**Bài tập chương 2**

**Hoàng Công Thành Nhân - 20110687**

**Nguyễn Kiều Châu Anh – 20110234**

**Lâm Hoàng Duyên – 20110174**

**Phan Thị Diễm Trinh - 20110193**

# Câu hỏi ôn tập Chương 2

1. Ràng buộc toàn vẹn là gì?

Là tập các quy tắc mà mọi dữ liệu trong CSDL phải tuân theo.

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

Để đảm bảo sự toàn vẹn của CSDL.

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

Là khóa ứng viên. Đóng vai trò như id của row. Mỗi row có một giá trị đặc biệt (unique value) với cột khóa chính.

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

Là attribute của 1 quan hệ tham chiếu tới khóa (là unique hoặc primary) trong một quan hệ khác. Dùng để xác lập mối quan hệ giữa các table

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

Là giới hạn các giá trị mà 1 thuộc tính có thể nhận.

1. Một thuộc tính có ràng buộc Unique?

Là mỗi giá trị chỉ được xuất hiện duy nhất 1 lần trong thuộc tính đó của bảng.

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

Là không có bất cứ giá trị null nào được xuất hiện trong thuộc tính đó của bảng.

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

Ràng buộc CHECK được sử dụng để giới hạn phạm vi giá trị có thể đặt vào trong thuộc tính.

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

Ta giả sử quan hệ Q đã được tạo trước và ta đang cần:

Thêm rang buộc khóa chính vào thuộc tính a1 của Q

ALTER TABLE Q ADD CONSTRAINT TenRangBuoc PRIMARY KEY (a1)

Xóa rang buộc khóa chính khỏi thuộc tính a1 của Q mới them vào

ALTER TABLE Q DROP CONSTRAINT TenRangBuoc

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 hành động được thực hiện khi một sự kiện (Data modification event - Insert, Update, Delete) xảy ra.

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

* Biến cố trong CSDL thường là: chèn, cập nhật, xóa dữ liệu trong bảng
* Điều kiện: khi biến cố xảy ra, thì trigger sẽ kiểm tra xem điều kiện có đúng hay không
* Hành động: nếu điều kiện đúng thì các hành động (lệnh) trong trigger sẽ được thực thi

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

Trigger time: là thời hạn kích hoạt các action thường gắn với lệnh Before hoặc After. Dùng Before nếu muốn xử lý hành động đó trước khi thay đổi dữ liệu trên bảng. Dùng After nếu cần xử lý hành động đó sau khi đã thay đổi trên bảng xong.

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 trigger tên AA trên quan hệ Q. Nhiệm vụ của trigger này là sau khi xóa một tuple từ Q thì kiểm tra xem có tuple nào có B > 2 hay không. Nếu có thì in ra thông báo “Notify Customer Relations”

1. Instead of trigger là gì?

Trigger INSTEAD OF là một trigger cho phép bạn thay thế một câu lệnh INSERT, DELETE hoặc UPDATE một bảng hoặc một view bằng cách thực thi câu lệnh khác được định nghĩa trong trigger.

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 ra ràng buộc có tên AA trong bảng Q sau khi thực hiện xóa trigger tự hoạt động tạo 2 biến @new, @old kiểm tra điều kiện nếu @old >2 thì thực hiện in thông báo 'Notify Customer Relations'. Trong quá trình hoạt động nếu có xảy ra lỗi Rollback sẽ trở lại hoạt động trước đó.

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?

Trigger chỉ được gọi sau khi thay đổi (delete) xảy ra trên Q.

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?

Từ khóa for xác định event nào kích hoạt trigger AA trong quan hệ Q. Ở đây là Delete.

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 trigger tên BB trên quan hệ Q1. Nhiệm vụ của trigger này là khi có lệnh thêm một tuple vào Q thì thay vào đó, ta sẽ thêm một tuple vào Q2 với các giá trị (newA = n.A, newB = n.B).

1. View là gì?

View là một bảng ảo có nội dung là các hàng và cột được định nghĩa bởi các query lấy data từ 1 hoặc nhiều bảng base hoặc từ các view khác trong database.

1. Mục đích của sử dụng view
2. Tập trung, đơn giản hóa và điều chỉnh góc nhìn phù hợp với từng user tới database.
3. Là một cơ chế bảo mật khi chỉ cho phép user truy cập tới data thông qua view mà không cho phép họ truy cập trực tiếp vào base table.
4. Để cung cấp 1 interface tương thích ngược để giả lập 1 quan hệ có schema đã được thay đổi.
5. Trong SQL server, cú pháp câu lệnh tạo, xóa, sửa view?

Tạo: **CREATE VIEW ten\_view AS SELECT ten\_cot(s) FROM ten\_bang WHERE dieu\_kien**

Xóa: **DROP VIEW ten\_view**

Chỉnh sửa: **ALTER VIEW ten\_view AS SELECT ten\_cot FROM ten\_bang WHERE dieu\_kien**

1. Phân loại View?

Có thể có 3 loại View cơ bản :

* Standard View: View được tạo bao gồm các cột là các bảng hoặc các View khác.
* Indexed View : View được tạo và được đặt chỉ mục Unique Clustered Index.
* Paritioned View : View được tạo bao gồm các dữ liệu các phân cụm ngang từ một hoặc nhiều bảng.

1. Ý nghĩa của câu lệnh

Create view cc as Select A, max(B) as ln from Q1 group by A;

Tạo view tên cc với các A nào có B là lớn nhất (đặt tên là ln) từ bảng Q1 gom nhóm bằng 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 TruongPhong NOT NULL (MaNQL)

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 mỗi NV đều có lương 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

# Bài tập 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?

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

Các ràng buộc quan trọng là vì cơ sở dữ liệu cần có chúng để đảm bảo tính toàn vẹn, nhất quán.

Khái niệm về tính toàn vẹn tham chiếu nói rằng bạn không thể thêm bản ghi bào bảng chứa trừ khi có một bản ghi tương ứng trong bảng được tham chiếu. Khi nói đến toàn vẹn tham chiếu, ta cũng thường nhắc tới các khái niệm cập nhật tầng (CASCADE) và xóa tầng nhằm đảm bảo các thực hiện cho bảng được tham chiếu được thể hiện trong bảng tham chiếu (bảng khóa ngoại).

**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 ngoại ở bảng Enrolled tham chiếu tới **sid** là khóa chính ở bảng Students.
* **cid** là khóa ngoại ở bảng Enrolled, bảng Teaches, bảng Meets\_In tham chiếu tới kháo chính **cid** ở bảng Courses
* **fid** là khóa ngoại ở bảng Teaches tham chiếu tới khóa chính **fid** ở bảng Faculty
* **rno** là khóa ngoại ở bảng Meets\_In tham chiếu tới khóa chính **rno** ở bảng Rooms

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.

* Thuộc tính address của Rooms cần phải có ràng buộc unique
* Ràng buộc check về GPA max và min mà một sinh viên có thể nhận được
* Ràng buộc check về cấp lớp max và min mà học sinh có thể học.

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

CREATE DATABASE QLSV

go

USE QLSV

go

CREATE TABLE STUDENTS(

s\_id varchar(20) not null primary key,

s\_name nvarchar(50),

s\_login varchar(50),

s\_age int,

s\_gpa real

);

go

CREATE TABLE FACULTY(

f\_id varchar(20) not null primary key,

f\_name nvarchar(50),

sal real

);

go

CREATE TABLE COURSES(

c\_id varchar(20) not null primary key,

c\_name nvarchar(50),

credits int

);

go

CREATE TABLE ROOMS(

r\_no varchar(20) not null primary key,

address nvarchar(50),

capacity int check (capacity between 50 and 200)

);

go

CREATE TABLE ENROLLED(

s\_id varchar(20) not null references STUDENTS(s\_id)

ON UPDATE CASCADE,

c\_id varchar(20) not null references COURSES(c\_id)

ON UPDATE CASCADE,

grade varchar(20) check (grade between 0 and 10)

primary key(s\_id,c\_id)

);

go

CREATE TABLE TEACHES(

f\_id varchar(20) not null references FACULTY(f\_id) ON UPDATE CASCADE,

c\_id varchar(20) not null references COURSES(c\_id) ON UPDATE CASCADE,

primary key(f\_id,c\_id)

);

go

CREATE TABLE MEETS\_IN(

c\_id varchar(20) not null references COURSES(c\_id) ON UPDATE CASCADE,

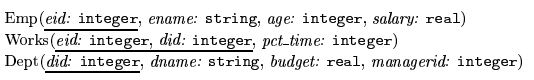
r\_no varchar(20) not null references ROOMS(r\_no) ON UPDATE CASCADE,

metting\_time timestamp,

primary key(c\_id,r\_no,metting\_time)

);

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.

Ta có thể có trường did trong quan hệ Works là khóa ngoại tham chiếu đến trường did trong quan hệ Dept.

Khi người dùng xóa một tuple trong Dept, chúng ta có 4 lựa chọn :

* Xóa các tuple trong Works có tham chiếu tới nó.
* Không cho phép xóa các tuple trong Dept nếu như có tuple nào đó trong Works tham chiếu tới nó.
* Với mỗi tuple Works có tham chiếu tới nó thì ta set trường did của nó thành giá trị mặc định nào đó như ‘default’.
* Với mỗi tuple Works có tham chiếu tới nó thì ta set trường did của nó thành giá trị null.

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 INTEGER,

Ename CHAR(10),

Age INTEGER,

Salary REAL,

PRIMARY KEY (eid))

CREATE TABLE Works (

Eid INTEGER,

Did INTEGER,

Pcttime INTEGER,

PRIMARY KEY (eid, did),

FOREIGN KEY (did) REFERENCES Dept(did),

FOREIGN KEY (eid) REFERENCES Emp(eid)

ON DELETE CASCADE)

CREATE TABLE Dept (

Did INTEGER,

Budget REAL,

ManagerId INTEGER,

PRIMARY KEY (did),

FOREIGN KEY (managerid) references Emp(eid)

ON DELETE SET NULL)

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 INTEGER,

Budget REAL,

ManagerId INTEGER **NOT NULL**,

PRIMARY KEY (did),

FOREIGN KEY (managerid) references Emp(eid)

ON DELETE SET NULL)

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

INSERT INTO Emp (eid, ename, age, salary) VALUES (101, ‘John Doe’, 32, 15000)

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

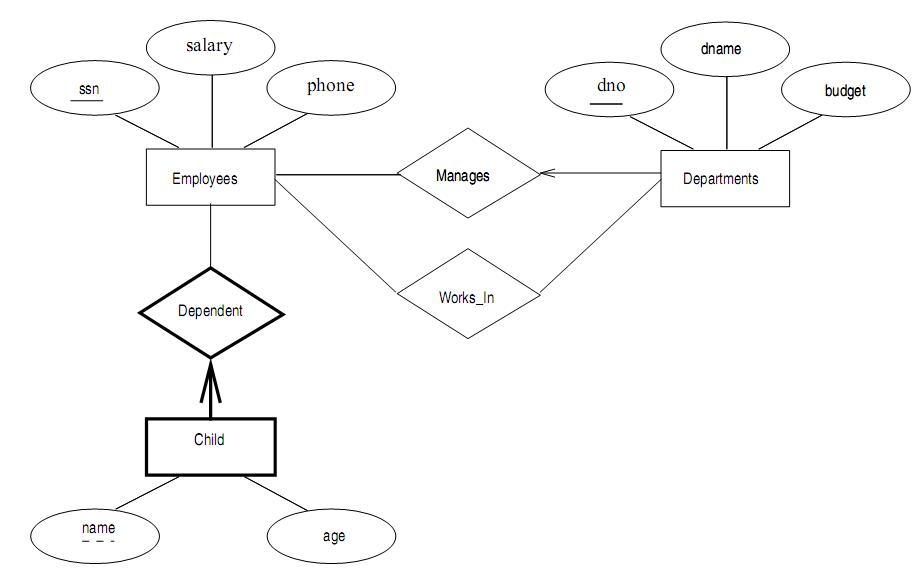
UPDATE Emp SET Emp.salary = Emp.salary \* 1.1

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 FROM Dept WHERE Dept.name = ‘Toy’

Ta thấy trường did trong Works là khóa ngoại tham chiếu tới quan hệ Dept. Chúng ta đã set ON DELETE CASCADE vào Works nên khi một phòng ban bị xóa, những tuple trong Works có tham chiếu đến phòng ban đó sẽ bị xóa theo.

**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 OFFICE

GO

USE OFFICE;

CREATE TABLE EMPLOYEES(

em\_ID int primary key,

salary real check(salary>1000000),

phone varchar(12) check(len(phone) between 6 and 12),

);

go

CREATE TABLE Child(

em\_ID int,

child\_name nvarchar(50),

age int,

foreign key(em\_ID) references EMPLOYEES(em\_ID)

on update cascade,

primary key(child\_name,em\_ID)

);

go

CREATE TABLE Departments(

dno varchar(20) primary key,

dname nvarchar(50),

budget bigint

);

go

ALTER TABLE EMPLOYEES

add department varchar(20) references Departments(dno);

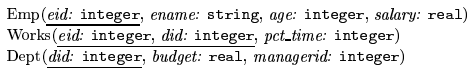
go

ALTER TABLE Departments

add manager int references EMPLOYEES(em\_ID)

**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

create table Emp (

eid int Primary key,

ename varchar(50),

age int,

salary real,

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 TABLE Dept (did INTEGER, buget REAL, managerid INTEGER, PRIMARY KEY (did), FOREIGN KEY (managerid) REFERENCES Emp, CHECK((SELECT E.age FROM Emp E, Dept D WHERE E.eid=D.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.

*Lựa chọn 1:*

CREATE TABLE Dept (did INTEGER, buget REAL, managerid INTEGER, PRIMARY KEY (did), FOREIGN KEY (managerid) REFERENCES Emp, CHECK((SELECT E.age FROM Emp E, Dept D WHERE E.eid=D.managerid)>=30))

*Lựa chọn 2:*

CREATE ASSERTION managerAge CHECK ((SELECT E.age FROM Emp E, Dept D WHERE E.eid = D.managerid ) >= 30 )

Nó kiểm tra khả năng vi phạm xác nhận bất cứ khi nào một trong các quan hệ được cập nhật -> tốt hơn.

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.

*Tạo view E\_Salary:*

Create view E\_Salary as

SELECT Emp.eid, salary, managerid

FROM Emp, Works, Dept

WHERE Emp.eid=Works.eid AND Works.did=Dept.did

*Tạo view Salary:*

Create view Salary as

SELECT E\_Salary.eid, E\_Salary.salary as eSalary, managerid as mid, Emp.salary as mSalary

FROM E\_Salary, Emp

WHERE E\_Salary.managerid = Emp.eid

*Xóa nhân viên có lương cao hơn quản lý*

Delete From Emp

Where EXISTS (Select \* from Salary where Salary.eSalary > Salary.mSalary and Salary.eid = Emp.eid)

**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 là một thủ tục được tự động gọi để phản hồi lại 1 sự thay đổi tới database.

Điểm mạnh của trigger là cho phép chúng ta thực hiện chuỗi hành động dựa vào kết quả của câu query. Những câu query như update, delete, insert, tạo table/view mới hay alter đều có thể kích hoạt trigger. Trigger có thể được cài đặt để thực thi trước (BEFORE) hoặc sau (AFTER) thay đổi được thực hiện lên database.

Điểm yếu của trigger là làm giảm hiệu năng khi truy xuất database vì làm tang số lượng công việc mà tầng csdl phải thực hiện. Hoạt động ngầm ở tầng database chứ không hiển thị ở tầng giao diện nên khó có thể chỉ ra được điều gì đã xảy ra ở tầng giao diện. Không thể thay thế hoàn toàn rang buộc toàn vẹn để kiểm tra tính toàn vẹn của dữ liệu. Và ràng buộc toàn vẹn thường dễ hiểu hơn dùng trigger.

**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

CREATE TABLE Emp (id INTEGER,

name CHAR(10),

age INTEGER,

salary REAL,

PRIMARY KEY (eid),

CHECK (salary >= 10000)

)

1. Mọi người quản lý cũng là một nhân viên

CREATE ASSERTION ManagerIsEmployee

CHECK ( ( SELECT COUNT(\*)

FROM Dept D

WHERE D.managerid NOT IN ( SELECT eid FROM Emp))=0)

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

CREATE TABLE Works ( eid INTEGER0, did INTEGER, pct time INTEGER, PRIMARY KEY (eid, did), CHECK ( (SELECT COUNT (W.eid) FROM Works W GROUP BY W.eid HAVING Sum(pct time) > 100) = 0))

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

CREATE ASSERTION QuanLyLuongCaoHonNV

CHECK (

SELECT E.ei

FROM Emp E, Emp M, Works W, Dept D

WHERE E.eid = W.eid //

AND W.did = D.did // *Chọn ra những người làm cùng phòng với nhau*

AND D.managerid = M.eid // *Chọn ra quản lý trong phòng đó*

AND E.salary > M.salary) // *Kiểm tra xem có nhân viên nào cao lương hơn quản lý không*

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

CREATE TRIGGER GiveRaise AFTER UPDATE ON Emp

WHEN old.salary < new.salary

FOR EACH ROW

BEGIN

UPDATE Emp M

SET M.salary = new.salary

WHERE M.salary < new.salary *// Nếu lương quản lý thấp hơn lương mới của nhân viên sau tăng*

AND M.eid IN (SELECT D.managerid *// Chọn ra tất cả các quản lý của tất cả các phòng ban*

FROM Emp E, Words W, Dept D

WHERE E.eid = new.eid

AND E.eid = W.eid

AND W.did = D.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 GiveRaise AFTER UPDATE ON Emp

WHEN old.salary < new.salary

FOR EACH ROW

DECLARE \_raise REAL;

BEGIN

\_raise := new.salary – old.salary;

UPDATE Emp M

SET M.salary = new.salary

WHERE M.salary < new.salary

AND M.eid IN (SELECT D.managerid

FROM Emp E, Words W, Dept D

WHERE E.eid = new.eid

AND E.eid = W.eid

AND W.did = D.did);

UPDATE Dept D

SET D.budget = D.budget + \_raise

WHERE D.did IN (SELECT W.did

FROM Emp E, Works W, Dept D

WHERE E.eid = new.eid

AND E.eid = W.eid

AND D.did = W.did

AND D.budget <

(SELECT Sum(E2.salary)

FROM Emp E2, Works W2

WHERE E2.eid = W2.eid

AND W2.dept = D.did));

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 V1 AS

SELECT Dname, Fname, Lname, Salary

FROM Department, Employee

WHERE D. Mgrssn=E.Ssn

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 V3 AS

SELECT Pname, Dname, (SELECT COUNT(Ssn) FROM WORKS\_ON W1 WHERE W1.Pno=P.Pno) AS Num\_Employees,

(SELECT SUM(Hours) FROM WORKS\_ON W2 WHERE W2.Pno=P.Pno) AS Total\_Hours

FROM Project P, Department D

WHERE P.Dno=D.Dno

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 V4 AS

SELECT L.Pname AS Project\_Name, L.Dname AS ControllingDeptName, COUNT(P.Ssn) AS NumberOfEmp, SUM(P.Hours) AS HoursOfWeek

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

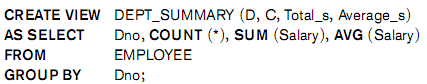
FROM WORKS\_ON INNER JOIN EMPLOYEE

ON WORKS\_ON.Essn = EMPLOYEE.Ssn) AS P

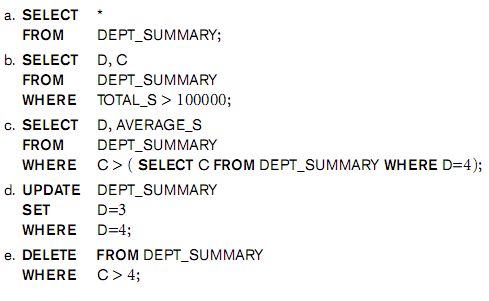
ON L.Pnumber = P.PnoGroup by L.Pname, L.Dname

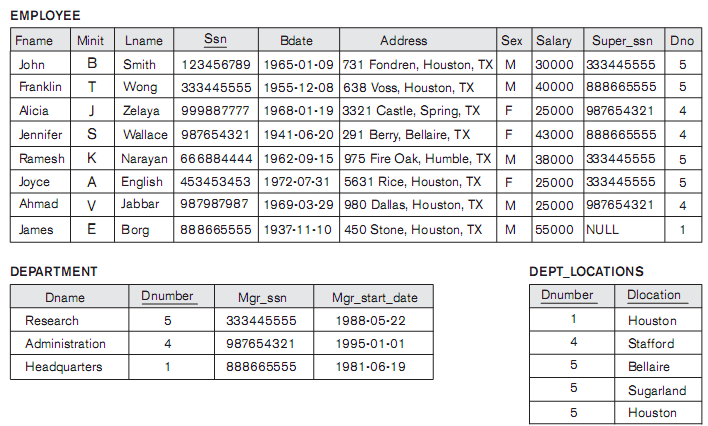
Having COUNT(P.Ssn)>1

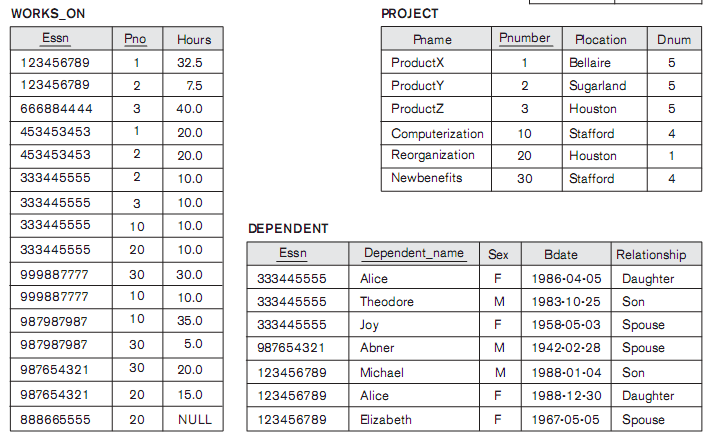
**Câu 9. Cho view DEPT\_SUMMARY được định nghĩa trên CSDL một công ty như sau.**



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.

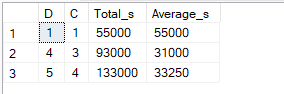






**a. SELECT \* FROM DEPT\_SUMMARY;**

**--> Được truy vấn và thao tác cập nhật nào sau đây được phép trên view**



**Hàm thay thế**

Select Dno as D,Count(\*) as C,sum(Salary) as Total\_s,AVG(Salary) as Average\_s

from EMPLOYEE Group by Dno;

**b. SELECT D,C FROM DEPT\_SUMMARY WHERE TOTAL\_S > 100000;**



-->Được truy vấn và thao tác cập nhật nào sau đây được phép trên view

**Hàm thay thế**

Select Dno as D,Count(\*) as C

from EMPLOYEE Group by Dno having sum(Salary)>100000;

**c. SELECT D, AVERAGE\_S**

FROM DEPT\_SUMMARY

WHERE C > ( SELECT C FROM DEPT\_SUMMARY WHERE D=4);



--> Được truy vấn và thao tác cập nhật nào sau đây được phép trên view

***Hàm thay thế***

Select Dno as D,AVG(Salary) as Average\_s

from EMPLOYEE Group by Dno having Count(\*)>(select Count(\*) from EMPLOYEE where Dno=4) ;

**d. UPDATE DEPT\_SUMMARY SET D=3 WHERE D=4;**

Cannot update the view or function 'DEPT\_SUMMARY' because it contains aggregates, or a DISTINCT or GROUP BY clause, or PIVOT or UNPIVOT operator.

--> Câu truy vấn trên đây được phép trên view

**e. DELETE FROM DEPT\_SUMMARY WHERE C > 4;**

Cannot update the view or function 'DEPT\_SUMMARY' because it contains aggregates, or a DISTINCT or GROUP BY clause, or PIVOT or UNPIVOT operator.

--> Câu truy vấn trên đây được phép trên view