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 của cơ sở dữ liệu

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

Để đảm bảo tính toàn vẹn của cơ sở dữ liệu

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

Các giá trị của khóa chính phải là UNIQUE và NOTNULL

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

Các giá trị trong cột khóa ngoại của bảng tham chiếu phải có trong cột tương ứng của bảng được tham chiếu

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

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

1. Một thuộc tính có ràng buộc Unique?

Gía trị của mỗi tuple của thuộc tính đó phải khác nhau

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

Gía trị của thuộc tính đó không được null

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

Khi insert , dữ liệu vào phải thỏa điều kiện Check của thuộc tính đó mới insert dữ liệu được

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

+khai báo :

CREATE TABLE MovieStar (

name CHAR(30) CONSTRAINT *NamelsKey* PRIMARY KEY,

address VARCHAR(255),

gender CHAR(1),

birthdate DATE,

**CONSTRAINT *RightTitle***

CHECK (gender = ’F’ OR name NOT LIKE ’Ms.%’ )

);

+Thêm :

ALTER TABLE name\_table ADD CONSTRAINT tên\_ràng\_buộc ĐiềuKiệnRàngBuộc(tên thuộc tính)

+Xóa

ALTER TABLE name\_table ADD CONSTRAINT tên\_ràng\_buộc ĐiềuKiệnRàngBuộc(tên thuộc tính)

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 hoạt động có liên quan đến các sự kiện nhất định và Trigger được thực hiện bất cứ khi nào các sự kiện này thực thi

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

Biến cố(certain event)

Trigger thực thi chỉ khi các biến cố đặc biệt xảy ra bởi các chương trình của database ví dụ như insert , delete , update

Nếu điều kiện của trigger thỏa , những hoạt động có liên quan đến trigger sẽ được thực thi bởi hệ quản trị cơ sở dữ liệu

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

AFTER chỉ định rằng Trigger chỉ được kích hoạt khi các câu lệnh thao tác dữ liệu liên quan đến Trigger(DML Trigger) được thực thi thành công.

Tất cả những referential cascade actions và constraint checks phải đúng trước khi trigger này được kích hoạt

Referential cascade actions là những hành động hệ thống thực hiện khi thay đổi dữ liệu xảy ra làm ảnh hưởng tới các quan hệ tham chiếu giữa các bảng

Ví dụ về Referential cascade actions

Giả sử bạn có hai bảng trong cơ sở dữ liệu của mình: **Orders** và **OrderDetails**. Bảng **OrderDetails** lưu thông tin chi tiết về từng mục đặt hàng trong **Orders**, và có một quan hệ khóa ngoại giữa hai bảng.

Bạn muốn tạo một trigger sau mỗi lần một dòng mới được chèn vào bảng **Orders**, trigger sẽ tự động thêm một số dòng mới vào bảng **OrderDetails** với các thông tin mặc định.

Tuy nhiên, bạn đã thiết lập một ràng buộc kiểm tra trên bảng **OrderDetails** sao cho mỗi đơn hàng chỉ có thể có tối đa 10 mục đặt hàng.

Giờ đây, nếu bạn cố gắng chèn một dòng mới vào bảng **Orders** và việc này làm cho số lượng mục đặt hàng của đơn hàng đó vượt quá 10, thì việc chèn dòng mới sẽ thất bại do ràng buộc kiểm tra. Trigger sau đó sẽ không được kích hoạt vì trigger yêu cầu thực hiện hành động sau khi chèn thành công một dòng vào bảng **Orders**. Do đó, trong trường hợp này, trigger không được kích hoạt khi các điều kiện không được đáp ứng, tức là việc chèn dòng mới vào **Orders** thất bại vì ràng buộc kiểm tra.

After không thể được định nghĩa trên 1 View

For thì có thể được định nghĩa trêm 1 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 một trigger AA ở bảng Q , sau khi câu lệnh delete thực thi xong trên bảng Q , các hành động liên quan đến trigger sẽ được thực thi :

-Khai báo biến @new và @old kiểu int

chúng ta sử dụng câu lệnh SELECT để lấy giá trị của cột B từ bảng tạm thời "Deleted" có tên là ol , đây là một bảng ảo chứa các bản ghi mới được được xóa từ bảng Q , gán vào biến @old

-Nếu giá trị @old lớn hơn 2 thì in ra view Notify Customer Relations

1. Instead of trigger là gì?

Là một loại trigger , nó sẽ thực hiện các DML Trigger (những hành động được định nghĩa trong trigger ) thay vì thực thi các triggering SQL statements,ghi đè lên các triggering SQL statements

VD instead of delete ( Triggering SQL statements) as statements (DML Trigger)

Instead of không được sử dụng cho updatable views(sử dụng check constraint)

Instead of không thể thực thi cho Ngôn ngữ định nghĩa dữ liệu (DDL) hoặc cho Logon trigger

Trên mỗi câu lệnh INSERT , UPDATE , hoặc DELETE trên một bảng hoặc view chỉ có thể có tối đa 1 trigger INSTEAD OF được định nghĩa

VD : -- Trigger 1

CREATE TRIGGER InsteadOfUpdate1

ON Products

INSTEAD OF UPDATE

AS

BEGIN

-- Các hành động của trigger 1

END;

GO

-- Trigger 2

CREATE TRIGGER InsteadOfUpdate2

ON Products

INSTEAD OF UPDATE

AS

BEGIN

-- Các hành động của trigger 2

END;

Sai bởi vì trong 1 bảng Products , chỉ có thể định 1 trigger UPDATE

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 một trigger AA ở bảng Q , sau khi câu lệnh delete thực thi xong trên bảng Q , các hành động liên quan đến trigger sẽ được thực thi :

-Khai báo biến @new và @old kiểu int

chúng ta sử dụng câu lệnh SELECT để lấy giá trị của cột B từ bảng tạm thời "Deleted" có tên là ol , đây là một bảng ảo chứa các bản ghi mới được được xóa từ bảng Q , gán vào biến @old

-Nếu giá trị @old lớn hơn 2 thì in ra view Notify Customer Relations và hủy bỏ mọi thay đổi được thực hiện trong cơ sở dữ liệu quay về trạng thái trước khi câu lệnh delete bắt đầu

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?

sau khi câu lệnh delete thực thi xong trên bảng Q , các hành động liên quan đến trigger sẽ được thực thi

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?

Sau khi câu lệnh delete thực thi xong trên bảng Q , các hành động liên quan đến trigger sẽ được thực thi

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 một trigger BB ở bảng Q1, nếu câu lệnh insert được thực thi vào Q1 thì câu lệnh insert không được thực thi thay vào đó các hành động liên quan các hành động liên quan đến trigger sẽ được thực thi(trigger BB) (ghi đè lên câu lệnh insert) :

-Khai báo biến @newA và @newB kiểu int

- Tiếp theo, chúng ta sử dụng câu lệnh SELECT để lấy giá trị của cột A và cột B từ bảng tạm thời "Inserted", đây là một bảng ảo chứa các bản ghi mới được chèn vào bảng Q1 , gán vào biến @newA và @newB

- Sau đó thực hiện insert vào bảng Q2 với @newA,@newB

1. View là gì?

View là một bảng ảo bao gồm hàng và cột được định nghĩa bởi những truy vấn dữ trên một hay nhiều bảng ( base tables ) hoặc những views khác trong database

1. Mục đích của sử dụng view

Tập trung , đơn giản hóa và điều chỉnh theo nhận thức của mỗi người dùng database

Một cơ chế bảo mật an toàn cho phép người dùng truy cập dữ liệu thông qua view

Không cho phép người dùng truy cập chính xác cơ sở dữ liệu bên dưới

Cung cấp tính năng backward compatible(tương thích ngược) để mô phỏng sự thay đổi của lược đồ cơ sở dữ liệu

VD Một ứng dụng cần sài 1 view

CREATE VIEW EmployeeView AS

SELECT EmployeeID, FullName, Department

FROM Employees

WHERE IsActive = 1;

Lúc này trong Employees được thêm một thuộc tính mới vào thì chỉ cần tạo lại view mới có tên giống view cũ

CREATE VIEW EmployeeView AS

SELECT EmployeeID, FullName, Department, PhoneNumber

FROM Employees

WHERE IsActive = 1;

Vậy là chương trình ứng dụng không bị ảnh hưởng khi vẫn sử dụng view trước đó vẫn có thể hoạt động mà không có thay đổi gì

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

Câu lệnh tạo :

CREATE VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition;

Câu lệnh xóa

DROP VIEW <name>

Câu lệnh sửa view

ALTER VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition;

1. Phân loại View?

View được cập nhật và View không được cập nhật ( Un-updatable View)

View không được cập nhật là view được định nghĩa bởi sử dụng groups và các hàm tổng hợp ( aggregate function) sẽ không được update

View được định nghĩa bởi nhiều bảng bằng cách sử dụng joins cũng không được cập nhật

Aggregate function là các hàm kiểu : COUNT(),MAX(),MIN(),SUM(),…

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 cc có cột thuộc tính A và ln ( giá trị của ln là giá trị max của thuộc tính B ) từ bảng Q1 , GROUP BY bởi thuộc tính A

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 RangBuocKhoaChinh MaPB PRIMARY KEY

ALTER TABLE Pban ADD CONSTRAINT RangBuocQuanLi MaNQL 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 ràng buộc tên chk\_luong\_min vào bảng NV , ràng buộc có ý nghĩa giá trị của thuộc tính salary 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