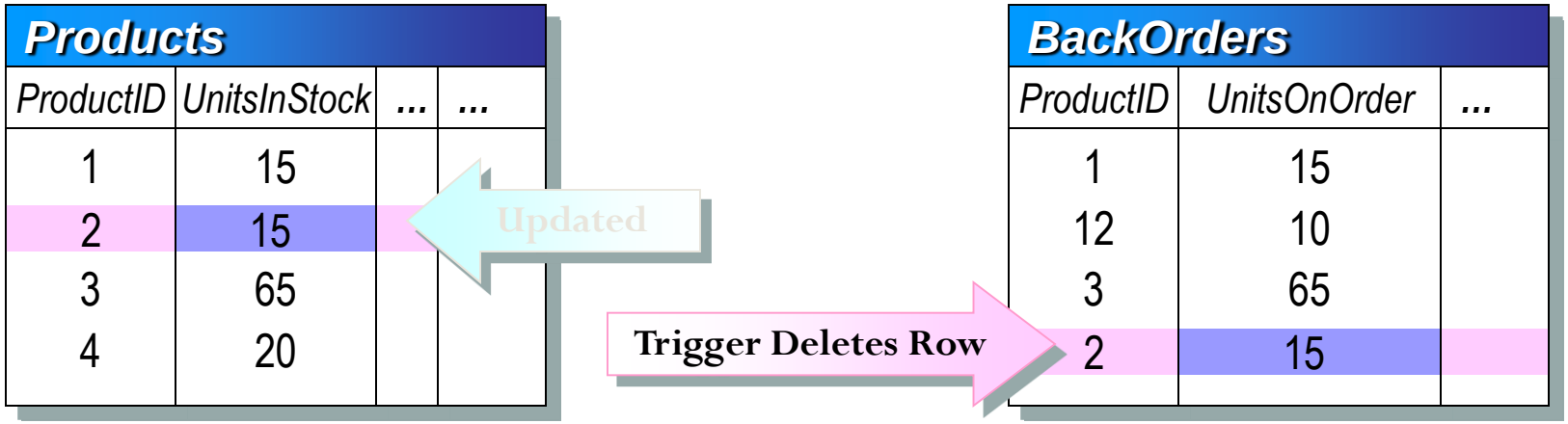
```
ALTER TABLE [Order Details] DISABLE TRIGGER ALL
```

- **Xóa Trigger**

```
DROP TRIGGER Trigger_Name
```

Đảm bảo tính nhất quán của dữ liệu

```
CREATE TRIGGER BackOrderList_Delete
    ON Products FOR UPDATE
AS
IF (SELECT BO.ProductID FROM BackOrders AS BO JOIN
    Inserted AS I ON BO.ProductID = I.Product_ID
    ) > 0
BEGIN
    DELETE BO FROM BackOrders AS BO
    INNER JOIN Inserted AS I
    ON BO.ProductID = I.ProductID
END
```



Đảm bảo tính nhất quán của dữ liệu

Products with Outstanding Orders Cannot Be Deleted

```
IF (Select Count (*)
  FROM [Order Details] INNER JOIN deleted
  ON [Order Details].ProductID = deleted.ProductID
   ) > 0
ROLLBACK TRANSACTION
```

DELETE statement executed on Product table

Trigger code checks the Order Details table

Transaction rolled back

Products

ProductID	UnitsInStock	...	...
1	15		
2	0		
3	65		
4	20		

Order Details

OrderID	ProductID	UnitPrice	Quantity	Discount
10522	10	31.00	7	0.2
10523	2	19.00	9	0.15
10524	41	9.65	24	0.0
10525	7	30.00		

```
'Transaction cannot be processed'
'This product has order history'
```

Cân nhắc khi thực hiện

- Triggers làm việc nhanh do các bảng Inserted và Deleted nằm trong Cache
- Thời gian thực thi được xác định bởi:
  - Số lượng bảng được tham chiếu
  - Số dòng bị ảnh hưởng
- Các Action nằm trong Trigger là 1 phần của giao tác (transaction)

- VD: **cập nhật điểm môn học tự động**

```
CREATE TRIGGER auto_updateGPA ON enroll
FOR UPDATE, DELETE
AS
UPDATE Student
SET GPA = agv(mark)
FROM Student s INNER JOIN enroll e ON s.SID = e.SID
WHERE  e.SID in (SELECT SID FROM deleted)
```

- VD: **ràng buộc 1 sinh viên không được học quá 10 môn**

```
 CREATE TRIGGER overTotalcCourse ON enroll
FOR INSERT
AS
IF EXISTS (SELECT 1 FROM enroll  WHERE SID in
                  (SELECT SID FROM inserted)
                GROUP BY SID
                HAVING COUNT(CID) > 10
                 )
ROLLBACK TRAN
```

Trigger

- **Cập nhật điểm môn học thông qua view**

```
CREATE VIEW V_enroll
AS
SELECT * FROM enroll

CREATE TRIGGER update_mark ON V_enroll
INSTEAD OF UPDATE
AS
    UPDATE enroll
    SET mark = v.mark
    FROM enroll e INNER JOIN inserted I on e.SID=
    i.SID AND e.CID = s.SID
```