

CTT102 – Cơ sở dữ liệu

Tháng 1/2013

Ràng buộc toàn vẹn và TRIGGER

Bộ môn HTTT - Khoa CNTT - ĐH

Tóm tắt nội dung bài thực hành:

Cài đặt các ràng buộc toàn vẹn sử dụng kỹ thuật
TRIGGER

Bộ môn **Hệ thống thông tin**

Khoa Công nghệ thông tin

ĐH Khoa học tự nhiên TP HCM



MỤC LỤC

1	Mục tiêu và tóm tắt nội dung.....	1
2	Hướng dẫn chi tiết.....	1
2.1	Giới thiệu	1
2.2	Các kỹ thuật cài đặt ràng buộc toàn vẹn đơn giản	2
2.2.1	Các kỹ thuật cơ bản.....	2
2.2.2	RULE.....	2
2.3	Kỹ thuật cài đặt RBTV nâng cao: TRIGGER	4
2.3.1	Giới thiệu.....	4
2.3.2	Cú pháp:.....	4
2.3.3	Một số ví dụ:.....	6
3	Bài tập tại lớp	9
4	Bài tập về nhà	10

CÀI ĐẶT RÀNG BUỘC TOÀN VỆN

1 Mục tiêu và tóm tắt nội dung

Sau khi hoàn thành bài thực hành này sinh viên sẽ biết được:

- Cài đặt các ràng buộc toàn vẹn trên CSDL sử dụng các kỹ thuật CHECK, RULE, TRIGGER, ...

2 Hướng dẫn chi tiết

2.1 Giới thiệu

Ràng buộc toàn vẹn là những quy tắc, quy định trên CSDL nhằm đảm bảo cho CSDL được nhất quán và đúng đắn với ngữ nghĩa của thực tế hay mong muốn của con người.

Các thành phần cơ bản của một ràng buộc toàn vẹn:

- Phát biểu RBTV bằng ngôn ngữ tự nhiên: là một phát biểu tự nhiên về ràng buộc toàn vẹn
- Bối cảnh: là những quan hệ liên quan đến RBTV. Những quan hệ “liên quan” là khi thực hiện những thao tác cập nhật dữ liệu lên những quan hệ này thì RBTV bị vi phạm.
- Nội dung: là phát biểu lại của RBTV bằng ngôn ngữ hình thức để thể hiện được sự chặt chẽ. Một số ngôn ngữ hình thức được sử dụng như: phép tính quan hệ, đại số quan hệ hoặc mã giả.
- Bảng tầm ảnh hưởng: Là bảng mô tả các sự ảnh hưởng đến RBTV của các thao tác cập nhật lên các bảng dữ liệu.

Các loại RBTV:

- RBTV miền giá trị
- RBTV duy nhất
- RBTV tham chiếu
- RBTV liên thuộc tính trên một quan hệ

- RBTV liên bộ trên một quan hệ
- RBTV liên bộ liên quan hệ
- RBTV thuộc tính tổng hợp

2.2 Các kỹ thuật cài đặt ràng buộc toàn vẹn đơn giản

2.2.1 Các kỹ thuật cơ bản

- NOT NULL
- PRIMARY KEY
- FOREIGN KEY
- CHECK

Các trường hợp sử dụng:

- *PRIMARY KEY*: Sử dụng dành riêng cho RBTV khoá chính. Mỗi bảng trong mô hình dữ liệu quan hệ của SQL Server có tối đa một khoá chính.
- *UNIQUE*: Sử dụng dành riêng cho RBTV duy nhất. Các thuộc tính được khai báo RBTV duy nhất có thể xem như các thuộc tính của khoá ứng viên. Mỗi bảng có thể khai báo nhiều khoá ứng viên.
- *FOREIGN KEY*: Sử dụng dành riêng cho việc tạo RBTV tham chiếu hoặc khoá ngoại. Các thuộc tính khoá ngoại phải tham chiếu đến các thuộc tính khoá (khoá chính hoặc khoá ứng viên).
- *CHECK*: Sử dụng dành riêng cho việc tạo các RBTV khác. Lúc này các RBTV được mô tả như một biểu thức điều kiện mà các dữ liệu phải thoả biểu thức điều kiện đó.

Cách sử dụng các kỹ thuật này được trình bày trong phần xây dựng cấu trúc Cơ sở dữ liệu.

2.2.2 RULE

RULE được thiết lập như là một quy tắc của một thuộc tính. Trình tự các bước tạo và sử dụng RULE:

Tạo RULE:

```
CREATE RULE [Tên_Rule]
AS [Biểu_thức_mô_tả_điều_kiện]
[;]
```

Trong đó:

- [Tên_Rule]: Tên do người lập trình đặt
- [Biểu_thức_mô_tả_điều_kiện]: Biểu thức tương ứng với nội dung của RULE.
Trong biểu thức này chỉ được sử dụng 1 biến (bắt đầu bằng @) để mô tả RULE. Khi gắn RULE cho thuộc tính nào thì biến tương ứng với thuộc tính đó.

Gắn RULE cho thuộc tính:

```
EXEC sp_bindrule [ @rulename = ] 'Tên_rule',
[ @objname = ] 'Tên_bảng.Tên_thuộc_tính'
```

Ghi chú: Sinh viên tìm hiểu thêm một số cú pháp sử dụng RULE khác: sử dụng các tham số futureonly, gắn RULE cho kiểu dữ liệu...

Tháo bỏ RULE cho thuộc tính:

```
EXEC sp_unbindrule [ @objname = ] 'Tên_bảng.Tên_thuộc_tính'
```

Ghi chú: Khi gỡ bỏ RULE thì gỡ bỏ toàn bộ RULE mà đã được gắn vào thuộc tính.

Xóa RULE

```
DROP RULE [Tên_Rule]
```

Ví dụ 1: Cài đặt RBTV lương của giáo viên thuộc khoảng (\$1000, \$20000) bằng cách sử dụng RULE.

Tạo RULE để biểu diễn ràng buộc thuộc 1 khoảng.

```
CREATE RULE range_rule
AS
@range >= $1000 AND @range < $20000;
```

Gắn RULE vừa tạo cho thuộc tính Lương của bảng GIAOVIEN

```
sp_bindrule 'range_rule' , 'GIAOVIEN.Luong'
```

Khi không sử dụng RBTV này nữa thì tháo bỏ RULE khỏi thuộc tính lương.

```
sp_unbindrule 'GIAOVIEN.Luong'
```

2.3 Kỹ thuật cài đặt RBTV nâng cao: TRIGGER

2.3.1 Giới thiệu

Là một cơ chế để đảm bảo ràng buộc toàn vẹn sử dụng khả năng lập trình của Hệ quản trị cơ sở dữ liệu.

2.3.2 Cú pháp:

Tạo Trigger

```
CREATE TRIGGER [Tên_trigger]
ON [Tên_bảng]
FOR [Các_thao_tác: insert, update hoặc delete]
AS
IF UPDATE (Tên_thuộc_tính)
BEGIN
-- Thân_của_trigger: Mã nguồn kiểm tra hoặc cập nhật
END
```

Một số lưu ý khi sử dụng trigger:

- Một trigger được gắn với 1 bảng để giám sát sự thay đổi dữ liệu của bảng đó. Mã nguồn trong phần Thân_của_trigger sẽ được tự động gọi thực hiện

khi xảy ra Các_thao_tác cập nhật dữ liệu (insert, update hoặc delete) lên bảng **Tên_bảng**. Do đó nội dung mã nguồn của **Thân_của_trigger** thường sẽ thực hiện những công việc như: kiểm tra dữ liệu, thay đổi dữ liệu, hủy bỏ thao tác để làm cho ràng buộc toàn vẹn không bị vi phạm.

- Trong phần thân của trigger để dễ dàng cho các thao tác kiểm tra dữ liệu, hệ quản trị cung cấp 2 bảng tạm cho người viết trigger sử dụng. Hai bảng này có cấu trúc giống hệt như bảng chính:
 - o Bảng **inserted**: chứa những dòng mới thêm vào
 - o Bảng **deleted** chứa những dòng vừa mới bị xóa đi.
- Lưu ý: Không có bảng updated vì thao tác cập nhật được xem là bao gồm thao tác xóa và thêm mới. Khi thực hiện thao tác cập nhật, bảng inserted chứa dữ liệu mới, bảng deleted chứa dữ liệu cũ.
- Trong MS SQL Server, trigger được gọi thực hiện sau khi thao tác tương ứng (insert, update, delete) được thực hiện trên bảng chính. Người dùng nếu muốn khôi phục lại dữ liệu trong bảng chính thì gọi lệnh **rollback**. Ngoài ra sử dụng hàm **raiserror** để thông báo lỗi khi phát hiện thấy sự vi phạm RBTV.

Xoá Trigger

```
DROP TRIGGER [Tên_trigger]
```

Cập nhật nội dung Trigger:

```
ALTER TRIGGER [Tên_trigger]
ON [Tên_bảng]
FOR [Các_thao_tác: insert, update hoặc delete]
AS
IF UPDATE (Tên_thuộc_tính)
BEGIN
    ■ Thân_của_trigger: Mã nguồn kiểm tra hoặc cập nhật
    -- TH1: Mã nguồn kiểm tra
```

```

IF (điều kiện để RBTV vi phạm )
BEGIN
    raiserror (N'Lỗi: XXXX ', 16, 1)
    rollback
END

-- TH2: Mã nguồn cập nhật
UPDATE ...

...
END

```

2.3.3 Một số ví dụ:

Ví dụ 2: Cài đặt RBTV “Lương của giáo viên phải ≥ 1000 ” sử dụng kỹ thuật trigger.

Nhận xét: RBTV này liên quan đến bảng GIAOVIEN (\rightarrow ON GIAOVIEN), khi thêm mới một dòng (*insert*) hoặc cập nhật thuộc tính LUONG (*update*) thì RBTV này có khả năng bị vi phạm. Khi xoá thì sẽ không ảnh hưởng đến RBTV này (\rightarrow FOR insert, update).

	T	X	S
GIAOVIEN	+	-	+ (LUONG)

Sử dụng trigger để kiểm tra những dữ liệu mới đưa vào, nếu vi phạm quy định \rightarrow báo lỗi và khôi phục lại dữ liệu.

```

CREATE TRIGGER trgLuong
ON GIAOVIEN
FOR insert, update
AS

```



```

if update(LUONG)
BEGIN
    if exists (select * from inserted where LUONG < 1000)
    begin
        raiserror (N'Lỗi: Lương của giáo viên phải >= 1000 ', 16, 1)
        rollback
    end
END

```

Giả sử có bảng **CTHD** (MaHD, MaSP, SoLuong, DonGia, ThanhTien)

Ví dụ 3: Cài đặt RBTV sau “Thành tiền phải bằng Số Lượng * Đơn giá”

Nhận xét: RBTV này liên quan đến bảng CTHD, khi thêm mới một dòng dữ liệu hoặc khi cập nhật các thuộc tính như SoLuong, DonGia thì cần cập nhật lại ThanhTien tương ứng.

	T	X	S
CTHD	+	-	+ (SoLuong, DonGia, ThanhTien)

Sử dụng trigger để cập nhật dữ liệu đúng với RBTV.

```

CREATE TRIGGER trgCapNhatThanhTien
ON CTHD
FOR insert, update
AS
IF update(SoLuong, DonGia, ThanhTien)
BEGIN
    Update CTHD
    Set ThanhTien = SoLuong * DonGia
    Where EXISTS (select * from inserted I where i.MAHD=CTHD.MAHD AND i.MASP =

```

CTHD,MASP)

END

Ví dụ 4: Cài đặt RBTV sau “Giáo viên làm trưởng bộ môn thì phải thuộc bộ môn đó”

	T	X	S
GIAOVIEN	-	-	+ (MABM)
BOMON	-	-	+ (TRBOMON)

Giả sử đã có các RBTV khoá chính, khoá ngoại.

CREATE TRIGGER trgTruongBoMon_BOMON

ON BOMON

FOR update

AS

IF update(TRBOMON)

BEGIN

IF exists (SELECT *

FROM INSERTED I

WHERE I.TRBOMON IS NOT NULL

AND I.TRBOMON NOT IN (SELECT G.MAGV

FROM GIAOVIEN G

WHERE G.MABM=I.MABM))

BEGIN

raiserror (N'Lỗi: Trưởng bộ môn phải là người trong bộ môn ', 16, 1)

rollback

END

END

GIAOVIEN

MAGV	MABM
001			HTTT
002			CNPM
003			HTTT

BOMON

MABM	TRBOMON
HTTT			002
CNPM			002

GIAOVIEN

MAGV	MABM
001			HTTT
002			CNPM → HTTT
003			HTTT

BOMON

MABM	TRBOMON
HTTT			001
CNPM			002

```

CREATE TRIGGER trgTruongBoMon_GIAOVIEN
ON GIAOVIEN
FOR update
AS
IF update(MABM)
BEGIN
    IF exists (SELECT *
              FROM BOMON B, INSERTED I
              WHERE B.TRBOMON=I.MAGV AND B.MABM <> I.MABM)
        raiserror (N'Lỗi: Trưởng bộ môn phải là người trong bộ môn ', 16, 1)
        rollback
    END
END

```