Bên trong trigger, bạn thiết lập SET NOCOUNT để ON để ngăn chặn số lượng hàng được trả lại khi trigger bị ảnh hưởng

Trigger là một đối tượng thường được sử dụng trong CSDL với mục đích đảm bảo tính toàn vẹn của dữ liệu. Tương tự như thủ tục lưu trữ, một trigger là một đối tượng chứa một tập các câu lệnh SQL và tập các câu lệnh này sẽ được thực thi khi người sử dụng có lời gọi đến chúng còn các trigger lại được gọi tự động khi xảy ra những giao tác làm thay đổi dữ liệu trong cá bảng

Mỗi một trigger được tạo ra và gắn liền với một bảng nào đó trong CSDL. Khi dữ liệu trong bảng bị thay đổi (tức là khi bảng chịu tác động của các câu lệnh INSERT, UPDATE, DELETE) thì trigger sẽ được tự động kích hoạt.

Sử dụng trigger một cách hợp lý trong CSDL sẽ có tác động rất lớn trong việc tăng hiệu năng của CSDL. Các trigger thực sự hữu dụng với những khả năng sau:

* Một trigger có thể nhận biết, ngăn chặn hoặc hủy bỏ những thao tác làm thay đổi trái phép dữ liệu trong CSDL.
* Các thao tác trên dữ liệu (xóa, cập nhật, bổ sung) có thể được trigger phát hiện ra và tự động thực hiện một loạt các thao tác trên CSDL 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 CSDL mà bản thân các ràng buộc không thể thực hiện được.

Một trigger là một đối tượng gắn liền với một bảng và được tự động kích hoạt khi xảy ra những giao tác làm thay đổi CSDL trong bảng. Định nghĩa một trigger bao gồm các yếu tố sau:

* Trigger sẽ được áp dụng đối với bảng nào?
* Trigger được kích hoạt khi câu lệnh nào được thực thei trên bảng?
* Trigger sẽ làm gì khi được kích hoạt?

# +MỤC TIÊU

* Một trong những đặc điểm cơ bản của csdl quan hệ là tính nhất quán dữ liệu và sql server thực hiện bằng cách thiết lập các ràng buộc khóa chính, khóa ngoại.
* Tuy nhiên khi chúng ta cần tạo ràng buộc trên 2 bảng khác nhau, vd khi thêm đơn hàng, tự động bảng hàng hóa sẽ cập nhật trừ lại số lượng hàng tồn =sl hiện tại trừ đi sl đã xuất, thì lúc này sql server chưa có cơ chế để đảm bảo ràng buộc toàn vẹn dữ liệu. Cách duy nhất để xử lý chính là dùng đến trigger
* Trigger giúp ta kiểm tra ràng buộc trên nhiều bảng hoặc trên nhiều dòng của bảng.
* Việc sử dụng trigger đồng nghĩa là có những hàm chạy ngầm và trigger có thể gây gối, khó khăn cho bảo trì và phát triển hệ thống lớn do trigger ẩn với người dùng, nên ta chỉ sử dụng khi thật sự cần thiết, với các thao tác trọng tâm, Các ràng buộc sẵn có trong SQL Server thì k cần dùng trigger
* Mục tiêu của buổi học hôm nay chúng ta sẽ đi tìm hiểu tổng quan về trigger, tìm hiểu một số loại trigger và trigger DML.

# SLIDE 5: TRIGGER

* Trigger là một dạng đặc biệt của SP được thực thi một cách tự động khi có sự kiện xảy ra.
* Có nhiều loại sự kiện khác nhau để kích hoạt trigger trong [SQL](https://vietnix.vn/sql-la-gi/). Có thể kể đến như việc chèn các hàng trong bảng, thay đổi cấu trúc bảng, xóa bảng hoặc thậm chí người dùng đăng nhập vào một phiên bản SQL Server.
* Trigger được lưu trữ trong DB Server và thường hay được dùng để kiểm tra ràng buộc toàn vẹn dữ liệu.
* Phân biệt giữa SP và trigger:
* SP: thực thi bằng lệnh EXECUTE; Trigger thực hiện tự động khi có sự kiện xảy ra
* SP: có tham số hoặc k; Trigger k có tham số
* Trigger k đảm bảo khôi phục một transaction bên trong trigger

# SLIDE 6: BẢNG INSERTED VÀ DELETED

* Thông thường dữ liệu các thao tác insert, update, delete được sao lưu vào bảng ảo.
* SQL Server cung cấp 2 loại bảng ảo đặc biệt cho trigger là bảng INSERTED và DELETED.
* SQL sử dụng bảng này để thu thập dữ liệu của bảng ghi đã sửa đổi trước và sau khi sự kiện xảy ra.
* Khi da hoan tat thay doi du lieu duoc chuyen ve cac table,view tuong ung

# SLIDE 7: CÁC LOẠI TRIGGER

* Có nhiều loại trigger trong SQL Server.
* DML (Data Modification Language) trigger: trigger dữ liệu ngôn ngữ thao tác, đây là loại trigger được sử dụng nhiều nhất, được kích hoạt khi xảy ra sự kiện INSERT, UPDATE hoặc DELETE dữ liệu trên bảng.
* Bên cạnh đó, DML trigger cũng có những loại khác như:
* FOR hoặc AFTER [[INSERT, UPDATE, DELETE]: Được sử dụng sau khi lệnh kích hoạt được kết thúc.
* INSTEAD OF [INSERT, UPDATE, DELETE]: Loại này trái ngược lại hoàn toàn với FOR (AFTER). Chúng thực thi chứ không phải sử dụng để thay cho câu lệnh kích hoạt. Hiểu rõ hơn, đây là loại trigger có thể thay thế câu lệnh kích hoạt. Việc này thật sự rất hữu ích khi người dùng cần tính toàn vẹn tham chiếu database chéo.
* DDL: Data Definition Language trigger: trigger dữ liệu ngôn ngữ định nghĩa, trigger này kích hoạt khi xảy ra sự kiện CREATE (tạo bảng), ALTER(sửa bảng) và DROP (xóa bảng). Hoặc trong các sự kiện liên quan đến server như thay đổi bảo mật hoặc sự kiện cập nhật thống kê.

Ngoài ra, **DML trigger** còn có các loại khác nhau:

* **FOR** hoặc **AFTER [[INSERT, UPDATE, DELETE]**: Các loại trigger này được thực thi sau khi câu lệnh kích hoạt kết thúc.
* **INSTEAD OF [INSERT, UPDATE, DELETE]**: Trái ngược với **FOR (AFTER)**, trigger I**NSTEAD OF** thực thi thay vì thay cho câu lệnh kích hoạt. Nói cách khác, loại trigger này thay thế câu lệnh kíc
* h hoạt. Điều này rất hữu ích trong trường hợp bạn cần có tính toàn vẹn tham chiếu database chéo.

# SLIDE 8: SO SÁNH 2 TRIGGER DML VÀ DDL

* Các trigger DDL và DML có cách sử dụng khác nhau và được thực thi với các sự kiện cơ sở dữ liệu khác nhau.

# SLIDE 9: TRIGGER DML

* Các trigger DML được thực thi khi sự kiện DML (INSERT, UPDATE và DELETE.) xảy ra trong các bảng hoặc khung nhìn.
* Các trigger DML gồm ba loại chính:Trigger INSERT, Trigger UPDATE, Trigger DELETE

# SLIDE 10 BẢNG INSERTED VÀ DELETED TRONG DML

* Khi trigger được thực hiện, SQL tự động tạo ra 2 bảng tạm với cùng cấu trúc với bảng chứa trigger là bảng INSERTED và DELETED.
* Đối với lệnh INSERT, bảng INSERTED chứa các bảng ghi đã được INSERT
* Đối với lệnh DELETE, bảng DELETETED chứa các bảng ghi đã được DELETE
* Đối với lệnh UPDATE, bảng DELETETED chứa các bảng ghi trước khi UPDATE, bảng INSERTED chứa các bảng ghi sau khi được UPDATE

# SLIDE 11: SƠ ĐỒ

Chung ta se xem xet so do sau de hieu ro hon ve vai tro cua cac bang ao inserted va deleted trong cac thao tac cap nhat

Khi ta co yeu cau cap nhat insert, update, delete, ta se thuc hien chen du lieu vao bang inserted (doi voi thao tac insert va update), bang deleted doi voi thao tac delete va update

Tiep den ta di kiem tra co trigger instead of khong, neu co thi thuc hien trigger instead of thuc hien xoa du lieu trong 2 bang tam deleted va inserted va ket thuc

, neu k co thi thuc hien cap nhat du lieu vao trong csdl

Tiep theo ta kiem tra co trigger After hay k, neu co thi thuc hien trigger after neu k thi thuc hien xoa du lieu trong 2 bang tam deleted va inserted va ket thuc

# SLIDE 12: VD

Ta co du lieu trong bang client, ta thuc hien them du lieu vao bang. Trong bang clients co tao trigger test dua tren hoat dong INSERT vao trong bang client kiem tra viec them du lieu.

Khi hoat dong insert xay ra, trigger duoc kich hoat se truy van tu bang tam inserted de thay du lieu ta moi them duoc luu tam vao trong bang tam. Du lieu trong lenh INSERT o tren da duoc them vao trong bang INSERTED. Du lieu nay la du lieu tam thoi, neu cac hoat dong INSERT du lieu thanh cong va cac trigger hoat dong thanh cong du lieu se duoc di chuyen tro lai vao trong bang clients

# SLIDE 13: TRIGGER INSERT

* Trigger INSERT được thực thi khi một bản ghi mới được chèn vào bảng
* Trigger INSERT đảm bảo rằng giá trị đang được nhập phù hợp với các ràng buộc được định nghĩa trên bảng đó.
* Khi thực hiện lệnh INSERT, các bảng ghi đã được INSERT sẽ chứa trong bảng INSERTED
* Về mặt vật lý bảng Inserted không tồn tại trong cơ sở dữ liệu
* Trigger INSERT được tạo ra bằng cách sử dụng từ khóa INSERT trong câu lệnh **CREATE TRIGGER và ALTER TRIGGER**.

# SLIDE 14: CÚ PHÁP TẠO TRIGGER DML

CREATE TRIGGER têntrigger ON tenbang

[FOR|AFTER|BEFORE|INSTEAD OF] INSERT|[INSERTUPDATE|DELETE] -- thaotaccapnhat

AS

BEGIN

--lỆNH SQL

END

* Ta xét vd tạo trigger kiểm tra việc chèn dữ liệu vào bảng CLIENS.
* Cứ tạo cái gì ta CREATE cái đó. Ở đây ta tạo trigger nên

CREATE TRIGGER têntrigger ON tenbang

FOR thaotaccapnhat

AS

BEGIN

--Lenh65trongtrigger, vd ở đây hiển thị dữ liệu trong bảng tạm INSERTED

END

* Để tạo DML Trigger, trước tiên ta cần tạo trigger mới, chỉ định tên trigger
* Sau đó, cần chỉ định tên bảng mà trigger sẽ được kích hoạt ngay khi một sự kiện xuất hiện: (ON tebang)
* Liệt kê một hay nhiều sự kiện sẽ call trigger tại lệnh AFTER|BEFORE|INSTEAD OF:
* (AFTER INSERT, DELETE)
* Nội dung của Trigger sẽ được khởi đầu với từ khóa AS:
* (AS
* BEGIN)
* Trong Trigger, người dùng cần thiết lập SET NOCOUNT để ON, việc này nhằm ngăn chặn số hàng sẽ bị trả lại trong trường hợp trigger gặp vấn đề:
* (SET NOCOUNT ON;)
* Trigger sẽ chèn cột vào bảng khi một bảng ghi được insert hoặc delete. Những dữ liệu được cung cấp để chèn từ INSERTED và DELETED
* Sau cùng, bắt đầu thực hiện tất cả các câu lệnh để tiến hành tạo trigger trong SQL. Bạn có thể tìm thấy trigger được tạo trong folder trigger.
* Thời gian kích hoạt sẽ có thể là BEFORE/AFTER. Người dùng cần định rõ thời gian kích hoạt. BEFORE được sử dụng khi muốn xử lý hành động trước khi thay đổi và AFTER là cần xử lý hành động sau quá trình thay đổi.
* Sự kiện có thể là những yếu tố INSERT, UPDATE, DELETE.
* Trình kích hoạt sẽ cần được liên kết với bảng cụ thể, lúc này từ khóa ON được dùng để xác định.

# SLIDE 15: VD TRIGGER INSERT TRONG DML

* VD ta thực hiện tạo trigger tên TR\_CHECKLUONGNV để Kiểm tra dữ liệu chèn vào bảng nhân viên có lương phai lớn hơn 5000.
* Nếu cột lương trong bảng INSERTED <5000, ta thực hiện in ra màn hình lương phài lớn hơn 5000 và đồng thời dùng lệnh ROLLBACK TRANSACTION để hủy không thực hiện lệnh INSERT vừa rồi, tất cả lệnh chèn vào bị hủy, chấm dứt transaction
* Sau khi tạo trigger, giả sử ta viết lệnh INSERT để thêm nv có lương 4000<5000, lúc này kết quả hiển thị thông báo tiền lương tối thiểu 5000 và thông báo lỗi là transaction bị hủy bởi trigger không cho thực hiện INSERT
* Ở phần dưới thông báo transaction trong trigger đã kết thúc. Lô xử lý đã được hủy bỏ.

# SLIDE 16: TRIGGER UPDATE

* Trigger UPDATE sao chép bản ghi gốc vào bảng Deleted và bản ghi mới vào bảng Inserted
* Nếu các giá trị mới là hợp lệ thì bản ghi từ bảng Inserted sẽ được sao chép vào bảng dữ liệu
* Trigger UPDATE được tạo ra bằng cách sử dụng từ khóa UPDATE trong câu lệnh CREATE TRIGGER và ALTER TRIGGER.
* Cú pháp hoàn toàn tương tự với trigger INSERT

# SLIDE 17: VD TRIGGER UPDATE

* Ví dụ ta tạo trigger cập nhật lương cho nhân viên, quy định lương >5000. Khi chúng ta thực hiện cập nhật, nội dung bản ghi gốc sẽ lưu vào bảng DELETED, nội dung bản ghi mới lưu vào bảng INSERTED. Ta hiển thị cột lương trong bảng INSERTED để kiểm tra nếu lương <=5000 thì ta hiển thị thông báo lương phải lớn hơn 5000 đồng thời ROLLBACK TRANSACTION để hủy thao tác cập nhật.

CREATE TRIGGER TR\_LUONGNV ON NHANVIEN FOR UPDATE

AS

BEGIN

IF (SELECT LUONG FROM inserted) <=5000

BEGIN

PRINT N'Lương phải lớn hơn 5000'

ROLLBACK TRANSACTION

END

END

GO

* Trigger này được kích hoạt khi ta thực hiện lệnh UPDATE. Gỉa sử ta viết lệnh cập nhật nv mã 001 có lương 3000. Lúc này kết quả hiển thị thông báo lương phải lớn hơn 5000 và thao tác cập nhật bị hủy. Ta nhận được thông báo transaction đã kết thúc. Lô xử lý bị hủy.

UPDATE NHANVIEN SET LUONG=3000 WHERE MANV='001'

# SLIDE 18: TRIGGER DELETE

* Thông thường ta sử dụng trigger DELETE để hạn chế người dùng không xoá một bản ghi cụ thể trong bảng
* Khi người dùng cố gắng xóa bản ghi:
* Bản ghi bị xóa khỏi bảng trigger và chèn vào bảng Deleted
* Kiểm tra các ràng buộc liên quan khi xóa
* Nếu có ràng buộc trên bản ghi để ngăn chặn việc xóa, trigger DELETE sẽ hiển thị thông báo lỗi
* Bản ghi đã xóa được lưu trữ trong bảng Deleted được sao chép ngược lại bảng trigger.
* WITH ENCRYPTION: Đây là tùy chọn để có mã hóa trigger hay không.
* Với hành động DELETE sẽ kích hoạt trigger.

# SLIDE 19: VD TRIGGER DELETE

* Ví dụ tạo trigger Delete tên TR\_XOANV005 không cho phép xóa nhân viên có mã 005. Cú pháp tương tự trigger INSERT và UPDATE.
* Khi thực hiện trigger DELETE thì các bảng ghi bị xóa sẽ lưu vào bảng tạm DELETED. Ta hiển thị cột MANV trong bảng DELETE và dùng IF để kiểm tra tập kết quả truy vấn có chứa 005 không, nếu có ta hiển thị thông báo không thể xóa nv 005 và hủy thao tác xóa bằng lệnh ROLLBACK TRANSACTION, bảng ghi bị xóa sẽ được sao chép ngược lại vào bảng trong csdl

CREATE TRIGGER TR\_XOANV005 ON NHANVIEN FOR DELETE

AS

IF '005' IN (SELECT MANV FROM DELETED)

BEGIN

PRINT N'Không thể xóa nv 005'

ROLLBACK TRANSACTION

END

GO

DELETE NHANVIEN WHERE MANV='005'

* Để kích hoạt trigger ta gọi thực hiện câu lệnh DELETE. Gỉa sử ta viết lệnh xóa nv mã 005. Lúc này kết quả hiển thị thông báo không thể xóa nv 005 và thao tác xóa bị hủy

# SLIDE 20 DEMO VD

Khi ta tạo trigger cho bảng nào thì sẽ xuất hiện trigger trong bảng đó. Vào bảng nhanvien, chon trigger, refresh ta thấy trigger đã được tạo.

--Để xem các trigger hiện có, vào TABLES/BẢNG CHỨA TRIGGER/TRIGGERS

--Viết trigger rang buộc quy định mức lương của nhân viên

--không được <5000 cho thao tác thêm và cập nhật bảng nv.

CREATE TRIGGER TR\_CHECKLUONG ON NHANVIEN FOR INSERT,UPDATE

AS

IF (SELECT LUONG FROM inserted)<5000

BEGIN

PRINT N'Lương nv phải >=5000'

ROLLBACK TRANSACTION

END

GO

UPDATE NHANVIEN SET LUONG=2000 WHERE MANV='011'

INSERT INTO NHANVIEN VALUES ('A1','B1','C1','012','','DC','N',3000,'001',19)

--Viết trigger rang buộc không được xóa nhân viên ở TP HCM

CREATE TRIGGER TR\_XOANV ON NHANVIEN FOR DELETE

AS

IF EXISTS (SELECT \* FROM deleted WHERE DCHI LIKE N'%HCM')

BEGIN

PRINT N'Không được xóa nv ở HCM'

ROLLBACK TRANSACTION

END

GO

DELETE NHANVIEN WHERE DCHI LIKE N'%HCM'

# LAB6 B1

--Ràng buộc khi thêm mới NV thì lương phải >15000,

--nếu vi phạm thì xuất thông báo “luong phải >15000’

--DISABLE TRIGGER TR\_LUONG5000 ON DATABASE

CREATE TRIGGER TR\_CHECKLUONG ON NHANVIEN

FOR INSERT

AS

IF (SELECT LUONG FROM inserted) <=15000

BEGIN

PRINT N'Lương phải > 15000'

ROLLBACK TRANSACTION

END

GO

INSERT INTO NHANVIEN VALUES ('A','B','C1','011',' ','CT',' ',15500,'001',5)

--Ràng buộc khi thêm mới NV thì độ tuổi phải

--nằm trong khoảng 18 <= tuổi <=65.

CREATE TRIGGER TR\_CHECKTUOI ON NHANVIEN

FOR INSERT

AS

BEGIN

IF (SELECT DATEDIFF(YY,NGSINH,GETDATE()) FROM inserted) <18 OR

(SELECT DATEDIFF(YY,NGSINH,GETDATE()) FROM inserted) >65

BEGIN

PRINT N'Nhân viên phải từ 18 đến 65 tuổi'

ROLLBACK TRANSACTION

END

END

GO

INSERT INTO NHANVIEN VALUES ('B','B1','B2','012','05/30/2000','CT',' ',16000,'001',5)

--Ràng buộc khi cập nhật NV thì không được

--cập nhật những nhân viên ở TP HCM

CREATE TRIGGER TR\_CHECKDC ON NHANVIEN

FOR UPDATE

AS

BEGIN

IF EXISTS (SELECT \* FROM deleted where DCHI LIKE N'%HCM')

BEGIN

PRINT N'Không được cập nhật NV ở HCM'

ROLLBACK TRANSACTION

END

END

GO

UPDATE NHANVIEN SET PHAI='Nam' WHERE MANV='002'