Chương 2.

1. Ràng buộc toàn vẹn là gì?

* Ràng buộc toàn vẹn là tập các quy tắc mà mọi dữ liệu trong CSDL phải tuân theo nhằm đảm bảo tính toàn vẹn của CSDL.
* Có nhiều ràng buộc khác nhau trong một CSDL. Một số dàng buộc cơ bản gồm có:
* Ràng buộc unique (duy nhất): mỗi giá trị trong cột có ràng buộc này phải là duy nhất trong cột đó.
* Ràng buộc Not Null (khác rỗng): Các ô trong cột có ràng buộc này phải có giá trị khác Null.
* Ràng buộc khóa ngoại (ràng buộc tham chiếu): Các giá trị trong cột khóa ngoại ở bảng tham chiếu phải có trong cột tương ứng ở bảng được tham chiếu.
* Ràng buộc khóa chính (ràng buộc toàn vẹn thực thể): các giá trị là khóa chính phải thỏa uniue và not null.
* Ràng buộc về miền giá trị: các giá trị trong cột phải nằm trong miền giá trị của cột

1. Tại sao các quan hệ phải có ràng buộc toàn vẹn?

Các quan hệ phải có các ràng buộc toàn vẹn để nhằm đảm bảo tính toàn vẹn của CSDL.

1. Khóa chính (primary key) của một quan hệ Q

Một trong các khóa ứng viên (siêu khóa với tập thuộc tính nhỏ nhất) sẽ được chọn làm khóa chính. Trong lược đồ quan hệ khóa chính thường được đánh dấu bằng gạch dưới. Khóa chính của một quan hệ phải thỏa điều kiện khóa chính (thỏa unique và not null)

1. Khóa ngoại (foreign key) của một quan hệ Q

Một hay nhiều thuộc tính (khóa chính hay thuộc tính có ràng buộc unique và not null) trong quan hệ A xuất hiện trong quan hệ B thì được gọi là khóa ngoại ở B. Trong lược đồ quan hệ, khóa ngoại thường được đánh dấu bằng gạch đứt. Khóa ngoại phải thỏa được ràng buộc về khóa ngoại

1. Ràng buộc về miền giá trị của 1 thuộc tính

Các giá trị trong cột phải nằm trong miền giá trị của cột

1. Một thuộc tính có ràng buộc Unique?

Các giá trị trong cột phải là duy nhất

1. Một thuộc tính có ràng buộc not null nghĩa là?

Các giá trị trong cột không được phép là giá trị Null

1. Một thuộc tính có ràng buộc Check (điều kiện) nghĩa là?

Nghĩa là các giá trị tron cột phải thỏa điều kiện của ràng buộc Check

1. Câu lệnh khai báo, thêm và xóa các ràng buộc của quan hệ Q.

Ta dùng kết hợp lệnh Alter table và Drop. Cú pháp: **Alter table** tên\_quan\_hệ **DROP CONSTRAINT** tên\_riêng\_ràng\_buộc

VD:

Text

Description automatically generated

1. Cho Q1(A int, B int), Q2(A int, C int). A trong Q2 là khóa ngoại tham chiếu qua A của Q1. Thao tác nào dưới **đây không thực hiện được**
2. Thêm 1 dòng có A = 1 vào Q2, nhưng trong Q1 chưa có dòng nào có A = 1.
3. Sửa 1 dòng trong Q2 có A = 1 thành A=0, nhưng trong Q1 chưa có dòng nào có A = 0.
4. **Cả (1) và (2) đều đúng**
5. Cả (1) và (2) đều sai
6. Trigger là gì?

Trigger là một chuỗi các hành động thát liên quan tới một hặc một vài sự kiện/biến cố (events), và sự được thực hiện bất kì khi nào sự kiện/ biến cố xảy ra. Hiểu đơn giản thì trigger là một stored procedure không có tham số. Thường thì trigger được sử dụng để kiểm tra ràng buộc trên nhiều quan hệ hoặc nhiều dòng. Trigger sẽ được thực thi một cách tự động khi một trong 3 câu lệnh Insert, Update, Delete làm thay đổi dữ liệu trên bảng có chứa trigger

**Note**: xem thêm về trigger: <https://bit.ly/3Ue4L9u>

1. Ý nghĩa của biến cố, điều kiện và hành động trong trigger

* Biến cố hay còn được gọi là sự kiện (events) là một sự việc mà tại đó trigger có thể xảy ra nếu có. Các events thường là INSERT, UPDATE, DELETE
* Nếu có điều kiện thì trigger sẽ kiểm tra điều kiện xem có thỏa hay không mới thực hiện các hành động. Nếu không có điều kiện thì trigger sẽ được chạy tự động.
* Hành động trong trigger: Là phần câu lệnh sql của trigger

1. Ý nghĩa của For và After trong câu lệnh tạo trigger

For: **Khi** biến cố xảy ra thì mới thực hiện các hành động của trigger

After: **Sau** khi biến cố xảy ra thì mới thực hiện các hành động của trigger

1. Ý nghĩa của câu lệnh dưới

Create trigger AA on Q after delete as declare @new int, @old int select @old=ol.B from deleted ol if (@old >2) print ('Notify Customer Relations');

* Tạo ra một trigger tên là ‘AA’ trên bảng Q
* Câu lệnh của trigger ‘AA’ chỉ chạy sau khi lệnh DELETE được thực hiện.
* Nếu trigger thỏa điều kiện để chạy, trigger sẽ thực hiện phần lệnh sql. Nếu phần lệnh sql được thực hiện, nó sẽ gắn giá trị của ol.B (thuộc bảng DELETED) vào @old và kiểm tra xem @old có lớn hơn 2 hay không. Nếu @old lớn hơn 2 thì in ra thông báo ‘Notify Customer Relations’. Ngược lại, không làm gì.

**Note:** Khi trigger thực hiện, nó sẽ tạo ra 2 bảng **Inserted** và **Deleted** giống hệt cấu trúc bảng mà trigger đang thực thi trên đó. Các bảng này chỉ tồn tại trong thời gian thực hiện trigger mà thôi. Xem thêm tại: <https://bit.ly/3xuWnIV>

1. Instead of trigger là gì?

Trigger **Instead of** là một trigger cho phép bỏ qua một câu lệnh INSERT, DELETE hoặc UPDATE một bảng/ view và thực thi câu lệnh được định nghĩa trong trigger. Các thao tác (có thể hiểu là sự kiện) chèn, xóa, cập nhật thực tế hoàn toàn không xảy ra. Nói cách khác, thay vì nhẽ ra phải xảy ra sự kiện mới xảy ra trigger, thì trigger Instead of cho phép bỏ qua sự kiện và thực hiện hành động của trigger luôn.

1. Ý nghĩa của câu lệnh dưới

Create trigger AA on Q after delete as declare @old int select @old=ol.B from deleted ol if (@old >2) Begin print ('Notify Customer Relations'); Rollback; end

* Tạo trigger tên ‘AA’ trên bảng Q sau khi xảy ra sự kiến **delete**
* Sau lệnh delete, trigger thực hiện phần lệnh sql.
* Khi chạy lệnh sql, trigger sẽ lấy giá trị của cột B trong bảng ol (bảng deleted) gán vào @old. Tiếp tới, trigger sẽ kiểm tra điều kiện @old>2. Nếu thỏa, thực hiện hành động in câu ‘**Notify Customer Relations**’ ra

1. Cho

Create trigger AA on Q after delete as declare @old int select @old=ol.B from deleted ol if (@old >2) Begin print ('Notify Customer Relations'); Rollback; end

Từ khóa “after” trong câu lệnh trên có nghĩa là gì?

Từ khóa **after** trong câu lệnh trên có nghĩa là trigger chạy sau khi lệnh delete được thực hiện

1. Cho

Create trigger AA on Q for delete as declare @old int select @old=ol.B from deleted ol if (@old >2) Begin print ('Notify Customer Relations');

Từ khóa “for” trong câu lệnh trên có nghĩa là gì?

Từ khóa **after** trong câu lệnh trên có nghĩa là trigger chạy trong khi lệnh delete được thực hiện

* Tạo trigger tên ‘AA’ trên bảng Q sau khi xảy ra sự kiến **delete**
* Sau lệnh delete, trigger thực hiện phần lệnh sql.
* Khi chạy lệnh sql, trigger sẽ lấy giá trị của cột B trong bảng ol (bảng deleted) gán vào @old. Tiếp tới, trigger sẽ kiểm tra điều kiện @old>2. Nếu thỏa, thực hiện hành động in câu ‘**Notify Customer Relations**’ ra

1. Ý nghĩa của câu lệnh dưới

Create trigger BB on Q1 instead of insert as Declare @newA int, @newB int Select @newA = n.A, @newB = n.B From Inserted n Insert into Q2 values(@newA, @newB)

* Tạo trigger tên ‘BB’ trên bảng Q1 sau khi xảy ra sự kiến **delete**
* Sau lệnh delete, trigger thực hiện phần lệnh sql.
* Khi chạy lệnh sql, giá trị n.A được gán cho @newA, giá trị n.B (với n là bảng inserted) được gán cho @newB. Sau đó giá trị của @newA và @newB sẽ được inserted vào bảng Q2.

1. View là gì?
2. Mục đích của sử dụng view
3. Trong SQL server, cú pháp câu lệnh tạo, xóa, sửa view?
4. Phân loại View?

Trong sql, view giống như bảng ảo được tạo ra bằng lệnh truy vấn viết sẵn và lưu bên trong cơ sở dữ liệu. Một View bao gồm một câu lệnh Select và khi chạy thì view sẽ hiện lên giống như mở một table. Có **3 loại view cơ bản** sau đây:

* Standard view: view được tạo ra bao gồm các cột cảu bảng hoặc các view khác.
* Indexed view: view được tạo ra và được đặt chỉ mục Unique Clustered Index.
* Pointed view: view được tạo bao gồm các dữ liệu được phân cụm ngang từ một hoặc nhiều

1. Ý nghĩa của câu lệnh

Create view cc as Select A, max(B) as ln from Q1 group by A;

Tạo ra một view tên là ‘c’ với dữ liệu được lấy từ bảng Q1. Cụ thể thì kết quả trả ra một bảng có 2 cột, một cốt là giá trị max của Q1.B, cột còn lại là giá trị Q1.A tương ứng với max(B)

1. Cho PBan(MaPB int, TenPB char(20), MaNQL int). Viết câu lệnh thêm ràng buộc đảm bảo mỗi phòng ban có 1 người quản lý

ALTER TABLE PBan

ALTER COLUMN MaNQL int NOT\_NULL;

1. Cho NV(MaNV int, Hoten char(30), Luong int) và câu lệnh alter table Emp add constraint chk\_luong\_min check (salary > 10000). Giải thích ý nghĩa của câu lệnh này.

Thay đổi bảng nhân viên bằng việc thêm ràng buộc check (được đặt tên là chk\_luong\_min) với điều kiện là salary>10000.

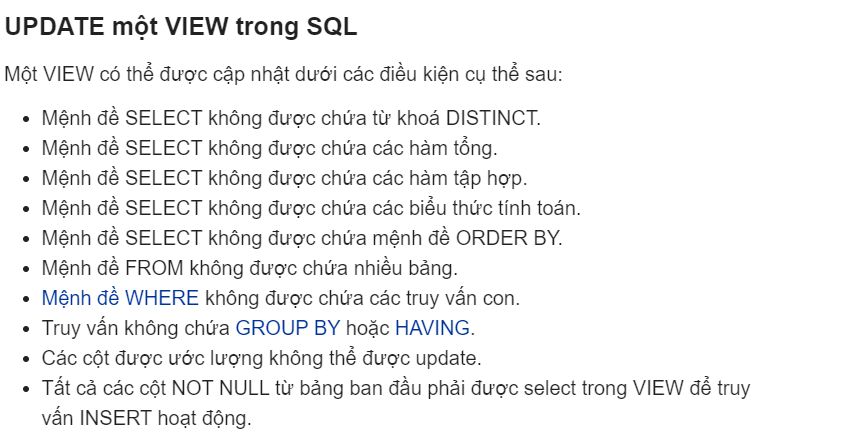
1. Cho NV(MaNV int, Hoten char(30), Luong int, MaPB int) và view được định nghĩa như sau Create view V\_NV(MaPB, SoNV, TongLg, LgTB) as Select MaPB, Count(\*), Sum(Luong), Avg(Luong) From NV Group by MaPB;

Kết quả của câu trên: Một view tên V\_NV được tạo ra gồm các cột MaPB, SoNV, TongLg, LgTB.

Các câu SQL nào thực hiện được trên view V\_NV

1. **Select \* From V\_NV**
2. Delete from V\_NV Where MaPB = 1
3. Cả (1) và (2) đều đúng
4. Cả (1) và (2) đều sai

Giải thích: câu 2 không thực hiện được vì câu truy vấn của View có chưa Group by.



<https://bit.ly/3Lm179m>