

Trường Đại học Khoa học Tự nhiên

Khoa Công nghệ Thông tin

Bộ môn Hệ thống Thông tin

Quản trị Cơ sở Dữ liệu

Chương IV : **RÀNG BUỘC DỮ LIỆU**

GV : TUẤN NGUYỄN HOÀI ĐỨC

E-mail : tnhduc@fit.hcmus.edu.vn

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Đặt vấn đề

- ❖ Ràng buộc dữ liệu (RBDL) là những quy định trên dữ liệu nhằm đảm bảo rằng các ràng buộc toàn vẹn được liệt kê trong quá trình thiết kế CSDL luôn được tuân thủ.
- ❖ Vấn đề : Số lượng RBDL là rất nhiều. Lượng thông tin và người dùng phải cập nhật vào máy tại một thời điểm là rất lớn → Sai sót do mệt mỏi và do không quán xuyến hết mọi RBDL cũng như không nắm bắt được tình trạng CSDL hiện hành → Kiểm tra RBDL bằng con người là không khả thi.

Đặt vấn đề

- ❖ Mục tiêu : Chuyển trách nhiệm kiểm tra RBDL cho SQL Server làm một cách tự động mỗi khi có thay đổi trên DL (Con người không cần kiểm tra thủ công)
- ❖ Cơ chế kiểm tra :
 - **Sẵn có : *PK, FK, Unique***
 - **Ép thỏa đơn giản : *Check, rule,...***
 - **Ép thỏa phức tạp : *Trigger***

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Ràng buộc khoá chính

- ◆ Tạo khóa chính ngay trong lúc tạo bảng bằng column constraint. Thích hợp khi PK chỉ gồm 1 thuộc tính

Create table HoatDongTiepThi

(

MaHoatDong varchar(10) not null **Primary key**,
TenHoatDong nvarchar(50),
LoaiHoatDong varchar(10),
NgayBatDau DateTime,
NgayKetThuc DateTime,
TongChiPhi float

)

Khai báo khóa chính trong Column constraint (Xem chương 2 : Xây dựng CSDL)

Ràng buộc khoá chính

- ❖ Tạo khóa chính ngay trong lúc tạo bảng bằng table constraint. Thích hợp khi PK gồm 2 thuộc tính trở lên hoặc khi muốn chỉ định tên khoá chính.

Create table ChiTietHoatDong

(

MaHoatDong varchar(10),

STT int,

TenChiTiet nvarchar(10),

Constraint PK_CTHD **primary key**(MaHoatDong,STT)

)

Khai báo khóa chính trong Table constraint (Xem chương 2 : Xây dựng CSDL)

Tên khóa chính do người dùng đặt

Ràng buộc khoá chính

❖ Tạo khóa chính cho bảng đã có

- *Đúng nhất là bảng chưa có dữ liệu, nếu bảng đã có dữ liệu, phải xem xét tình trạng CSDL trên tổ hợp thuộc tính muốn chọn làm PK. SQL Server có thể báo lỗi.*
- *Khi chưa có khóa chính thì chỉ cần tạo khóa chính mới.*
- *Thuộc tính khóa chính phải NOT NULL.*
- *Khi đã có khóa chính cũ, cần hủy khóa chính cũ trước khi tạo khóa chính mới.*
 - **Alter table Tab_name drop constraint PK_name**

Ràng buộc khoá chính

- ❖ Tạo khóa chính cho bảng đã có
***Alter table Tab_name add constraint
PK_name primary key(colname [...,n])***
- ❖ Ví dụ
***Alter table DanhMucChiPhi add constraint
Pk_DMCP primary key(MaChiPhi)***

Thuộc tính này bắt buộc phải
là thuộc tính **not null**

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Ràng buộc tồn tại

- ❖ Ràng buộc tồn tại thường được các DBMS cài đặt thành khóa ngoại. SQL Server cũng vậy
- ❖ Cần chú ý tổ hợp thuộc tính hợp thành khóa ngoại phải hoàn toàn giống với tổ hợp thuộc tính làm khóa chính tương ứng về mặt kiểu dữ liệu và thứ tự chỉ định trong danh sách khi khai báo khóa ngoại

Ràng buộc tồn tại

- ◆ Tạo khóa ngoại ngay trong lúc tạo bảng bằng column constraint. Thích hợp khi FK chỉ gồm 1 thuộc tính

Create table ChiTietHoatDong

(

***MaHoatDong varchar(10) foreign key
References HoatDong(MaHoatDong)
On delete cascade On update cascade,
STT int,
TenChiTiet nvarchar(10),
Constraint PK_CTHD primary key(MaHoatDong,STT)***

)

**Khai báo khóa ngoại
trong
Column constraint
(Xem chương 2 : Xây
dựng CSDL)**

Phần optional

Ràng buộc tồn tại

- ◆ Tạo khóa ngoại ngay trong lúc tạo bảng bằng table constraint. Thích hợp khi FK gồm 2 thuộc tính trở lên hoặc khi muốn chỉ định tên khoá ngoại.

Create table ChiTietHoatDong

```
(  
    MaHoatDong varchar(10),  
    STT int,  
    TenChiTiet nvarchar(10),  
    Constraint PK_CTHD primary key(MaHoatDong,STT)  
    Constraint FK_CTHD_HD foreign key (MaHoatDong)  
    References HoatDong(MaHoatDong)  
    On update cascade On delete cascade  
)
```

Nếu nhiều thuộc tính thì thứ tự giống khi tạo bảng

Khai báo khóa ngoại trong Table constraint
(Xem chương 2 : Xây dựng CSDL)

Tên khóa ngoại do người dùng đặt

Ràng buộc tồn tại

- ❖ Tạo khóa ngoại cho bảng đã có
 - **Đúng nhất là bảng chưa có dữ liệu, nếu bảng đã có dữ liệu, phải xem xét tình trạng CSDL trên tổ hợp thuộc tính muốn chọn làm FK, so sánh với tình trạng CSDL trên tổ hợp thuộc tính làm PK tương ứng bên ref_Tab. SQL Server có thể sẽ báo lỗi**
 - **Trường hợp tham chiếu đến tổ hợp thuộc tính không phải khóa ngoại cũng gây báo lỗi.**

Ràng buộc tồn tại

- ❖ Tạo khóa ngoại cho bảng đã có
***Alter table Tab_name add constraint
FK_name foreign key(colname [...,n])
References ref_Tab(ref_colname [...,n])
[On update {cascade / No action}]
[On delete {cascade / No action}]***

Ràng buộc tồn tại

- ◆ Tạo khóa ngoại cho bảng đã có

- ◆ Ví dụ :

*Alter table **ChiTietHoatDong***

*Add constraint **FK_CTHD_HD***

*Foreign key (**MaHoatDong**)*

*References **HoatDong(MaHoatDong)***

On delete cascade

On update cascade

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Ràng buộc duy nhất

- ◆ Tạo RB duy nhất ngay trong lúc tạo bảng bằng column constraint.

Create table HoatDongTiepThi

```
(  
    MaHoatDong varchar(10) not null Primary key,  
    TenHoatDong nvarchar(50) not null unique,  
    LoaiHoatDong varchar(10),  
    NgayBatDau DateTime,  
    NgayKetThuc DateTime,  
    TongChiPhi float  
)
```

Khai báo ràng buộc duy nhất trong Column constraint (Xem chương 2 : Xây dựng CSDL)

Ràng buộc duy nhất

- ◆ Tạo RB duy nhất ngay trong lúc tạo bảng bằng table constraint. Thích hợp khi muốn đặt tên cho RB duy nhất hoặc muốn khai báo 1 lần cho nhiều thuộc tính

Create table HoatDongTiepThi

(

MaHoatDong varchar(10) not null **Primary key**,

TenHoatDong nvarchar(50),

LoaiHoatDong varchar(10),

NgayBatDau DateTime,

NgayKetThuc DateTime,

TongChiPhi float

Constraint u_tenHD unique (TenHoatDong)

)

Khai báo RBDN trong Table constraint (Xem chương 2 : Xây dựng CSDL)

Tên RBDN do người dùng đặt

Danh sách này có thể có nhiều thuộc tính

Ràng buộc duy nhất

- ❖ Tạo ràng buộc duy nhất cho bảng đã có
***Alter table Tab_name add constraint
U_name unique(colname [...,n])***

- ❖ Ví dụ
***Alter table DanhMucChiPhi add constraint
u_TenChiPhi unique(TenChiPhi)***

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Ràng buộc kiểm tra

- ❖ Đây là loại ràng buộc dùng để kiểm tra các RBTV miễn giá trị một cách tự động.
- ❖ Cũng như các constraint khác, ràng buộc kiểm tra gắn liền với 1 table cụ thể và dùng để kiểm tra miễn giá trị của các thuộc tính trong table đó mỗi khi có thay đổi trên dữ liệu
- ❖ Ví dụ : Ta có RBTV sau : Điểm thi có giá trị từ 0 đến 10 và chính xác đến 0.25

Ràng buộc kiểm tra

- ◆ Tạo ràng buộc kiểm tra ngay khi tạo bảng bằng column constraint

Create table KetQua

(

MaSV varchar(10) not null,

MaMH varchar(10) not null,

LanThi int,

Diem float **check**

(Diem >=0 And Diem <=10 And
Diem*4=Round(Diem*4,0)),

Constraint PK_KQ primary key(MaSV, MaMH)

)

Khai báo RBKT
trong column
constraint (Xem
chương 2 : Xây
dựng CSDL)

Ràng buộc kiểm tra

- ◆ Tạo ràng buộc kiểm tra ngay khi tạo bảng bằng table constraint
Create table KetQua

(

MaSV varchar(10) not null,

MaMH varchar(10) not null,

LanThi int,

Diem float,

Constraint PK_KQ primary key(MaSV, MaMH),

Constraint ck_Diem check

(Diem >=0 And Diem <=10 And
Diem*4=Round(Diem*4,0))

)

Khai báo RBKT
trong Table
constraint (Xem
chương 2 : Xây
dựng CSDL)

Ràng buộc kiểm tra

- ❖ Tạo RBKT trên bảng đã có. Nếu bảng này đã được nhập liệu, phải đảm bảo dữ liệu hiện hành không vi phạm RBKT sắp tạo.
- ❖ `Alter Tab_name add constraint check_name check(Logical expression)`
- ❖ Ví dụ :

*Alter table **Ketqua** add constraint **ck_diem**
Check(**Diem** >=0 And **Diem** <=10 And
Diem*4 = Round(**Diem***4,0))*

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Ràng buộc mặc nhiên

- ❖ Có những thuộc tính của bảng dữ liệu mà ở hầu hết các dòng, thuộc tính ấy mang cùng một giá trị
- ❖ Có những thuộc tính mà ngay khi nhập liệu nó phải mang một giá trị khởi đầu nào đó để làm cơ sở cho các tình toán tích lũy về sau
- ❖ Từ đó xuất hiện nhu cầu về thuộc tính có giá trị mặc định. Điều này gọi là ràng buộc mặc nhiên. Trong SQL Server, RBMN được cài đặt thành đối tượng **default**

Ràng buộc mặc nhiên

- ❖ Một hình thức của default là gắn vào một table nhất định nào đó. Nó thực sự là một constraint như các constraint đã tìm hiểu
- ❖ Ví dụ : Tạo default khi tạo table

Create table SanPham

(

MaSP varchar(10) primary key,
TenSP nvarchar(50) not null unique,
LuongTon float **default 0**

)

Khai báo
default trong
column
constraint
(Xem
chương 2 :
Xây dựng
CSDL)

Ràng buộc mặc nhiên

- ❖ Tạo default cho bảng đã có sẵn

*Alter table **Tab_name** add constraint
Def_name default **def_value** for **column***

Ví dụ :

**Alter table SinhVien add constraint
Def_HCM default N'Tp.Hồ Chí Minh'
For TinhThanh**

Ràng buộc mặc nhiên

- ◆ Một hình thức thứ hai : Khác với các constraint đã tìm hiểu, đối tượng mặc nhiên không phải một constraint
 - **Một constraint luôn định nghĩa trên 1 và chỉ 1 table nhất định**
 - **Default không thuộc về một table nào cả, nó là một đối tượng độc lập**
- ◆ Khi muốn default phát huy hiệu lực trên 1 thuộc tính của một bảng, ta phải gắn default vào thuộc tính ấy

Ràng buộc mặc nhiên

- ❖ Một default có thể cùng lúc gắn vào nhiều thuộc tính của nhiều bảng khác nhau
- ❖ Khi không muốn default phát huy hiệu lực trên thuộc tính (đã gắn default), ta phải gỡ default ra khỏi thuộc tính đó
- ❖ Một default chỉ có thể bị xóa đi khi nó không còn gắn vào bất kỳ thuộc tính nào

Ràng buộc mặc nhiên

◆ Khai báo default

Create default **def_zero** as 0

Create default **def_HCM** as **N'Tp.Hồ Chí Minh'**

Tên đối tượng default

Giá trị đối tượng default

◆ Gắn default vào thuộc tính

Sp_bindefault **'def_zero'**, **'SanPham.LuongTon'**

Sp_bindefault **'def_HCM'**, **'SinhVien.TinhThanh'**

Tên đối tượng default
Tham số truyền dạng chuỗi

Thuộc tính của bảng
Tham số truyền dạng chuỗi

Ràng buộc mặc nhiên

- ◆ Gỡ default ra khỏi thuộc tính
 - *Sp_unbindefault* '**SanPham.LuongTon**'
 - *Sp_unbindefault* '**SinhVien.TinhThanh**'
- ◆ Xoá default
 - *Drop default* **def_zero**
 - *Drop default* **def_HCM**

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Đối tượng luật

- ❖ Ta đã biết check constraint là 1 constraint thuộc về 1 và chỉ 1 table, dùng để kiểm tra RBTV miễn giá trị cho thuộc tính của table đó
- ❖ Đối tượng luật (rule) là 1 công cụ tương tự check constraint, nhưng rule không thuộc về 1 table nào cả. Cách phát huy hiệu lực của rule trên thuộc tính của table hoàn toàn tương tự đối tượng default
- ❖ Ví dụ :

Đối tượng luật

Create rule ***rl_0_10_25*** as

(@GiaTri <=10 and @GiaTri >=0 And
@GiaTri*4=Round(@GiaTri*4,0))

Go

Sp_bindrule '***rl_0_10_25***','KetQua.Diem'

Sp_unbindrule '***KetQua.Diem***'

Drop rule ***rl_0_10_25***

Dùng biến tạm thay cho thuộc tính cần kiểm tra, tên biến tạm luôn phải bắt đầu với 1 ký tự @

Đối tượng luật

◆ So sánh với check

• **Check hơn Rule :**

- Trên 1 thuộc tính có thể có đồng thời nhiều check phát huy hiệu lực nhưng trên 1 thuộc tính tại 1 thời điểm chỉ chấp nhận 1 rule.
- Check có thể kiểm tra trên nhiều thuộc tính.

• **Rule hơn Check :**

- Rule có thể phát huy hiệu lực trên nhiều table khác nhau (tính tái sử dụng) trong khi check chỉ phát huy hiệu lực trên 1 table thôi.

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Một số quy tắc chung

- ❖ Các quy tắc này đúng cho mọi constraint có gắn với table cụ thể, bất kể là loại constraint gì
- ❖ Khi tạo constraint
 - ***Từ khóa with check : Có kiểm tra tình trạng dữ liệu hiện có khi tạo constraint***
 - ***Từ khóa with nocheck : Không kiểm tra tình trạng dữ liệu hiện có khi tạo constraint***
- ❖ Ví dụ :

Một số quy tắc chung

- ❖ Khi tạo constraint về điểm của sinh viên, không bận tâm các điểm hiện có, chỉ quan tâm các điểm sắp được ghi vào DB mà thôi

*Alter table **Ketqua** With Nocheck*

*add constraint **ck_diem***

*Check(**Diem** >=0 And **Diem** <=10 And
Diem*4 = Round(**Diem***4,0))*

Một số quy tắc chung

- ❖ Khi tạo constraint về điểm của sinh viên, kiểm tra cả các điểm hiện có, các điểm này phải thỏa check sắp tạo

*Alter table **Ketqua** With check
add constraint **ck_diem***

*Check(**Diem** >=0 And **Diem** <=10 And
Diem*4 = Round(**Diem***4,0))*

Một số quy tắc chung

- ❖ Vô hiệu hóa constraint đang có hiệu lực

*Alter table **Tab_name** Nocheck constraint
{**constraint_name** [...n]}*

- ✱ Ví dụ :

*alter table **SINHVIEN**
nocheck constraint
u_CMND, chk_Nam*

***Lưu ý : Primary, Unique, Default** luôn ở trạng
thái hoạt động*

Một số quy tắc chung

- ❖ Hiệu lực hóa constraint đã mất hiệu lực

*Alter table **Tab_name** Check constraint
{**constraint_name** [...n]}*

- ✱ Ví dụ :

*alter table **SINHVIEN**
Check constraint
u_CMND, chk_Nam*

Nội dung trình bày

- ◆ Đặt vấn đề
- ◆ Ràng buộc khóa chính
- ◆ Ràng buộc tồn tại
- ◆ Ràng buộc duy nhất
- ◆ Ràng buộc kiểm tra
- ◆ Ràng buộc mặc nhiên
- ◆ Đối tượng luật
- ◆ Quy tắc chung
- ◆ Trigger

Trigger

- ◆ Trigger dùng để kiểm tra ràng buộc toàn vẹn phức tạp (không phải ràng buộc miền giá trị)
- ◆ Trigger được tự động kích hoạt khi có thay đổi trên dữ liệu
- ◆ Trigger thực hiện kiểm tra nhờ vào 2 bảng ảo : Inserted và Deleted
- ◆ Một trigger xác định bởi : Một table X, một hay một vài actions trong số 3 actions : I, U, D
- ◆ Một RBTV có thể cần nhiều hơn 1 trigger

Trigger

- ❖ Inserted : Khi có thay đổi trên DL,có thể sẽ có 1 số dòng được thêm vô, inserted chỉ chứa các dòng này. Cấu trúc giống X.
- ❖ Deleted : Khi có thay đổi trên DL,có thể sẽ có 1 số dòng bị xóa đi, inserted chỉ chứa các dòng này. Cấu trúc giống X.

Trigger

❖ Trigger có thể phân thành 3 dạng dựa trên cách giải quyết vấn đề :

- **Dạng 1 :**

- Kiểm tra RBTV.
- Nếu vi phạm báo lỗi và ngăn chặn.

- **Dạng 2 :**

- Kiểm tra RBTV.
- Nếu vi phạm tự động sửa cho thỏa.

- **Dạng 3 :**

- Đồng bộ hóa DL để tránh vi phạm.

Trigger dạng 1

```

CREATE TRIGGER trigger_name
ON { table }
FOR | AFTER { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS
[ BEGIN ]
  { DETERMINING CONFLICT }
  IF (CONFLICT)
  BEGIN
    RAISERROR ('Message', , )
    Rollback transaction
  END
[ END ]

```

Tên table X

Các hành động
cần kiểm tra

Khi vi phạm

Xử lý, tính toán
nhằm xác định thay
đổi vừa làm có gây
vi phạm RBTV
đang xét hay không

Báo lỗi và hủy hết các
thay đổi vừa làm

Trigger dạng 1

❖ Ví dụ :

- ***Cho CSDL có 2 table***

- DonHang (MaDH,...,NgàyDatHang)
- PhieuGH (MaPG, *MaDH*,...,NgàyGiaoHang)

- ***Cho RBTV : Ngày giao hàng phải sau ngày đặt hàng và không trễ quá 1 tháng kể từ ngày đặt hàng.***

Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	+ (NgàyDatHang)	—
PhieuGH	+	+ (MaDH, NgàyGiaoHang)	—

Trigger dạng 1

◆ Quy tắc :

- *Mỗi dấu cộng trong bảng tầm ảnh hưởng viết thành một trigger*
- *Tuy nhiên các dấu cộng trên cùng một dòng (cùng một table) thì có thể xem xét để gom lại viết chung trong 1 trigger được hay không.*

Trigger dạng 1

Create trigger trg_DH_PGH **On** DonHang

For update

As [Begin]

*If exists(select * from Insreted I, PhieuGH P*

Where P.MaDH=I.MaDH And

(P.NgayGiaoHang < I.NgayDatHang Or

Datediff(month, I.NgayDatHang, P.NgayGiaoHang) > 1))

Begin

Raiserror('Ngày dat hang khong hop le',0,1)

Rollback transaction

End

[End]

Vì phạm RBTV nên phải hủy
bỏ mọi thay đổi, trở về trạng
thái ban đầu

Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	+ (NgàyDatHang)	—
PhieuGH	+	+ (MaDH, NgàyGiaoHang)	—

Viết gộp

Viết riêng

Trigger dạng 1

Create trigger trg_DH_PGH **On** PhieuGH

For insert,update

As [Begin]

*If exists(select * from Insreted I, DonHang D*

Where D.MaDH=I.MaDH And

(I.NgayGiaoHang < D.NgayDatHang Or

Datediff(month, D.NgayDatHang, I.NgayGiaoHang) > 1))

Begin

Raiserror('Ngày dat hang khong hop le',0,1)

Rollback transaction

End

[End]

Vi phạm RBTV nên phải hủy
bỏ mọi thay đổi, trở về trạng
thái ban đầu

Trigger dạng 1

❖ Ví dụ : Cho CSDL như sau

- *DonHang*(*MaDH*,*SoMH*)

- *CTDonHang*(*MaDH*,*MaMH*,*SoLuong*)

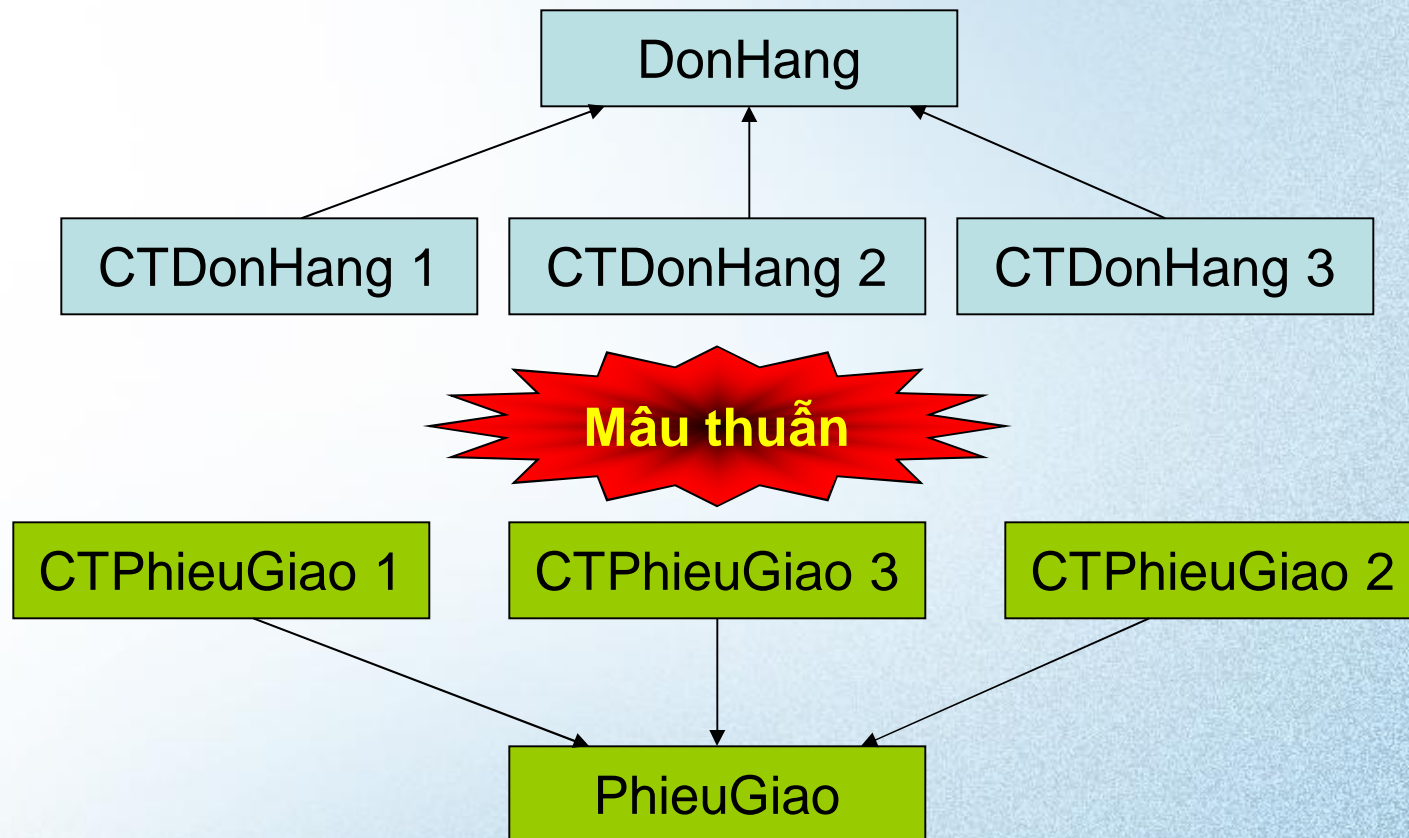
- *PhieuGiao*(*MaPG*,*MaDH*,*NgayGiao*)

- *CTPhieuGiao*(*MaPG*,*MaMH*,*SoLuong*)

Trigger dạng 1

- ❖ Cài đặt ràng buộc toàn vẹn sau : Chỉ giao những mặt hàng có đặt.
- ❖ Phân tích ràng buộc : Chỉ giao những mặt hàng có đặt → mâu thuẫn là khi : *Tồn tại một mặt hàng nào đó được giao cho đơn hàng D nhưng lại không được đặt trong D. Đây là mâu thuẫn giữa các chi tiết đặt hàng và các chi tiết giao hàng tương ứng.*

Trigger dạng 1



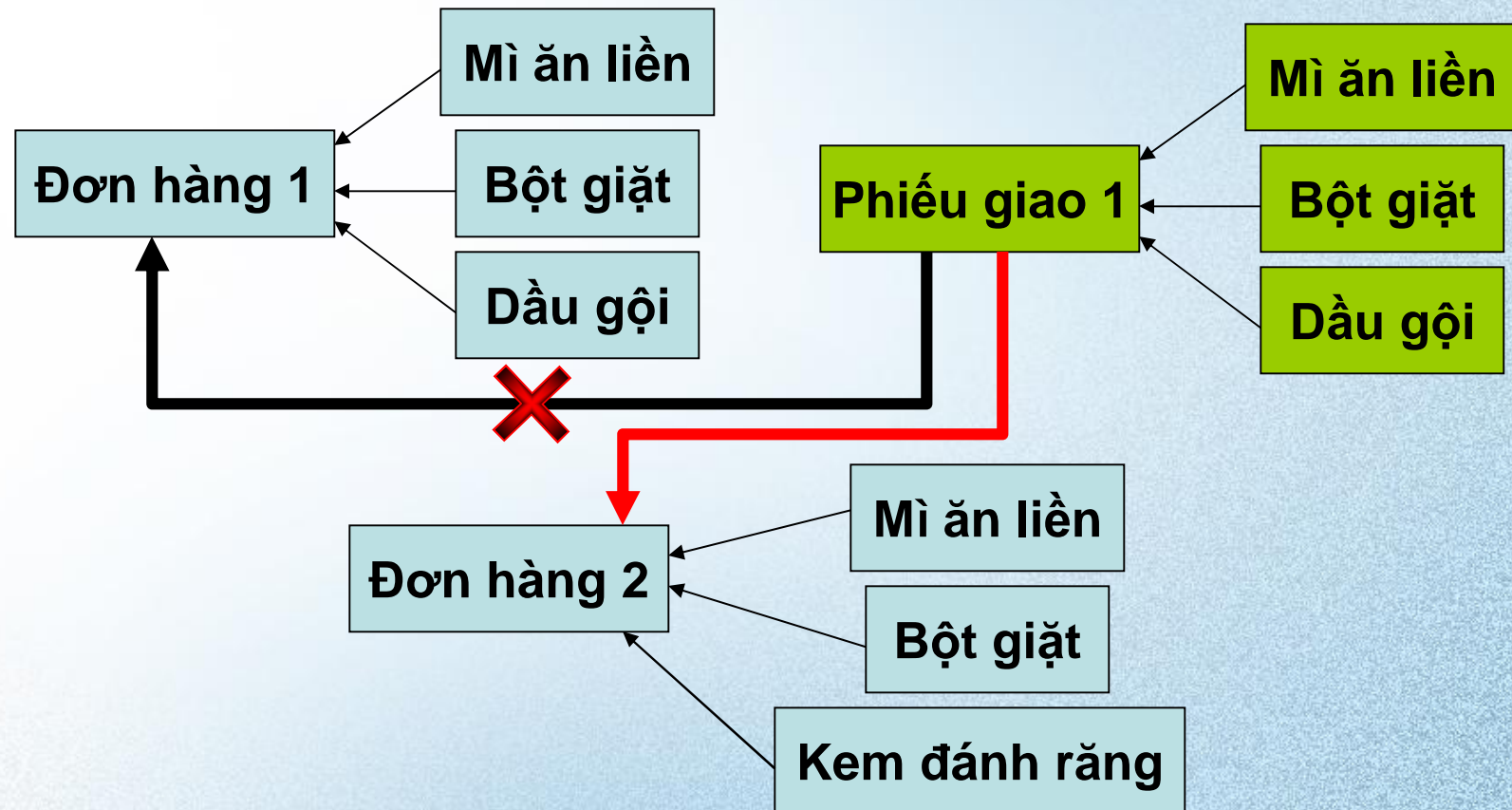
Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	—	—
PhieuGiao	—		
CTPhieuGiao			
CTDonHang			

Trigger dạng 1

♦ Ví dụ tình huống :



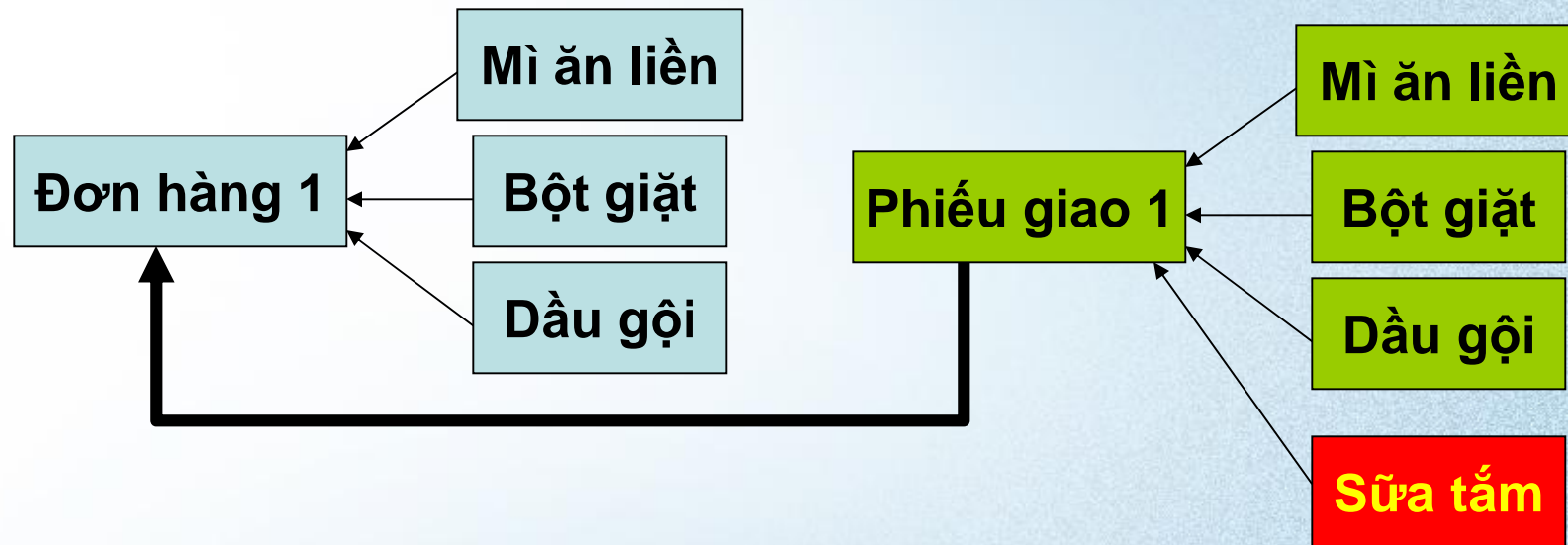
Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	—	—
PhieuGiao	—	+	—
CTPhieuGiao			
CTDonHang			

Trigger dạng 1

♦ Ví dụ tình huống :



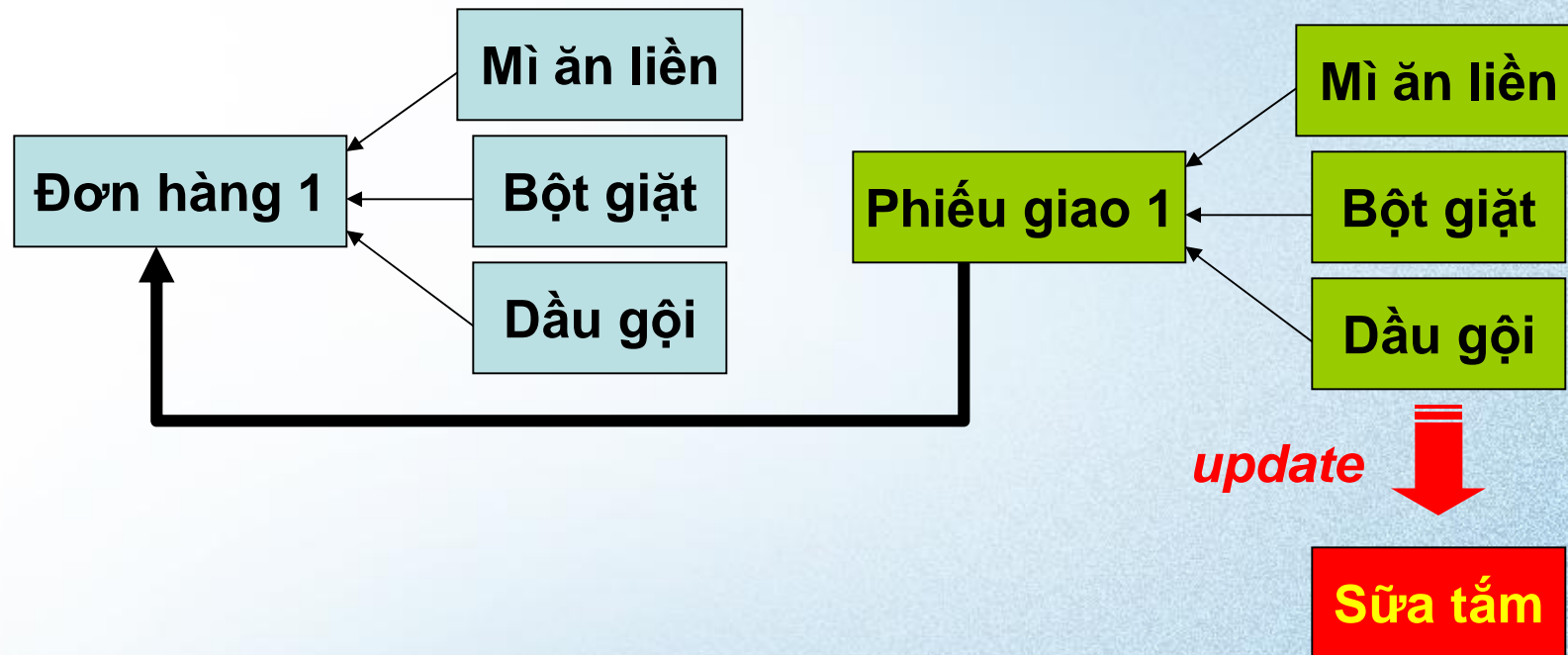
Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	—	—
PhieuGiao	—	+	—
CTPhieuGiao	+		
CTDonHang			

Trigger dạng 1

◆ Ví dụ tình huống :



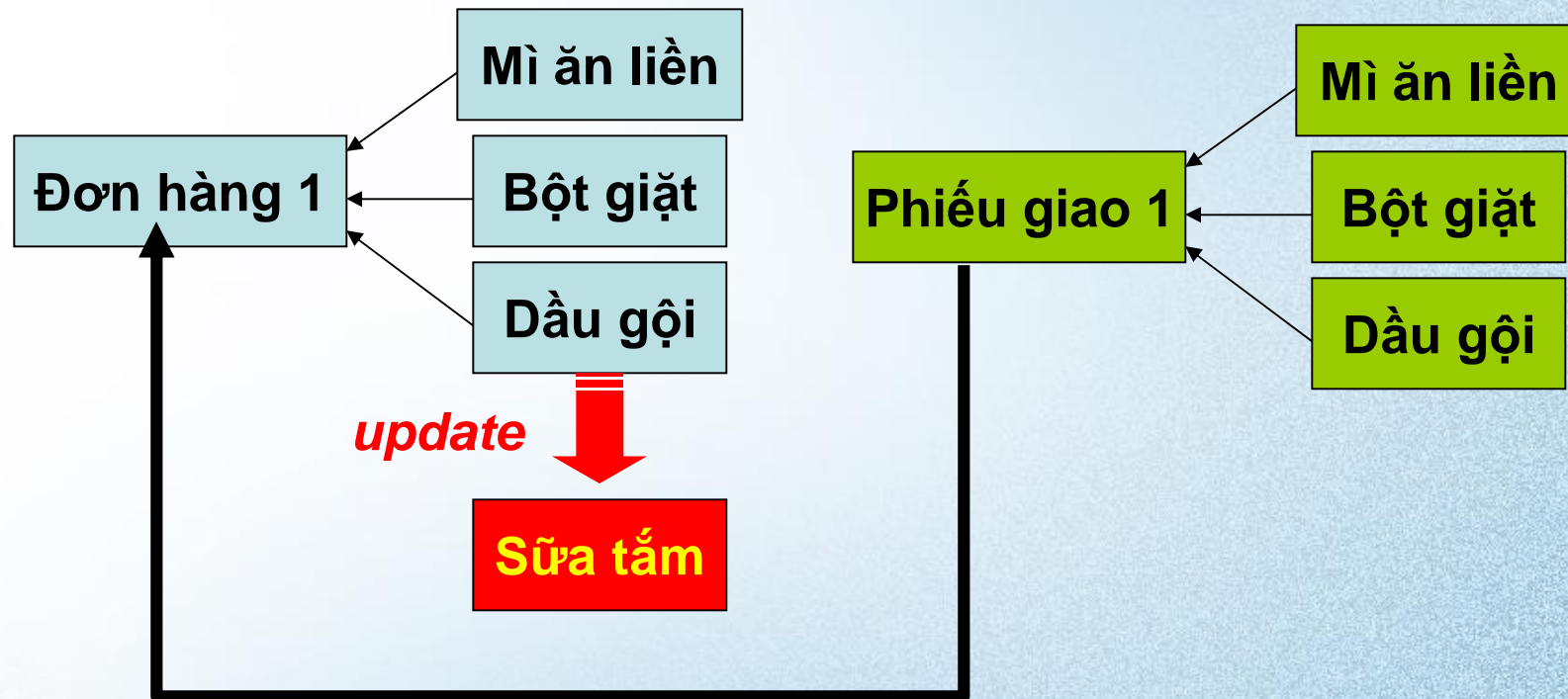
Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	—	—
PhieuGiao	—	+	—
CTPhieuGiao	+	+	—
CTDonHang	—		

Trigger dạng 1

♦ Ví dụ tình huống :



Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	—	—
PhieuGiao	—	+	—
CTPhieuGiao	+	+	—
CTDonHang	—	+	

Trigger dạng 1

♦ Ví dụ tình huống :



Trigger dạng 1

◆ Bảng tầm ảnh hưởng

	insert	update	delete
DonHang	—	—	—
PhieuGiao	—	<i>INSERTED</i> + ?	—
CTPhieuGiao	+	<i>INSERTED</i> +	—
CTDonHang	—	+	<i>DELETED</i> +

Trigger dạng 1

❖ Viết trigger trên bảng PhieuGiao

- **Create trigger trg_PhieuGiao**
- **On PhieuGiao for Update**
- **As**
- **If exists (vi phạm)**
- **Begin**
 - Raiserror...
 - Rollbacktransaction
- **End**

Select *
from Inserted I, CTPhieuGiao CTPG
Where CTPG.MaPG = I.MaPG And
not exists
(select * from CTDonHang CTDH
where CTDH.MaDH = I.MaDH
And CTDH.MaMH = CTPG.MaMH)

Trigger dạng 1

◆ Viết trigger trên bảng CTDonHang

- *Create trigger trg_PhieuGiao*
- *On CTDonHang for Update,Delete*
- *As*
- *If update(MaDH) Or update(MaMH)*
- *Or not exists(select * from inserted)*
- *If exists (vi phạm)*
- *Begin*
 - Raiserror...
 - Rollbacktransaction
- *End*

Select *
from Deleted D
Where exists
(select * from PhieuGiao PG,
CTPhieuGiao CTPG
where PG.MaDH = D.MaDH
And CTPG.MaPG = PG.MaPG
And D.MaMH = CTPG.MaMH)

Trigger dạng 1

- ❖ Viết trigger trên bảng CTPhieuGiao
 - *Sinh viên tự viết, xem như bài tập*

Trigger dạng 2

❖ Ví dụ :

- **Cho bảng KetQua(MaSV, MaMH, Diem)**
- **RBTV : Điểm tính chính xác đến 0.25, nếu vi phạm tự động làm tròn thành 0.0/0.25/0.5/0.75**
- **Giải pháp :**

Create trigger trg_lamtron On KetQua

For insert,update

As

Update KetQua set Diem = Round(Diem*4,0)/4

Where Diem*4 <> Round(Diem*4,0)

GO

Trigger dạng 2

◆ Ví dụ :

• **Cách khác :**

Create trigger trg_lamtron On KetQua

For insert,update

As

Begin

Declare cur_i cursor for

select* from inserted where Diem*4 <> round(Diem*4,0)

Declare @MaSV varchar(10), @MaMH varchar(10), @Diem float

Fetch next from cur_i into @MaSV,@MaMH,@Diem

While (@@fetch_status=0)

Begin

*Update KetQua set Diem = round(Diem*4,0)/4*

Where MaSV = @MaSV and MaMH = @MaMH

Fetch next from cur_i into @MaSV,@MaMH,@Diem

End

Close cur_i

Deallocate cur_i

End

Trigger dạng 3

❖ Ví dụ :

- ***Cho các bảng :***

- Lop(MaLop, TenLop, Siso, Khoa, KhoaHoc,...)
- SinhVien(MaSV, MaLop, HoTen,...)

- ***RBTV : Sĩ số của một lớp phải luôn đúng bằng số sinh viên thực sự thuộc về lớp ấy.***

- ***Giải pháp : Viết trigger sau đây...***

Trigger

❖ Sửa trigger

Alter trigger trg_DH_PGH On DonHang

For update, Insert

As

Begin

If exists(select * from Insreted I, PhieuGH P
Where P.MaDH=I.MaDH And
(P.NgayGiaoHang < I.NgayDatHang Or
Datediff(MM, I.NgayDatHang, P.NgayGiaoHang) > 1))
Begin

Raiserror('Thao tac da bi huy',0,1)

Rollback transaction

End

End

❖ Xóa trigger

Drop trigger tên_trigger_cần_xóa [,...n]

Ví dụ:

Drop trigger trg_DH_PGH

Trigger

❖ Lưu ý :

- *Bảng ảo Inserted và deleted : trên tinh thần chứa nhiều dòng dữ liệu.*
- *Trigger của SQL Server là After trigger*
- *Ví dụ :*
 - Lop(MaLop,...,SiSoMax)
 - SinhVien(MaSV,...,*MaLop*)
 - Ràng buộc : Số sinh viên thực tế không vượt quá số sinh viên max của lớp.

Trigger

- ◆ Giả sử viết trigger cho hành động insert sinh viên X.

Điều kiện vi phạm

...

```
If exists(select * from Inserted I,Lop L  
Where I.MaLop = L.MaLop and  
(select count(*) from SinhVien SV  
where SV.MaLop = L.MaLop)  
>= > L.SiSoMax
```

...

Sai

Đúng

Trigger

❖ Lưu ý :

- *Các trigger trên đây thuộc loại DML trigger (**D**ata **M**anipulation **L**anguage Trigger) : Loại trigger dùng để kiểm soát các thao tác cập nhật dữ liệu (thêm, xóa, sửa...).*
- *Ngoài ra còn có những trigger thuộc loại DDL trigger (**D**ata **D**efinition **L**anguage Trigger) : Loại trigger kiểm soát các hành động định nghĩa dữ liệu (tạo bảng, xóa bảng...)*

Trigger

◆ DDL Trigger :

- Một số sự kiện được kiểm soát : *create_databse, create_table, drop_table, alter_table, alter_view, create_view, drop_view, ... (tham khảo thêm trong BOL)*

- **Vd :**

Create trigger trg_XoaBang

On DataBase

For drop_table

As

Print N' Ai cho phép bạn xóa bảng >.<'

Rollback tran

Go

**Tác dụng lên
DB hiện hành**

Chỉ có với
phiên
Bản SQL
Server 2005
Trở lên

**Để viết chi tiết DDL
trigger, cần tìm hiểu thêm
hàm EventData()**

◆ Instead of:

- *Trigger được gọi thực hiện thay cho thao tác delete/ insert/ update tương ứng*
- *Các dòng mới được thêm chỉ chứa trong bảng inserted*
- *Các dòng bị chỉ định xoá nằm đồng thời trong bảng deleted và bảng dữ liệu (dữ liệu không bị xoá).*
- *Trigger Instead of thường được dùng để xử lý cập nhật trên khung nhìn (view).*

◆ Trigger đệ quy (recursive trigger)

- **Xảy ra khi 1 trigger bị kích hoạt và đến lượt nó lại kích hoạt một trigger khác**
- **Hệ quản trị hỗ trợ tối đa 32 mức.**
- **Có 2 loại:**

- Trục tiếp : 1 trigger kích hoạt chính nó (có thể thông qua 1 trigger khác loại).
- Gián tiếp : 1 trigger kích hoạt chính nó mà phải thông qua trigger khác cùng loại.

- **Chống đệ quy**

- Cách 1 :

```
Sp_configure 'nested_triggers', 0  
GO  
Reconfigure  
GO
```

- Cách 2 :

```
ALTER DATABASE [dbname]  
SET recursive_triggers ON | OFF
```

Dùng 1 cách chỉ chống được đệ quy trực tiếp, muốn chống đệ quy gián tiếp phải dùng đồng thời 2 cách
(<http://msdn.microsoft.com/en-us/library/ms190739.aspx>)

Hết chương IV

