Nguyễn - -Nhóm 6

Chương 2.

R Câu 1 Ràng buộc khóa ngoại là gì? Tại sao các ràng buôc như vậy là quan trọng? Tính toàn vẹn tham chiếu là gì?

Ràng buộc khóa ngoại ( ràng buộc tham chiếu ) là quy tắt, quy luật của DBMS về việc INSERT, DELETE, UPDATE dữ liệu giữa hai bảng có mối quan hệ ràng buộc khóa ngoại với nhau, để đảm bảo tính đúng đắn của dữ liệu. Các thuộc tính được tham chiếu của quan hệ thứ hai phải được khai báo DUY NHẤT hoặc KHÓA CHÍNH cho quan hệ của chúng.

Ràng buộc khóa ngoại trong bảng cơ sở dữ liệu quan trọng là vì giúp ngăn cản chèn thêm bất kỳ dữ liệu không hợp lệ trong trường khóa ngoại vì trường khóa ngoại chỉ chấp nhận những giá trị đã có trong trường khóa chính.

Tính toàn vẹn tham chiếu là đảm bảo tính đúng đắn của dữ liệu trong quá trình insert, delete, update của các bảng có ràng buộc khóa ngoại.

R Câu 2. Cho lược đồ CSDL gồm các lược đồ quan hệ sau:



1. Liệt kê tất cả các ràng buộc khóa ngoại giữa các quan hệ này.

Các ràng buộc khóa ngoại :

o Sid : - là khóa chính trong bảng Students, nhưng lại xuất hiện trong bảng Enrolled đóng vai trò là khóa ngoại kết nối 2 bảng

o Fid : - là khóa chính trong bảng Faculty, nhưng lại xuất hiện trong bảng Teaches đóng vài trò là khóa ngoại kết nối 2 bảng

o cid : - là khóa chính trong bảng Course, nhưng lại xuất hiện trong bảng Teaches và bảng Enrolled đóng vai trò là khóa ngoại

o rno : - là khóa chính trong bảng Rooms, nhưng lại xuất hiện trong bảng Meets\_In đóng vai trò là khóa ngoại kết nối giữa 2 bảng

1. Cho một ví dụ về một ràng buộc (hợp lý) bao gồm một hoặc nhiều quan hệ trên nhưng không phải là ràng buộc khóa chính hay khóa ngoại.

Vd: Không có môn nào vượt quá 10 tín chỉ

1. Viết các câu lệnh tạo CSDL trên bao gồm tất cả ràng buộc có thể có.

CREATE DATABASE School;

use School;

CREATE TABLE Students (

studentid nvarchar(10) primary key,

name nvarchar(max),

login varchar(50),

age int,

gpa float

);

CREATE TABLE Faculty (

fid nvarchar(10) primary key,

fname nvarchar(max),

sal float

);

CREATE TABLE Courses (

cid nvarchar(10) primary key,

cname nvarchar(50),

credits int,

check(credits < 10)

);

CREATE TABLE Rooms (

rno int primary key,

raddress nvarchar(200),

capacity int

);

CREATE TABLE Enrolled (

studentid nvarchar(10) FOREIGN KEY REFERENCES Students(studentid),

cid nvarchar(10) FOREIGN KEY REFERENCES Courses(cid),

grade varchar(10)

);

CREATE TABLE Teaches (

fid nvarchar(10) FOREIGN KEY REFERENCES Faculty(fid),

cid nvarchar(10) FOREIGN KEY REFERENCES Courses(cid),

);

CREATE TABLE Meets\_In (

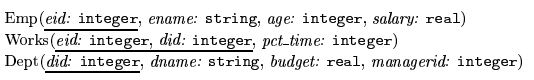
cid nvarchar(10) FOREIGN KEY REFERENCES Courses(cid),

rno int FOREIGN KEY REFERENCES Rooms(rno),

time varchar(20)

);

R Câu 3. Cho lược đồ CSDL gồm các lược đồ quan hệ sau:



Hãy trả lời các câu hỏi sau :

1. Cho một ví dụ về ràng buộc khóa ngoại có liên quan tới quan hệ Dept. Các lựa chọn nào có thể thực hiện để tuân thủ ràng buôc này khi người dùng xóa một bộ trong quan hệ Dept.

Vd: dept.managerid là khóa ngoại tham chiếu đến emp.eid. Khi xóa 1 bộ trong quan hệ dept thì không có ràng buộc nào để tuân thủ.

1. Viết các câu lệnh SQL để tạo các quan hệ trên bao gồm tất cả các RBTV khóa chính và khóa ngoại.

Ghi chú: tạo tất cả quan hệ và ràng buộc (kể cả managerid)

1. Dùng SQL định nghĩa lại quan hệ Dept sao cho mọi department được đảm bảo có một người quản lý.

Ghi chú: tạo 1 trigger mọi department được đảm bảo có một người quản lý

1. Viết một câu lệnh SQL để thêm nhân viên ‘John Doe’ với eid = 101, age = 32, và salary = 15.000

Ghi chú: insert

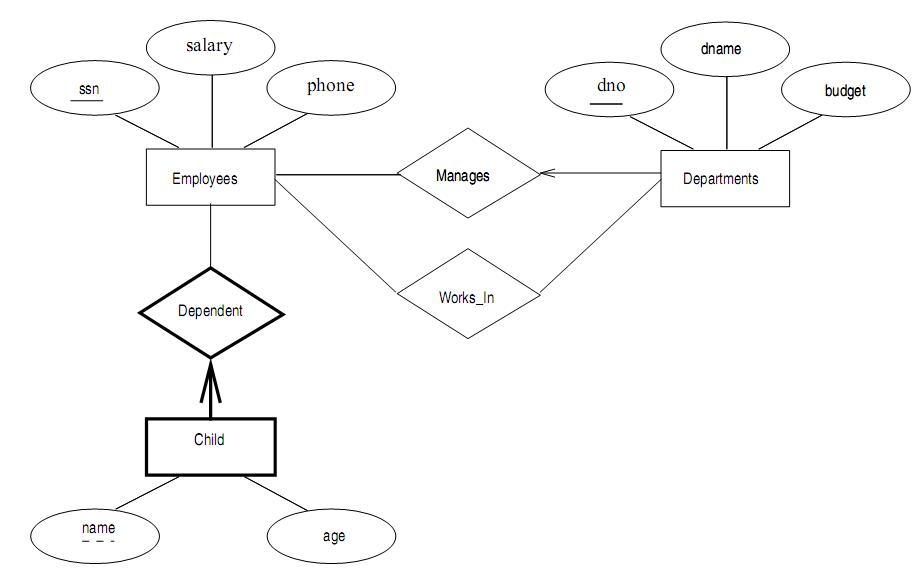
1. Viết một câu lệnh SQL để tăng 10% lương cho mọi nhân viên.

Trigger: set

1. Viết câu lệnh SQL để xóa department ‘Toy’. Với ràng buộc toàn vẹn tham chiếu bạn đã chọn cho lược đồ này, hãy giải thích điều gì xảy ra khi câu lệnh xóa này được thực hiện.

R Câu 4. Giả sử bạn đã thiết kế lược đồ ER cho CSDL một công ty như sau. Hãy viết các câu lệnh SQL để tạo các quan hệ tương ứng bao gồm nhiều nhất các ràng buộc có thể.

Nếu có ràng buộc nào bạn không định nghĩa được, hãy giải thích tại sao.



create database Emp

using Emp

create table Employees(

SSN int PRIMARY KEY,

Salary real,

Phone nvarchar(50)

);

create table Departments(

Dno int PRIMARY KEY,

Budget real,

DName nvarchar(50)

);

create table Works\_in(

SSN int,

Dno int,

PRIMARY KEY (SSN,Dno),

FOREIGN KEY (SSN) references Employees(SSN),

FOREIGN KEY (Dno) references Departments(Dno)

);

create table Manages(

SSN int,

Dno int,

PRIMARY KEY (Dno),

FOREIGN KEY (SSN) references Employees(SSN),

FOREIGN KEY (Dno) references Departments(Dno)

);

create table Dependents(

SSN int,

Name nvarchar(50),

Age int,

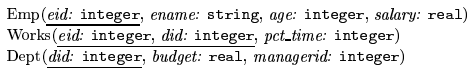
PRIMARY KEY (SSN,Name),

FOREIGN KEY (SSN) references Employees(SSN)

);

Câu 5.

Cho lược đồ quan hệ sau:



Hãy trả lời các câu hỏi sau.

1. Định nghĩa một **ràng buộc** mức bảng trên Emp sao cho đảm bảo mọi nhân viên đều có lương ít nhất là 10.000

ALTER TABLE Emp ADD Check(salary > 10000);

1. Định nghĩa một **ràng buộc** mức bảng trên Dept sao cho đảm bảo tất cả người quản lý đều có tuổi lớn hơn 30.

CREATE FUNCTION checkAgeManager(@managerid int)

RETURNS int AS

BEGIN

DECLARE @age int;

SELECT @age = E.age

FROM Emp E

WHERE E.eid = @managerid;

IF (@age IS NULL)

RETURN 0;

IF (@age > 30)

RETURN 1;

RETURN 0;

END;

ALTER TABLE Dept ADD CONSTRAINT checkAgeManagerCS CHECK (dbo.checkAgeManager(managerid) = 1);

1. Định nghĩa một assertion trên Dept sao cho đảm bảo là mọi người quản lý đều có tuổi lớn hơn 30. So sánh assertion này với ràng buộc tạo ở câu 2. Hãy giải thích cái nào tốt hơn.

CREATE TRIGGER checkManagerAgeTr on Dept

AFTER UPDATE as

declare @oldmanagerid int, @newmanagerid int, @newage int, @did int

select @oldmanagerid=ol.managerid, @newmanagerid=ne.managerid, @did=ol.did

from inserted ne, deleted ol

where ne.managerid=ol.managerid

select @newage=Emp.age

from Emp

where Emp.eid=@newmanagerid

if (@newage < 30)

Begin

print 'error'

rollback

end

1. Viết câu lệnh SQL để xóa tất cả thông tin về những Employees có lương cao hơn lương của người quản lý của họ.

Phải đảm bảo là tất cả các ràng buộc toàn vẹn liên quan phải được thỏa mãn sau khi cập nhật.

Đặt bối cảnh và các vấn đề:

a. Một nhân viên có lương cao hơn người quản lý của họ, nhưng nhân viên đó đang có các task công việc ở bảng Works, nếu xoá người nhân viên, DBMS sẽ thông báo lỗi (hoặc xoá lan truyền trên bảng Works)

b. Một nhân viên có thể làm việc ở nhiều phòng ban, nếu người này có lương thấp hơn người quản lý của họ, nhưng lại cao hơn với người quản lý phòng ban khác mà họ cũng đang làm việc, thì thật sự rất khó phân định.

c. Một quản lý của phòng ban A có thể là nhân viên làm việc với phòng ban B, trường hợp người QLA có lương cao hơn người quản lý B. Việc xoá người QLA sẽ khiến DBMS thông báo lỗi.

Đặt bối cảnh có thể thực hiện được lệnh xoá theo đề bài mà không lo lắng về vi phạm ràng buộc: 1 quản lý chỉ quản lý và làm việc với 1 phòng ban của mình. Các nhân viên có thể làm việc với nhiều phòng ban, nếu lương của họ cao hơn bất kỳ người quản lý nào mà họ làm việc cùng thì nhân viên đó sẽ bị xoá.

delete from Emp

where Emp.eid in (select Emp.eid

from Emp as Employ, Emp as Manager, Works, Dept

where Manager.eid = Dept.managerid and

Dept.did = Works.did and

Works.eid = Employ.eid and

Employ.salary > Manager.salary

);

R Câu 6. Hãy cho biết điểm mạnh và điểm yếu của cơ chế trigger. Trình bày sự tương phản giữa trigger với các ràng buộc toàn vẹn khác được hỗ trợ bởi SQL.

**Trả Lời:**

Điểm mạnh của trigger:

-        Sử dụng trigger để kiểm tra tính toàn vẹn của csdl.

-        Trigger có thể bắt lỗi logic nghiệp vụ (business logic) ở mức csdl.

-       Có thể dùng trigger là một cách khác để thay thế việc thực hiện những công việc hẹn giờ theo lịch.

-         Trigger rất hiệu quả khi được sử dụng để kiểm soát những thay đổi của dữ liệu trong bảng.

Điểm yếu của trigger:

-         Trigger chỉ là một phần mở rộng của việc kiểm tra tính hợp lệ của dữ liệu chứ không thay thế được hoàn toàn công việc này.

-         Trigger hoạt động ngầm ở trong csdl, không hiển thị ở tầng giao diện. Do đó, khó chỉ ra được điều gì xảy ra ở tầng csdl.

-         Trigger thực hiện các update lên bảng dữ liệu vì thế nó làm gia tăng lượng công việc lên csdl và làm cho hệ thống chạy chậm lại

|  |  |
| --- | --- |
| **Trigger** | **Integrity Constraint** |
| Một trigger là một khối mã chứa 1 tập hợp các lệnh T-SQL sẽ được kích hoạt để đáp ứng một hành động ( action) nào đó.  Trigger không thể được tạo ra trên temporary hay system table.  Chỉ được kích hoạt sau khi đã insert, update, delete đã xảy ra  Trigger cho phép thông báo tùy chọn và quản lý lỗi phức tạp hơn | Tất cả các bộ của những quan hệ có liên quan trong cơ sở dữ liệu đều phải thỏa mãn ở bất kỳ thời điểm nào  Có thể tạo ra trên system table  Kiểm tra data trước khi cho phép nhập vào table  Các constraint có thể thông báo lỗi thông qua hệ thống báo lỗi tiêu chuẩn |

Câu 7. Cho lược đồ quan hệ dưới đây. Một nhân viên có thể làm viêc ờ nhiều phòng ban. Cột pct\_time trong quan hê Works chỉ tỉ lệ % thời gian nhân viên đó làm việc cho phòng ban tương ứng.

Viết các RBTV, assertions hoặc triggers để đảm bảo các yêu cầu sau:

1. Mỗi nhân viên phải có lương tối thiểu là 1000
2. Mọi người quản lý cũng là một nhân viên
3. Tổng tỉ lệ % thời gian làm việc cho các phòng ban của một nhân viên phải dưới 100%.
4. Một người quản lý phải luôn có lương cao hơn bất kỳ một nhân viên nào mà người đó quản lý.
5. Bất cứ khi nào một nhân viên được tăng lương, lương người quản lý cũng phải được tăng tương ứng.
6. Bất cứ khi nào một nhân viên được tăng lương, lương người quản lý cũng phải được tăng tương ứng. Hơn nữa, bất cứ khi nào một nhân viên được tăng lương, ngân sách của phòng ban tương ứng cũng phải được tăng lớn hơn tổng lương của tất cả nhân viên thuộc phòng đó.

R Câu 8. Cho CSDL quan hệ của một công ty như dưới đây.

Dùng SQL để xác định các views sau :

1. Một view có department name, manager name và manager salary của mọi phòng ban.

**Trả lời:**

Create view DeptManager as

select DEPARTMENT.Dname, EMPLOYEE.Fname as ManagerName, EMPLOYEE.Salary as ManagerSlary

from EMPLOYEE inner join DEPARTMENT on EMPLOYEE.Ssn = DEPARTMENT.Mgr\_ssn

1. Một view có employee name, supervisor name và employee salary của mỗi nhân viên thuộc phòng ‘Research’

**Trả lời:**

Create view ResearchDept as

select LowerGrade.Fname as EmpName, Superior.Fname as SuperName, LowerGrade.Salary as EmpSalary

from EMPLOYEE as LowerGrade inner join EMPLOYEE as Superior on LowerGrade.Super\_ssn = Superior.ssn

1. Một view có project name, controlling department name, number of employees và tổng số giờ được làm việc mỗi tuần của mỗi dự án.

**Trả lời:**

create View Project\_dept as

select T.PName as Project\_Name, T.DName as Controlling\_DepartmentName,Count(T1.ESSN) as NumberOfEmployee, SUM (T1.Hours) as HoursOfWeek

from (select PROJECT.PName,DEPARTMENT.DName,PROJECT.DNum,PROJECT.PNumber

from DEPARTMENT inner join PROJECT on DEPARTMENT.DNumber=PROJECT.DNum)as T inner join (select \*

from Works\_On) as T1 on T.PNumber=T1.Pno

group by T.DName,T.PName

1. Một view có project name, controlling department name, number of employees và tổng số giờ được làm việc mỗi tuần của mỗi dự án với những dự án có nhiều hơn một nhân viên làm việc cho dự án đó.

**Trả lời:**

create view Project\_Dept\_Morethan\_1 as

select L.Pname as Project\_Name, L.Dname as Controlling\_Dept\_Name, COUNT(P.Ssn) as Number\_of\_Emp, SUM(P.Hours) as Hours\_of\_week

from (select PROJECT.Pname, DEPARTMENT.Dname, PROJECT.Pnumber

from DEPARTMENT inner join PROJECT on DEPARTMENT.Dnumber = PROJECT.Dnum) as L inner join

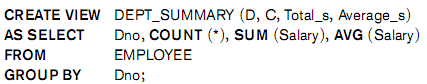
(select WORKS\_ON.Hours, EMPLOYEE.Ssn, WORKS\_ON.Pno

from WORKS\_ON inner join EMPLOYEE on WORKS\_ON.Essn = EMPLOYEE.Ssn) as P on L.Pnumber = P.Pno

group by L.Pname, L.Dname

having COUNT(P.Ssn)>1

Câu 9 Cho view DEPT\_SUMMARY được định nghĩa trên CSDL một công ty như sau.



Hãy cho biết các truy vấn và thao tác cập nhật nào sau đây được phép trên view. Nếu một query hoặc update nào được phép, hãy cho biết câu truy vấn hoặc cập nhật tương ứng được thực hiện trên các quan hệ cơ sở tương ứng với view.

