Chương 2.

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?

Trả lời: Đòi hỏi cột chỉ được phép chứa giá trị xuất hiện trong cột khoá chính của bảng khác

Các ràng buộc như vậy quan trọng vì

+ Tạo ra sự liên kết giữa các bảng

+ Đảm bảo việc toàn vẹn tham chiếu

+ Chống lại những hành động có thể làm sai dữ liệu

Tính toàn vẹn tham chiếu: Là 1 bộ giá trị trong một quan hệ tham chiếu tới một quan hệ tham chiếu tới một bộ giá trị đã tồn tại trong một quan hệ khác

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.

Trả lời : Các ràng buộc khoá ngoại là

* 1. Students (Sid) – Enrolled (Sid),
  2. Courses (Cid) – Enrolled (Cid),
  3. Faculty (Fid) – Teaches (Fid),
  4. Courses (Cid) – teaches (Cid),
  5. Courses (Cid) – Meet-in (Cid),
  6. Room (Rno) – Meet-in (Rno)

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.

Ràng buộc Check, GPA phải >=0 và <=10

Alter Table KQMH Add Constraint CheckGPA CHECK(Diem>=0 and Diem <=10)

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

CREATE DATABASE Exam3

USE Exam3

GO

CREATE TABLE Students(

sid NVARCHAR(10) PRIMARY KEY,

name NVARCHAR(50),

login NVARCHAR(50),

age INT,

gpa REAL

)

CREATE TABLE Faculty(

fid NVARCHAR(10) PRIMARY KEY,

fname NVARCHAR(50),

sal REAL

)

CREATE TABLE Courses(

cid NVARCHAR(10) PRIMARY KEY,

cname NVARCHAR(50),

credits INT

)

CREATE TABLE Rooms(

rno INT PRIMARY KEY,

address NVARCHAR(50),

capacity INT

)

CREATE TABLE Endrolled(

sid NVARCHAR(10),

cid NVARCHAR(10),

grade NVARCHAR(50),

CONSTRAINT PM\_Enroll PRIMARY KEY(sid,cid),

FOREIGN KEY (sid) REFERENCES dbo.Students(sid),

FOREIGN KEY (cid) REFERENCES dbo.Courses(cid)

)

CREATE TABLE Teaches(

fid NVARCHAR(10),

rno INT,

time NVARCHAR(10),

CONSTRAINT PM\_Teaches PRIMARY KEY(fid,rno),

FOREIGN KEY (fid) REFERENCES dbo.Faculty(fid),

FOREIGN KEY (rno) REFERENCES dbo.Rooms(rno)

)

CREATE TABLE Meets\_In(

cid NVARCHAR(10),

rno INT,

time NVARCHAR(10),

CONSTRAINT PM\_Meets\_In PRIMARY KEY(cid,rno),

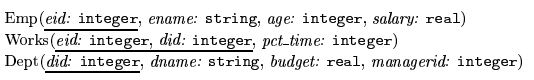
FOREIGN KEY (cid) REFERENCES dbo.Courses(cid),

FOREIGN KEY (rno) REFERENCES dbo.Rooms(rno)

)

go

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.

Trả lời: Mỗi bộ của 1 Dept sẽ có Id của 1 người quản lý (managerId), đó chính là khoá chính của quan hệ Emp (Eid). Trong trường hợp này, thì việc xoá một bộ trong quan hệ Dept không làm ảnh hưởng gì cả, vậy nên không có lựa chọn nào được đưa ra.

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

Trả lời:

create table Emp(

eid int not null primary key,

ename nvarchar(30),

age varchar(10),

salary varchar(10)

)

create table Works(

eid int not null,

did int not null,

constraint pk\_Bang primary key(eid,did),

pct\_time int,

)

create table Dept(

did int not null primary key,

dname nvarchar(30) not null,

budget real,

managerid int

)

alter table Works

add constraint fk\_eid

foreign key(eid)

references Emp(eid)

alter table Works

add constraint fk\_did

foreign key(did)

references Dept(did)

Alter table Dept

Add Constraint Fk\_ManagerId

foreign key (ManagerId)

references Emp(Eid)

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

Trả lời: ALTER TABLE Dept ALTER Column ManagerId INT NOT NULL;

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

Trả lời: insert into Emp values(101,'John Doe',32,15000)

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

Trả lời: update Emp set salary = salary\*1.1

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.

Trả lời:

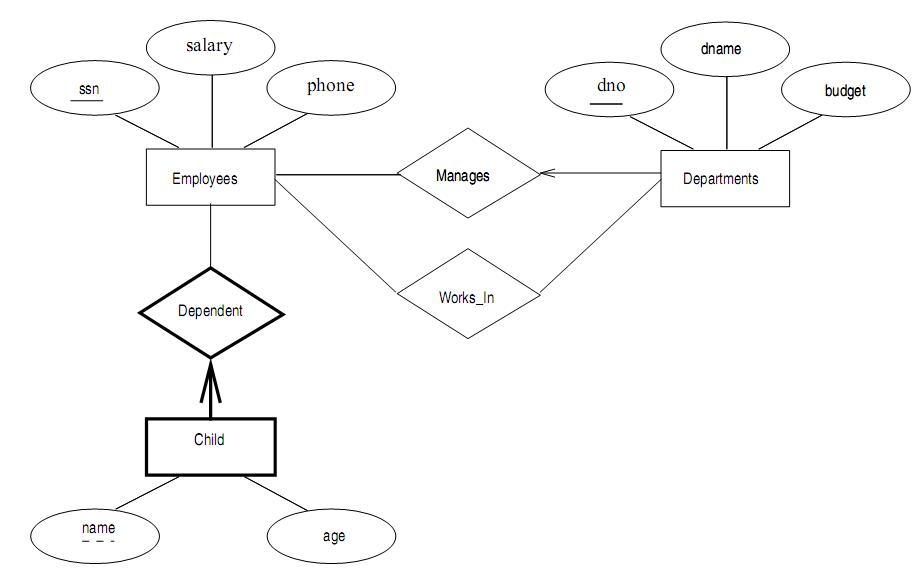
Delete

from Dept

where Dept.Did='Toy'

Câu lệnh này không thể được thực hiện, vì ở bảng Works thì Did là khoá chính, mà khoá chính thì không thể set NULL, vì vậy không thể xoá Department với Did=”Toy”

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.



Trả lời:

create table Employees(

SSN int PRIMARY KEY,

Salary real,

Phone nvarchar(13)

);

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)

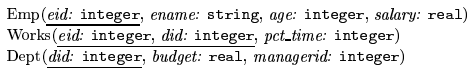
);

Database Diagram



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

Trả lời: alter table Emp add constraint chk\_salary 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.
2. 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.

Trả lời:

create assertion AgeOfManager

CHECK ((select Employee.Age

from Employee,Department

where Employee.Eid=Department.ManagerId)>30)

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.

Trả lời:

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.

|  |  |
| --- | --- |
| **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

Trả lời:

Alert Table Emp

Add Constraint CK\_minimumSalary

CHECK (Salary >= 1000)

1. Mọi người quản lý cũng là một nhân viên

Trả lời:

Create Trigger AddManager on Dept

After Update as

declare @managerid int

select @managerid=ol.managerid

from inserted ne,deleted ol

where ne.managerid =ol.managerid

If( 0 = ( select count(eid)

from Emp

where Emp.eid=@managerid as num))

Begin

Delete from Dept

Wh ere managerid=@old

end;

//cách 2

create function dbo.MyFunction(@ManagerId int)

returns int

as begin

DECLARE @kq int

select @kq=COUNT(\*)

from Dept,Emp

where Emp.Eid=@ManagerId

return @kq

end;

Alter table Dept

Add Constraint ManagerIsEmployee

CHECK (dbo.MyFunction(Dept.ManagerId)<>0)

1. Tổng tỉ lệ % thời gian làm việc cho các phòng ban của một nhân viên phải dưới 100%.

Trả lời:

create trigger TimeForWorks on Works

after update,insert

as

DECLARE @Eid int, @Did int

select @Eid=ne.Eid, @Did=ne.Did

from inserted ne

if ((select SUM(Works.Pct\_time)

from Works

where Works.Eid=@Eid)>100)

begin

raiserror('Tong Thoi Gian Lam Viec Cua Nhan Vien Phai Nho Hon 100',16,1)

Update Works

set Pct\_time='0'

where Eid=@Eid and Did=@Did

end

1. Một người quản lý phải luôn có lương cao hơn bất kỳ một nhân viên nào mà người đó quản lý.

Trả lời:

Create trigger Salary on Emp

After Update, Insert as

Declare @eid int, @salary int

select @eid=ne.eid, @salary=ne.salary

from inserted ne

If(@eid in (select managerid from Dept))

Begin

If(not((select count(salary)

from Emp, Dept, Works

where Emp.eid=Works.eid and Works.did=Dept.did and Emp.salary=@salary)=1 and

(select max(salary)

from Emp, Dept, Works

where Emp.eid=Works.eid and Works.did=Dept.did and managerid=@eid)=@salary))

Begin

raiserror('Lương của người quản lí phải cao hơn lương của nhân viên được quản lí',16,1)

End

End

Else

Begin

If(not((select max(salary)

from Emp, Dept, Works

where Emp.eid=Dept.ManagerId and Works.did=Dept.did and Emp.eid=@eid)=@salary))

Begin

raiserror('Lương của người quản lí phải cao hơn lương của nhân viên được quản lí',16,1)

End

End

1. Bất cứ khi nào một nhân viên được tăng lương, lương người quản lý cũng phải được tăng tương ứng.

Trả lời:

Create trigger UpdateSalary on Emp

After Update as

Declare @eid int, @salary int, @oldsalary int

select @eid=ne.eid, @salary=ne.salary, @oldsalary=ol.salary

from inserted ne, deleted ol

If(@salary>@oldsalary and @eid not in (select managerid from Dept))

Begin

Update Emp

Set salary=salary+(@salary-@oldsalary)

where eid in (select distinct managerid

from Dept,Works

where Works.eid=@eid and Dept.did=Works.did)

End

1. 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.

Trả lời:

Create trigger UpdateBudget on Emp

After Update as

Declare @eid int, @salary int, @oldsalary int

select @eid=ne.eid, @salary=ne.salary, @oldsalary=ol.salary

from inserted ne, deleted ol

If(@salary>@oldsalary and @eid not in (select managerid from Dept))

Begin

Update Emp

Set salary=salary+(@salary-@oldsalary)

where eid in (select distinct managerid

from Dept,Works

where Works.eid=@eid and Dept.did=Works.did)

While((select count(distinct Dept.did)

from Dept, (select did, sum(salary) as s\_salary

from Emp, Works

where Emp.eid=Works.eid Group By did) as A

where Dept.did=A.did and budget<=s\_salary)>0)

Begin

Update Dept

Set budget=budget+(@salary-@oldsalary)\*2

where did in (select Dept.did

from Dept, (select did, sum(salary) as s\_salary

from Emp, Works

where Emp.eid=Works.eid Group By did) as A

where Dept.did=A.did and budget<s\_salary)

End

End

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 NameAndSalary As

Select did, managerid, salary

From Dept, Emp

Where Dept.eid = Emp.eid;

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

Trả lời:

Create View Research As

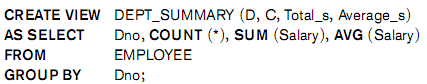
Select eid, managerid, salary

From Dept, Emp

Where Dept.eid = Emp.eid And Dept.dname = ’ Research’;

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.
2. 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.

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.

