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?

* Ràng buộc khóa ngoại ( ràng buộc tham chiếu ) là quy tắt, quy luật của DBMS về việc INSERT, DELETE, UPDATE dvb cxữ liệu giữa hai bảng có mối quan hệ ràng buộc khóa ngoại với nhau, để đảm bảo tính đúng đắn của dữ liệu.
* Ràng buộc khóa ngoại quan trọng vì ngăn cản chèn thêm dữ liệu không hợp lệ và giúp chúng ta quản lý, kiểm soát được các thao tác trên dữ liệu một cách chính xác nhất.
* Tính toàn vẹn tham chiếu là đảm bảo tính đúng đắn của dữ liệu về CSDL giữa 2 quan hệ tham chiếu với nhau.

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.

+ Sid : là khóa chính trong bảng Students nhưng lại xuất hiện đóng vai trò là khóa ngoại trong bảng Enrolled để kết nối hai bảng

+ Fid : là khóa chính trong bảng Faculty nhưng lại xuất hiện đóng vai trò là khóa ngoại trong bảng Teaches để kết nối 2 bảng

+ Cid : là khóa chính trong bảng Coures nhưng lại xuất hiện đóng vai trò là khóa ngoại trong bảng Teaches để kết nối 2 bảng

+ Rno : là khóa chính trong bảng Rooms nhưng lại xuất hiện đóng vai trò là khóa ngoại trong bảng Meets\_In để kết nối 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 : Ràng buộc thuộc tính sal trong bảng Faculty với sal>4000

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

create database student\_cau2

go

use student\_cau2

go

create table Students(

sid char(30) primary key,

name char(50),

login char(100),

age int,

gpa float

);

create table Faculty(

fid char(30) primary key,

fname nvarchar(50),

sal float

);

create table Courses(

cid char(30) primary key,

cname nvarchar(100),

credits int

);

create table Rooms (

rno int primary key,

address nvarchar(200),

capacity int

);

create table Enrolled(

sid char(30),

cid char(30),

primary key (sid, cid),

grade char(50),

foreign key (sid) references Students (sid),

foreign key (cid) references Courses (cid),

);

create table Teaches(

fid char(30),

cid char(30),

primary key (fid, cid),

foreign key (fid) references Faculty (fid),

foreign key (cid) references Courses (cid),

);

create table Meets\_In(

cid char(30),

rno int,

primary key (cid, rno),

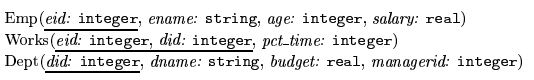
foreign key (cid) references Courses (cid),

foreign key (rno) references Rooms (rno),

time char(20)

);

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.

Mỗi bộ Dept sẽ có 1 ID của người quản lý, đó chính là khóa chính trong quan hệ trong bảng Emp. Vì thế khi ta xóa một bộ trong quan hệ Dept không cần phải tuân thủ ràng buộc và sẽ không làm ảnh hưởng gì cả.

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

create database Cau3\_Emp

go

use Cau3\_Emp

go

create table dbo.Emp(

eid int primary key,

ename nvarchar(250) not null,

age int not null,

salary float

);

create table dbo.Dept(

did int primary key,

dname nvarchar(250),

budget real,

managerid int,

);

create table dbo.Work (

eid int,

did int,

primary key (eid , did),

pct\_time int

foreign key (eid) references Emp (eid),

foreign key (did) references Dept (did)

);

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

alter table Dept add foreign key (managerid) references Emp (eid)

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

insert into Emp values (101, 'John Doe', 32, 15000);

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

update Emp

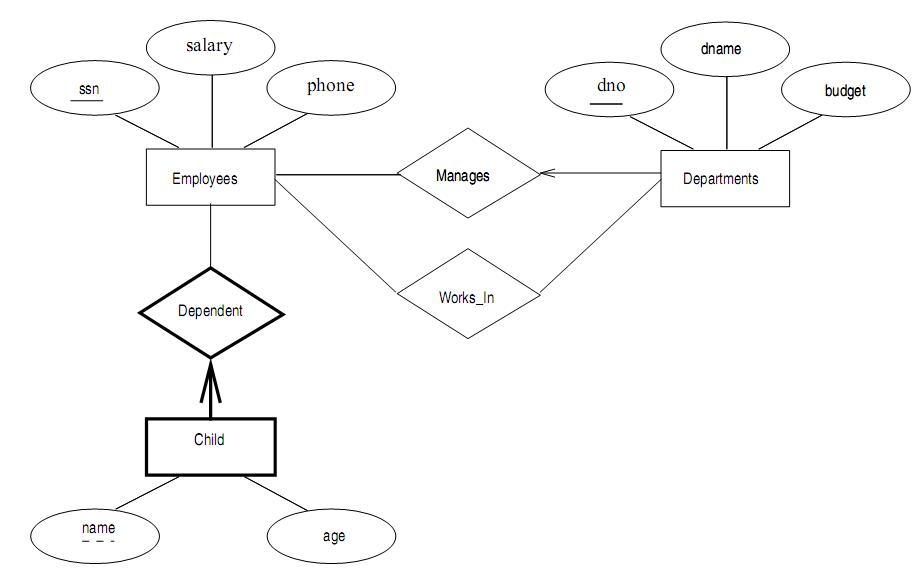
set Emp.salary = Emp.salary \* 1.1

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.

delete Dept

where dname = '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.



create database Bai4\_Employees

go

use Bai4\_Employees

go

create table dbo.Employees(

ssn int primary key,

phone int,

salary float

);

create table dbo.Deparments(

dno int primary key,

dname nvarchar(250),

budget char(50),

manages int,

foreign key (manages) references Employees (ssn)

);

create table dbo.Works\_in(

ssn int,

dno int,

primary key (ssn, dno),

foreign key (ssn) references Employees (ssn),

foreign key (dno) references Deparments (dno)

);

create table dbo.Child(

ssn int,

name nvarchar(250),

age int,

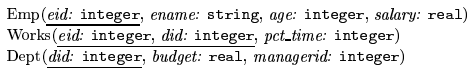
primary key(ssn, name),

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 constraint LuongToiThieu

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 MyFunction(@managerid int)

returns int

AS BEGIN

DECLARE @kq int

SELECT @kq= dbo.Emp.age

FROM Emp

WHERE Emp.eid=@managerid

RETURN @kq

END;

alter table Dept

Add Constraint CheckAge

CHECK (dbo.MyFunction(Dept.managerid)>30)

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 AgeOfManager on Dept

for insert, update

as

declare @managerid int

select @managerid = i.managerid from Inserted i;

if((select Emp.age

from Emp

where Emp.eid=@managerid)<30)

begin

raiserror ('Quản lý phải có tuổi lớn hơn 30',16,1);

end

go

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.

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

);

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.

|  |  |
| --- | --- |
| **Điểm mạnh Trigger** | **Điểm yếu Trigger** |
| + Kiểm tra tính toàn vẹn  + Được thực thi một cách tự động khi có câu lệnh update, insert, delete  + Có thể bắt lỗi logic nghiệp vụ ở mức csdl  + Rất hiệu quả khi được sử dụng để kiểm soát những thay đổi của dữ liệu trong bảng  + Có thể dùng trigger khác là một cách thay thế việc thực hiện những công việc hẹn giờ theo lịch | + 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. | 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 |
| Trigger không thể được tạo ra trên temporary hay system table. | Có thể tạo ra trên system table |
| Chỉ được kích hoạt sau khi đã insert, update, delete đã xảy ra | Kiểm tra data trước khi cho phép nhập vào table |
| Trigger cho phép thông báo tùy chọn và quản lý lỗi phức tạp hơn | 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

alter table Emp

add constraint minSalary

check (salary >= 1000)

1. Mọi người quản lý cũng là một nhân viên

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;

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

CREATE TRIGGER TimeForWorks ON Works

AFTER update,insert

AS

DECLARE @Eid INT, @Did INT

SELECT @Eid=ne.eid, @Did=ne.đi

FROM inserted ne

IF ((SELECT SUM(Works.Pct\_time)

FROM Works

WHERE [Works.eid=@Eid)>100](mailto:Works.eid=@Eid)%3e100))

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.
2. Bất cứ khi nào một nhân viên được tăng lương, lương người quản lý cũng phải được tăng tương ứng.
3. 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.

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.

CREATE VIEW NameAndSalary AS

SELECT did, managerid, salary

FROM Dept, Emp

WHERE Dept.did = Emp.eid;

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

CREATE VIEW Research AS

SELECT eid, managerid, salary

FROM Dept, Emp

WHERE Dept.did = 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.

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.

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

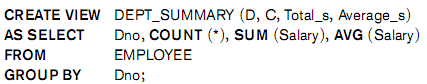
(select WORKS\_ON.Hours, EMPLOYEE.Ssn, WORKS\_ON.Pno

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.

