**Nhóm 1:**

* Đặng Thị Duyên – 16110299
* Phan Thị Thùy Dương – 16110303
* Huỳnh Lê Hữu Hưng – 16110347
* Phan Văn Phước Thịnh – 16110474
* Phạm Ngọc Diêu – 16110291
* Nguyễn Thanh Tân - 16110541

**BÀI TẬP CHƯƠNG 2**

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

* Ràng buộc khóa ngoại (ràng buộc tham chiếu) là quy tắc, 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.

Ví dụ: Có bảng A và bảng B là 2 bảng của DBMS, bảng A có khóa ngoại tham chiếu tới bảng B. Theo ràng buộc khóa ngoại thì bắt buộc phải nhập dữ liệu bên B, vì theo quy luật của ràng buộc khóa ngoại thì bên B phải có dữ liệu thì bên A mới tham chiếu tới được.

* Các ràng buộc quan trọng như vậy vì giúp ta quản lý, kiểm soát được các thao tác INSERT, DELETE, UPDATE dữ liệu của các bảng có quan hệ với nhau một cách chính xác nhất.
* Tính toàn vẹn tham chiếu là đảm bảo tính đúng đắn của dữ.

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 của bảng Student nhưng lại xuất hiện trong bảng Enrolled đóng vai trò là khóa ngoại kết nối hai bảng.
* Fid: là khóa chính của bảng Faculty nhưng lại xuất hiện trong bảng Teacher đóng vai trò là khóa ngoại kết nối hai bảng.
* Cid: là khóa chính của bảng Courses nhưng lại xuất hiện trong bảng Teacher đóng vai trò là khóa ngoại kết nối hai bảng.
* Rno: là khóa chính của bảng Rooms nhưng lại xuất hiện trong bảng Meets\_In đóng vai trò là khóa ngoại kết nối hai bảng.

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

Ràng buộc thuộc tính sal trong bảng Facculty với sal>4000.

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

create table Students(

sid NVARCHAR(10) PRIMARY KEY,

name NVARCHAR(max),

login NVARCHAR(50),

age INT,

gpa FLOAT

)

-----------------------------------

create table Facutly(

fid NVARCHAR(10) PRIMARY KEY,

fname NVARCHAR(max),

sal FLOAT,

CHECK(sal>4000)

)

------------------------------------

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 Enrolled(

sid NVARCHAR(10),

cid NVARCHAR(10),

grade NVARCHAR(10),

PRIMARY KEY(sid,cid),

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

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

)

--------------------------

create table Teaches(

fid NVARCHAR(10),

cid NVARCHAR(10),

PRIMARY KEY(fid,cid),

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

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

)

-------------------------

create table Meets\_In(

cid NVARCHAR(10),

rno INT,

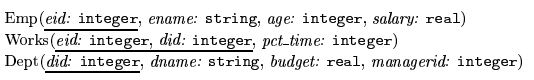
PRIMARY KEY(cid,rno),

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

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

)

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



Hãy trả lời các câu hỏi:

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 thue ràng buộc này khu 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 của quan hệ Emp. Trong trường hợp này, việc xóa 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

CREATE TABLE Emp(

eid INT NOT NULL PRIMARY KEY,

ename NVARCHAR(50),

age INT,

salary FLOAT

)

--------------------------

CREATE TABLE Dept(

did INT NOT NULL PRIMARY KEY,

dname NVARCHAR(50),

budget FLOAT,

managerid INT,

FOREIGN KEY(managerid) REFERENCES dbo.Emp(eid)

)

--------------------------

CREATE TABLE Works(

eid INT NOT NULL,

did INT NOT NULL,

pct\_time INT,

PRIMARY KEY(eid, did),

FOREIGN KEY(eid) REFERENCES dbo.Emp(eid),

FOREIGN KEY(did) REFERENCES dbo.Dept(did)

)

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

CREATE TABLE Dept(

did INT NOT NULL PRIMARY KEY,

dname NVARCHAR(50),

budget FLOAT,

managerid INT NOT NULL,

FOREIGN KEY(managerid) REFERENCES dbo.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 dbo.Emp (eid, ename, age, salary)

VALUES (101, -- eid - int

N'John Doe', -- ename – nvarchar (50)

32, -- age - int

15000 -- salary - float

)

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

UPDATE dbo.Emp

SET salary=1.1\*salary

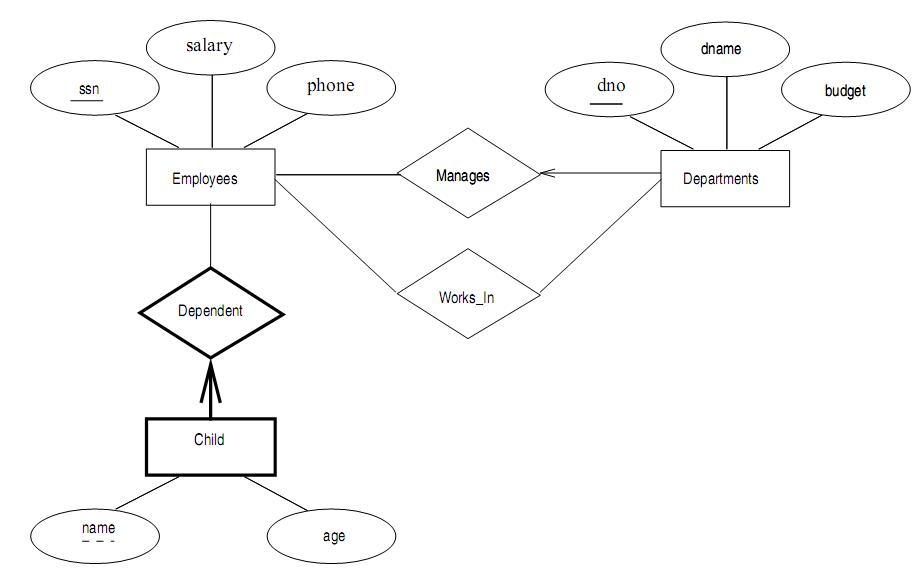
1. Viết câu lệnh SQL để xóa department ‘Toy’. Với ràng buộc toàn vẹn tham chiếu bạn đã chọn cho lược đồ này, hãy giải thích điều gì xảy ra khi câu lệnh xóa này được thực hiện.

DELETE FROM dbo.Dept

WHERE dname='Toy'

Với ràng buộc toàn vẹn tham chiếu đã chọn cho lượt đồ này, thì khi viết lệnh xóa thông thường sẽ báo lỗi do bọ Contrainst ràng buộc. Để xóa thành công được department ‘Toy’ thì phải dùng lệnh: ON DELETE SET NULL trong lúc tạo bảng bằng code để tạo khả năng xóa cho bảng

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 TABLE Employees(

ssn INT PRIMARY KEY,

salary FLOAT,

phone NCHAR(11)

)

--------------------------

CREATE TABLE Child(

ssn INT,

name NVARCHAR(50),

age INT,

DependentID INT,

PRIMARY KEY(ssn, name),

FOREIGN KEY(DependentID) REFERENCES dbo.Employees(ssn)

)

--------------------------

CREATE TABLE Departments(

dno INT PRIMARY KEY,

dname NVARCHAR(50),

budget FLOAT,

managerID INT,

FOREIGN KEY(managerID) REFERENCES dbo.Employees(ssn)

)

--------------------------

CREATE TABLE Work\_In(

ssn INT,

dno INT,

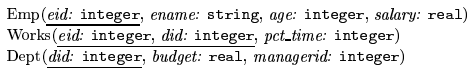
PRIMARY KEY(ssn, dno),

FOREIGN KEY(ssn) REFERENCES dbo.Employees(ssn),

FOREIGN KEY(dno) REFERENCES dbo.Departments(dno)

)

Câu 5: Cho lược đồ quan hệ sau:



Hãy trả lời các câu hỏi sau:

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

create table Emp(

eid INT PRIMARY KEY,

ename NVARCHAR(50),

age INT,

salary FLOAT,

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 dbo.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.
2. 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,Works, (select Emp.eid, Emp.salary, Dept.did

from Emp, Dept

where Emp.eid=Dept.managerid) as Manager

where Emp.eid=Works.eid and Works.did=Manager.did and Emp.salary>Manager.salary

)

Nếu đã thiết lập khóa ngoại cho bảng Works hoặc Dept thì không thể xóa

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 :

+ Sử dụng Trigger để kiểm tra tính toàn vẹn của cơ sở dữ liệu.

+ Trigger có thể bắt lỗi logic ở mức cơ sở dữ liệu.

+ 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 sử dụng để kiểm soát những thay đổi của dữ liệu trong bảng.

* Điểm yếu :

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

+ Trigger hoạt động ngầm 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 tăng lượng công việc lên csdl và làm cho hệ thống chạy chậm

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 CK\_minimumSalary

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;

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%.

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](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.

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 nhânviên 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.

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.

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.

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’.

CREATE VIEW Research AS

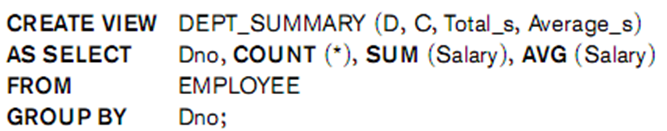
SELECT eid, managerid, salary

FROM Dept, Emp

WHERE Dept.didid = 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.

