Chương 2.

1. Ràng buộc toàn vẹn là gì?

Tập các quy tắc mà mọi dữ liệu trong CSDL phải tuân theo nhằm đảm bảo tính toàn vẹn ( để đảm bảo dữ liệu luôn luôn đúng) của cơ sở dữ liệu.

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

nhằm đảm bảo tính toàn vẹn ( để đảm bảo dữ liệu luôn luôn đúng) của cơ sở dữ liệu

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

**Khóa chính (primary key):** Một trong các khóa ứng viên sẽ được chọn làm **khóa chính** cho quan hệ.

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

Một hay nhiều thuộc tính (khóa chính hay thuộc tính có ràng buộc unique) trong một quan hệ A xuất hiện trong một quan hệ B khác được gọi là khóa ngoại trong quan hệ B.

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?

mỗi giá trị trong cột có ràng buộc này phải là duy nhất trong cột đó.

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

Các ô trong cột có ràng buộc này bắt buộc phải có giá trị khác null

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

Là thuộc tính phải thỏa điều kiện check của rảng buộc đó

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

Khai báo ở mức cột :

+ col datatype Check(condition)

Khai báo ở mức bảng:

+CONSTRAINT name\_constrain check( condition)

Thêm:

ALTER TABLE table\_name ADD CONSTRAINS condition\_name condition(col) ( tuỳ chọn áp dụng cho mức cột hay bảng )

+ Xoá:

ALTER TABLE table\_name DROP CONSTRAINS condition\_name

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 loại stored procedure đặc biệt (không có tham số) được thực thi (execute) một cách tự động khi có một sự kiện thay đổi dữ liệu (data modification)

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

- Trigger chỉ được gọi khi xảy ra một số sự kiện nhất định do người lập trình cơ sở dữ liệu chỉ định (thường là chèn, xoá, cập nhật)

- Sau khi được gọi bởi sự kiện kích hoạt của nó, trình kích hoạt sẽ kiểm tra một điều kiện

- Nếu điều kiện của trình kích hoạt được thoả mãn, thì hành động được liên kết với trình kích hoạt sẽ được thực hiện bởi DBMS

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

FOR: trigger sẽ được thực hiện **trước** khi thay đổi dữ liệu thực sự được áp dụng vào cơ sở dữ liệu. "FOR" mà không có "AFTER" hoặc "INSTEAD OF" đi kèm, thì **"AFTER"** được coi là giá trị mặc định

AFTER:

+chỉ định rằng DML trigger chỉ được kích hoạt khi tất cả các hoạt được chỉ định trong phần sql\_statement trigger đã thực thi thành công. Tất cả các hành động **referential cascade**  và **constraint checks** cũng phải thành công trước khi trigger kích hoạt.

+ không thể sử dụng AFTER cho mức **view**

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 trigger tên AA trong table Q, sau khi có hành động delete trong Q, trigger được thực hiện --> tạo ra hai biến old và new, old = những cái vừa bị xoá trong cột B,nếu old > 2 thì xuất ra thông báo

1. Instead of trigger là gì?

-instead of: Chỉ định rằng trình kích hoạt DML được thực thi thay vì câu lệnh SQL kích hoạt, do đó, ghi đè hành động của câu lệnh kích hoạt. INSTEAD OF không thể được chỉ định cho DDL hoặc trình kích hoạt đăng nhập. Tối đa, một trình kích hoạt INSTEAD OF cho mỗi câu lệnh INSERT, UPDATE hoặc DELETE có thể được xác định trên một bảng hoặc dạng xem. Trình kích hoạt INSTEAD OF không được phép trên các chế độ xem có thể cập nhật sử dụng WITH CHECK OPTION.

- DML trigger thường được sử dụng để thực thi các quy tắc và dữ liệu kinh doanh

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 ra trigger tên AA trong bảng Q, trigger thực hiện sau khi thực hiện lệnh xoá trong bảng Q , khai báo biến old bằng cái vừa bị xoá trong cột B, nếu old > 2 thì in ra thông báo 'Notify Customer Relations' và khôi phục dữ liệu vừa xoá.

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?

Khi biến cố delete xảy ra thì trigger sql statement là declare @old int select @old=ol.B from deleted ol if (@old >2) Begin print ('Notify Customer Relations'); Rollback; end , khi thực hiện thành công và không bị rollback thì delete ( dml trigger ) mới được chính thức 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 khoá for có nghĩa khi lệnh delete được thực hiện. Trigger cũng đồng thời được kích hoạt Sql statement ( khai báo bién old bằng cái vừa bị xoá trong cột B nếu > 2 thì in ra 'Notify Customer Relations , cuối cùng dữ liệu bị xoá từ lệnh delete chính thức tác động lên bảng Q.

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)

+ khi có một lệnh insert và bảng Q1 được thực hiện trên bảng Q1 thì trigger được kích hoạt thay vì lệnh insert vào bảng Q1.

+ Khai báo hai biến newA và newB kiểu dữ liệu int

+ Thực hiện truy vấn để newA , newB lần lượt bằng giá trị cột "A", "B" trong bảng Inserted ( bảng chứa dữ liệu tạm thời sẽ được insert vào Q1)

+ Thực hiện lệnh Insert để chèn hai giá trị newA,newB vào bảng Q2.

1. View là gì?

Dựa vào thông tin trên, một view là một bảng ảo trong cơ sở dữ liệu. Nó không lưu trữ dữ liệu thực tế mà thay vào đó lấy dữ liệu từ một hoặc nhiều bảng gốc (base tables) hoặc các view khác trong cơ sở dữ liệu thông qua một truy vấn. View cho phép bạn xem và truy vấn dữ liệu từ các bảng gốc một cách thuận tiện bằng cách tạo một giao diện ảo với cấu trúc cột và dòng được xác định bởi truy vấn đó.

1. Mục đích của sử dụng view

* **Tập trung, Đơn giản hóa và Tùy chỉnh Quan điểm Người dùng:**

Chế độ xem cho phép bạn tùy chỉnh và đơn giản hóa cách mỗi người dùng tương tác với cơ sở dữ liệu. Những người dùng hoặc ứng dụng khác nhau có thể có các chế độ xem phù hợp riêng, chỉ hiển thị dữ liệu liên quan, giúp làm việc với cơ sở dữ liệu dễ dàng hơn.

* **Cơ chế Bảo mật**: Chế độ xem có thể đóng vai trò là cơ chế bảo mật bằng cách cấp cho người dùng quyền truy cập vào dữ liệu thông qua chế độ xem trong khi hạn chế hoặc không cấp cho họ quyền trực tiếp để truy cập vào các bảng cơ sở bên dưới. Điều này cho phép kiểm soát chi tiết hơn việc truy cập dữ liệu và giúp bảo vệ thông tin nhạy cảm.
* **Giao diện Tương thích ngược**: Các khung nhìn có thể được sử dụng để cung cấp giao diện tương thích ngược khi lược đồ của bảng đã thay đổi. Thay vì sửa đổi các truy vấn và mã hiện có, bạn có thể cập nhật dạng xem để trình bày dữ liệu theo cách duy trì khả năng tương thích với các ứng dụng hoặc truy vấn hiện có.

1. Trong SQL server, cú pháp câu lệnh tạo, xóa, sửa view?

Declare by:

CREATE VIEW view\_name AS  
 SELECT column\_name(s)  
 FROM table\_name  
 WHERE condition;

Drop by:

DROP VIEW <name>

Modify by:

ALTER VIEW view\_name AS  
 SELECT column\_name(s)  
 FROM table\_name  
 WHERE condition;

1. Phân loại View?

View và Un-updatable Views

1. Ý nghĩa của câu lệnh

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

Truy vấn nhóm các bộ theo thuộc tính A, và tìm max thuộc tính B ở mỗi nhóm, tạo một view cc có hai cột gồm A và B các thuộc tính A và max thuộc tính B ở mỗi nhóm trong Q1.

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 ADD CONSTRAINT CHECK (MaNQL IS 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.

Thêm vào bảng NV một constraints tên chk\_luong\_min và thực hiện ràng buộc salary trong bảng NV phải lớn hơn 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;

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