**MỤC LỤC**

Contents

[LỜI MỞ ĐẦU 3](#_Toc530306849)

[CHƯƠNG 1: CÁC KHÁI NIỆM CƠ BẢN TRONG CƠ SỞ DỮ LIỆU QUAN HỆ 4](#_Toc530306850)

[1.1. Tổng quan về cơ sở dữ liệu 4](#_Toc530306851)

[1.1.1. Khái niệm cơ bản về cơ sở dữ liệu 4](#_Toc530306852)

[1.1.2. Kiến trúc một hệ CSDL 5](#_Toc530306853)

[1.1.3. Lược đồ khái niệm và mô hình dữ liệu 6](#_Toc530306854)

[1.2. Mô hình cơ sở dữ liệu quan hệ 6](#_Toc530306855)

[1.2.1. Các khái niệm cơ bản 6](#_Toc530306856)

[1.2.2. Khái niệm khóa (KEY) 8](#_Toc530306857)

[1.2.4. Các mô hình CSDL 11](#_Toc530306858)

[CHƯƠNG 2. NGÔN NGỮ THAO TÁC DỮ LIỆU 20](#_Toc530306859)

[2.1 Đại số quan hệ 20](#_Toc530306860)

[2.1.1 Phép chiếu 20](#_Toc530306861)

[2.1.2 . Phép chọn 20](#_Toc530306862)

[2.1.3. Phép kết nối 21](#_Toc530306863)

[2.1.4 Phép tập hợp 24](#_Toc530306864)

[2.2 Ngôn ngữ tân từ SQL (Structured query language) 28](#_Toc530306865)

[2.2.1 Tạo, cập nhật, xóa dữ liệu 29](#_Toc530306866)

[2.2.2 Truy Vấn 32](#_Toc530306867)

[CHƯƠNG 3: RÀNG BUỘC TOÀN VẸN DỮ LIỆU 41](#_Toc530306868)

[3.1. Khái niệm ràng buộc toàn vẹn 41](#_Toc530306869)

[3.2. Các yếu tố của ràng buộc toàn vẹn 41](#_Toc530306870)

[3.2.1. Điều kiện của RBTV 41](#_Toc530306871)

[3.2.2. Bối cảnh RBTV 42](#_Toc530306872)

[3.2.3. Tầm ảnh hưởng RBTV 42](#_Toc530306873)

[3.2.4. Hành động khi RBTV bị vi phạm 43](#_Toc530306874)

[3.3. Phân loại ràng buộc toàn vẹn 43](#_Toc530306875)

[CHƯƠNG 4. CHUẨN HÓA VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 50](#_Toc530306876)

[4.1 Phụ thuộc hàm 50](#_Toc530306877)

[4.2 Các mệnh đề và tiên đề Amstrong 51](#_Toc530306878)

[4.3 Bao đóng (CLOSURE) 51](#_Toc530306879)

[4.4. Sự tương đương của phụ thuộc hàm 54](#_Toc530306880)

[4.5. Phụ thuộc hàm tối thiếu 55](#_Toc530306881)

[4.6 Khóa và giải thuật tìm khóa 57](#_Toc530306882)

[4.7. Chuẩn hóa lược đồ quan hệ 59](#_Toc530306883)

[4.8 Kiểm tra phép tách quan hệ không làm mất thông tin 63](#_Toc530306884)

# LỜI MỞ ĐẦU

Cơ sở dữ liệu đóng vai trò quan trọng trong các ngành học về công nghệ thông tin. Nghiên cứu về cơ sở dữ liệu là nghiên cứu về cách tổ chức lưu trữ, quản lí, khai thác và xử lí thông tin trên cơ sở hiểu rõ về các kiểu dữ liệu, cấu trúc dữ liệu và ý nghĩa của dữ liệu. Trong bốn mươi năm qua đã có rất nhiều nhà khoa học nghiên cứu về cơ sở dữ liệu và đưa cơ sở dữ liệu vào trong các chương trình đào tạo.

Cơ sở dữ liệu đã trở thành một môn học cơ sở của nhiều chuyên ngành như khoa học máy tính, các hệ thống thông tin, công nghệ phần mềm và cả các ngành khoa học, kinh tế khác. Người ta có thể tiếp cận cơ sở dữ liệu từ cơ bản đến nâng cao. Giáo trình cơ sở dữ liệu này trình bày những khái niệm cơ bản cơ sở dữ liệu và một số kĩ thuật cơ bản để xây dựng các cơ sở dữ liệu truyền thống.

Với những kiến thức đó sinh viên có thể lập trình các ứng dụng để sử dụng nó trong cuộc sống hàng ngày và lấy đó làm cơ sở để nghiên cứu sâu hơn với các hệ cơ sở dữ liệu nâng cao, tạo ra các cơ sở dữ liệu đáp ứng các yêu cầu ứng dụng hiện đại.

Giáo trình cơ sở dữ liệu tập trung trình bày những vấn đề cốt lõi nhất của môn cơ sở dữ liệu. Các bài học được trình bày ngắn gọn, có nhiều ví dụ minh hoạ. Cuối mỗi chương đều có bài tập để sinh viên luyện tập.

Chúng tôi mong rằng các sinh viên tự tìm hiểu trước mỗi vấn đề và kết hợp với bài giảng trên lớp của giáo viên để việc học môn này đạt hiệu quả.

Giáo trình bao gồm 04 chương:

Chương 1: Các khái niệm cơ bản trong cơ sở dữ liệu quan hệ;

Chương 2: Các ngôn ngữ thao tác dữ liệu;

Chương 3:Ràng buộc toàn vẹn;

Chương 4: Chuẩn hóa và thiết kế cớ sở dữ liệu.

Trong quá trình giảng dạy và biên soạn giáo trình này, chúng tôi đã nhận được sự đóng góp về mặt chuyên môn của các cán bộ giảng viên khoa Công Nghệ Thông Tin

Tôi xin chân thành cảm ơn và hy vọng rằng giáo trình này sẽ giúp cho việc dạy và học môn cơ sở dữ liệu của trường chúng ta ngày càng tốt hơn.

# CHƯƠNG 1: CÁC KHÁI NIỆM CƠ BẢN TRONG CƠ SỞ DỮ LIỆU QUAN HỆ

## 1.1. Tổng quan về cơ sở dữ liệu

### 1.1.1. Khái niệm cơ bản về cơ sở dữ liệu

**Dữ liệu** trong cơ sở dữ liệu là một phạm vi rất rộng bao gồm: chữ số, văn bản, đồ hoạ, video…

**Cơ Sở Dữ Liệu**là một tập dữ liệu gắn kết logic với nhau, thể hiện các khía cạnh khác nhau của thế giới thực. Một CSDL được xây dựng và sử dụng cho một số mục đích cụ thể. Nó được sử dụng từ một tập người dùng hay một ứng dụng cụ thể ngay từ khi mới thiết kế.

***Ví dụ 1.1:*** Một CSDL



Ưu điểm của cơ sở dữ liệu

* Giảm sự trùng lắp thông tin xuống mức thấp nhất và do đó bảo đảm được tính nhất quán và toàn vẹn dữ liệu.
* Đảm bảo dữ liệu có thể truy xuất theo nhiều cách khác nhau.
* Khả năng chia sẻ thông tin cho nhiều người sử dụng.

Mục tiêu của CSDL

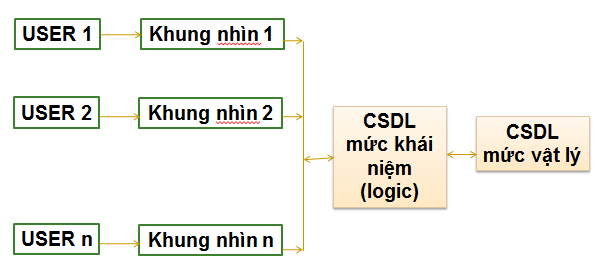
* Bảo đảm tính độc lập dữ liệu
* Biểu diễn dưới dạng lưu trữ vật lý là duy nhất
* Nhằm thoả mãn nhu cầu khai thác thông tin của một hay nhiều người sử dụng phục vụ cho nhiều mục đích khác nhau.

Một số đặc tính của CSDL

* ***Tính tự mô tả***: Hệ CSDL không chỉ chứa bản thân CSDL mà còn chứa định nghĩa đầy đủ (mô tả) của CSDL
* ***Tính độc lập giữa chương trình và dữ liệu***: khi có thay đổi nhỏ về cấu trúc ta ít phải sửa lại chương trình
* ***Tính trừu tượng dữ liệu***: Hệ CSDL cho phép trình bày dữ liệu ở một mức trừu tượng cho phép, nhằm che bớt những chi tiết lưu trữ thật của dữ liệu
* ***Tính nhất quán***: Lưu trữ dữ liệu thống nhất, tránh được tình trạng trùng lắp thông tin, tránh được việc tranh chấp dữ liệu, bảo đảm dữ liệu luôn đúng tại mọi thời điểm

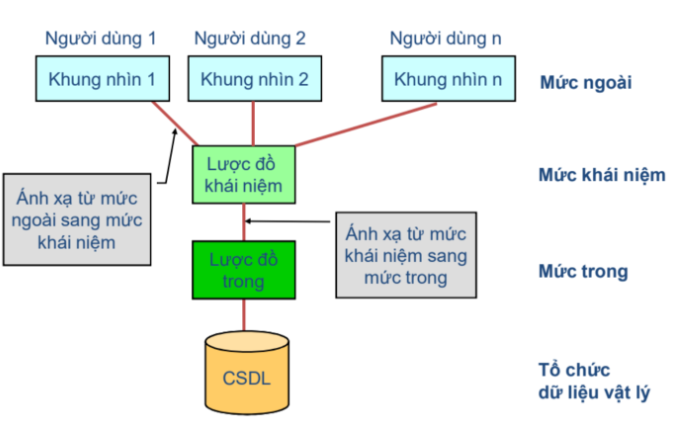
### 1.1.2. Kiến trúc một hệ CSDL

Một CSDL được phân thành các mức khác nhau



* ***CSDL mức vật lý***: là các tệp dữ liệu theo một cấu trúc nào đó được lưu trên các bộ nhớ thứ cấp. (là sự cài đặt cụ thể của CSDL mức khái niệm)
* ***CSDL mức khái niệm* (logic)**: là sự biểu diễn trừu tượng của CSDL vật lý tương ứng.
* ***Các khung nhìn* (view):** là quan niệm (cách nhìn) của từng người sử dụng đối với CSDL mức khái niệm.

**Kiến trúc 3 lược đồ**



* **Mức ngoài:**là khung nhìn của người sử dụng CSDL, mô tả phần CSDL tương ứng với người dùng đó. Chỉ bao gồm các thực thể, thuộc tính, quan hệ và người dùng quan tâm. Mỗi người dùng có một khung nhìn khác nhau. Ví dụ một khung nhìn có thể hiện ngày dạng dd-mm-yy (ngày tháng năm), nhưng khung nhìn khác lại hiện dạng mm-dd-yy (tháng ngày năm).
* **Mức khái niệm:**là khung nhìn của người thiết kế CSDL, mô tả dữ liệu nào được lưu trong CSDL và mối quan hệ giữa chúng. Họ sẽ thấy toàn bộ cấu trúc logic của CSDL.
* **Mức trong:**thể hiện biểu diễn về mặt vật lý của CSDL trong máy tính, mô tả cách thức lưu dữ liệu trong CSDL. Nó bao gồm các cấu trúc dữ liệu và tổ chức tệp lưu trữ dữ liệu trong các thiết bị nhớ.
* **Mức vật lý:**nằm dưới mức trong, được quản lý bởi hệ điều hành dưới sự chỉ dẫn của hệ quản trị CSDL.

### 1.1.3. Lược đồ khái niệm và mô hình dữ liệu

* **Lược đồ khái niệm** là sự biểu diễn thế giới thực bằng một loạt ngôn ngữ phù hợp. Hệ quản trị CSDL cung cấp ngôn ngữ định nghĩa dữ liệu để xác định lược đồ khái niệm.

**Mô hình dữ liệu** là sự trừu tượng hoá môi trường thực. Mỗi loại mô hình dữ liệu đặc trưng cho một cách tiếp cận dữ liệu khác nhau của những nhà phân tích thiết kế CSDL. Mô hình dữ liệu gồm:Mô hình phân cấp, Mô hình mạng, Mô hình quan hệ, Mô hình thực thể, Mô hình hướng đối tượng.

## 1.2. Mô hình cơ sở dữ liệu quan hệ

### 1.2.1. Các khái niệm cơ bản

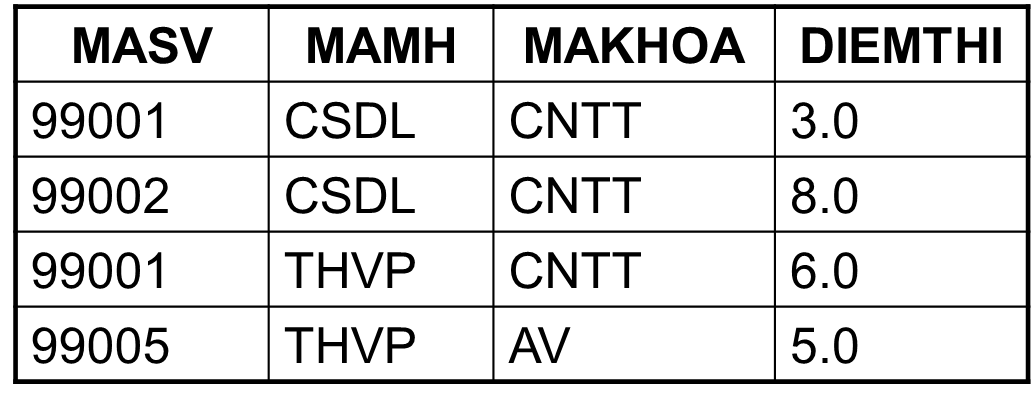
* *Mô hình Cơ sở dữ liệu quan hệ* (gọi tắt là Mô hình Quan hệ) được Ted Codd đưa ra đầu những năm 1970 và gây được chú ý ngay tức khắc vì tính đơn giản và các cơ sở toán học của nó
* Mô hình Cơ sở dữ liệu Quan hệ thể hiện dữ liệu dưới góc độ logic, biểu thị cơ sở dữ liệu như một tập các quan hệ.
* *Một mô hình CSDL quan hệ* bao gồm:
  + Các khái niệm nhằm mô tả dữ liệu dưới dạng dòng và cột như quan hệ, bộ, thuộc tính, khóa chính, khoá ngoại,
  + Các phép toán thao tác với dữ liệu\_ Đại số quan hệ
  + Ràng buộc toàn vẹn quan hệ
* Các Hệ quản trị CSDL quan hệ (Relation Database Management System - RDBMS) được xây dựng dựa trên lý thuyết mô hình quan hệ.
* Dữ liệu lưu trong CSDL quan hệ được tổ chức thành các quan hệ(relation)
* *Một quan hệ* có:
  + Một tên
  + Tập hợp các thuộc tính (attribute), có tên, kiểu dữ liệu và miền giá trị.
  + Tập hợp các bộ (tuple), có thể thay đổi theo thời gian

**a. Quan hệ- bảng**

* + Mỗi quan hệ được biểu diễn như một bảng giá trị, mỗi dòng được gọi là một bộ, mỗi cột được gọi là một thuộc tính.
  + Mỗi một dòng trong bảng biểu thị một tập hợp các giá trị dữ liệu liên quan với nhau. Tên bảng và tên các cột dùng để giúp giải thích ý nghĩa của các giá trị trong mỗi hàng. Mọi giá trị trong một cột đều cùng một kiểu dữ liệu.

***Thuật ngữ tương đương:***

* + Quan hệ, bộ, thuộc tính (Relation, tuple, attribute)
  + Bảng, dòng, cột (Table, row, column)



**b. Thuộc tính**

* ***Tên thuộc tính***
  + Tên phân biệt, giúp diễn giải ý nghĩa thuộc tính (thể hiện đặc điểm riêng của đối tượng hay thực thểtrong mô hình kết hợp)
* ***Kiểu dữ liệu thuộc tính***
  + Số nguyên, số thực, văn bản, logic,…
* ***Miền giá trị xác định***
  + Có thể bị áp đặt bởi qui tắc nghiệp vụ, hay ràng buộc dữ liệu
  + Có thể NULL (rỗng)

**c.** **Lược đồ quan hệ - relation schema**

* *Mô tả cấu trúc của quan hệ* 
  + Các thuộc tính và Mối liên hệ giữa các thuộc tính
  + Mỗi lược đồ quan hệ luôn kèm **một tân từ** để diễn tả ý nghĩa của nó.

***Ví dụ 1.2:*** tân từ của lược đồ quan hệ Sinh viên: Mỗi sinh viên có MaSV duy nhất. Mỗi MASV xác định các thuộc tính còn lại của sinh viên đó như: HOTENSV, GT, NGAYSINH, NOISINH, TINH, MALOP

* Kí hiệu:

**SINHVIEN(MASV,HOTENSV,GT,NGAYSINH,NOISINH,TINH,MALOP)**

**d. Bộ - dòng**

* Mỗi bộ (dòng) là một tổ hợp các giá trị tương ứng với các thuộc tính của quan hệ mô tả về một thực thể hay một mối kết hợp có trong thế giới thực
* Một tập các bộ xác định tại một thời điểm, gọi là một thể hiện của lược đồ quan hệ
* Ký hiệu: r(Ketqua)
* Không có 2 bộ trùng nhau trong một quan hệ (dựa trên khóa)
* Trật tự của các bộ (và các thuộc tính) là không quan trọng đối với DBMS.

### 1.2.2. Khái niệm khóa (KEY)

* Khóa của một quan hệ là một (hoặc một tập) thuộc tính được dùng để phân biệt hai bộ trong một cơ sở dữ liệu quan hệ. Hay nói cách khác, không tồn tại hai bộ có giá trị bằng nhau trên mọi thuộc tính của khóa.
* Mỗi giá trị của khóa là xác định duy nhất.
* Trong lược đồ quan hệ có thể có rất nhiều khóa.

***Ví dụ 1.3:***

* Gọi S là một tập các thuộc tính của lược đồ quan hệ R
  + S được gọi là một siêu khóa (superkey) của lược đồ quan hệ R, nếu với hai bộ bất kỳ trong R thì giá trị của các thuộc tính trong S là khác nhau
  + Siêu khoá có ít thuộc tính nhất được gọi là khóa (key) hay khóa dự tuyển (candidate key)
  + Một lược đồ quan hệ có thể có nhiều khóa dự tuyển.
  + Một khóa được chọn để cài đặt gọi là khóa chính (primary key).
  + Việc chọn khóa chính tốt nhất là khóa có một thuộc tính (hoặc số các thuộc tính ít nhất) và có giá trị hợp lệ (không chứa giá trị NULL)
  + **Khóa ngoại (foreign key)** là thuộc tính của lược đồ quan hệ này nhưng lại là khóa chính của lược đồ quan hệ khác
  + **Khóa phức (composite key)** là khóa có nhiều hơn một thuộc tính (từ hai thuộc tính trở lên)
  + **Thuộc tính khoá**: là các thuộc tính có tham gia vào một khoá (khoá dự tuyển hay khoá chính). Ngược lại, thuộc tính không tham gia vào một khoá nào gọi là thuộc tính không khoá.

***Ví dụ 1.4*: Monhoc(Mamon, Tenmon, Sotiet)**

*r(Monhoc)*

|  |  |  |
| --- | --- | --- |
| Mamon | Tenmon | Sotiet |
| THVP | Tin hoc văn phòng | 30 |
| LTC | Lập trình C | 60 |
| CSDL1 | Co so du lieu | 45 |
| CSDL2 | Co so du lieu | 45 |

Siêu khóa : {Mamon}, {Mamon, Tenmon}, {Mamon, Sotiet},{Mamon, Tenmon, Sotiet}

Khóa (khóa dự tuyển, khóa chính) : {Mamon}

***Ví dụ 1.5*: Sinhvien(MaSV,Hoten,Phai,soCMND)**

rSV

|  |  |  |  |
| --- | --- | --- | --- |
| MaSV | Hoten | Phai | soCMND |
| 99001 | Nguyen van anh | Nam | 01245012 |
| 99002 | Tran Le Tuan | Nam | 02209875 |
| 99003 | Nguyen Thi Hong | Nu | 04563711 |
| 99004 | Do van Thuan | Nam |  |

Siêu khóa : {MaSV} , {MaSV, Hoten}, … , {soCMND} , {soCMND, Hoten},…

Khóa (khóa dự tuyển): {MaSV} , {soCMND}

Khóa chính : {MaSV}

**1.2.3.** **Các phép tính cơ bản trên CSDL QH**

**Phép chèn (Insert - thêm mới)**

Phép chèn cung cấp một danh sách các giá trị cho một bộ mới được chèn vào trong một quan hệ R.

Cú pháp:

Insert(r; A1=d1, A2=d2, ... , An=dn)

Trong đó:

Ai là các thuộc tính

di là các giá trị mới

***Ví dụ 1.6:***

INSERT (R,msv='01',ht='nguyen van Anh',lop='12',14)

Kết quả của phép Insert thường gây ra một số sai sót:

* + Bộ mới được thêm vào không phù hợp với lược đồ quan hệ cho trước
  + Một số giá trị của một số thuộc tính nằm ngoài miền giá trị của thuộc tính đó
  + Giá trị khóa của bộ mới có thể là giá trị đã có trong quan hệ đang lưu trữ

**Phép xóa (Delete – loại bỏ)**

Phép xoá được sử dụng để xoá một hoặc nhiều bộ giá trị của một quan hệ R.

Cú pháp:

Delete(r; A1=d1, A2=d2, ... , An=dn)

Trong đó:

Ai là các thuộc tính

di là các giá trị mới

Không cần đầy đủ thông tin về cả bộ

* + Phép xoá không thực hiện được trong trường hợp bộ bị xoá được tham chiếu bởi một khoá ngoài từ các bộ khác trong cơ sở dữ liệu.
  + Để chỉ rõ một phép xoá, cần phải đưa ra một (hoặc một số) điều kiện cơ bản trên các thuộc tính của quan hệ để chọn các bộ sẽ bị xoá

***Ví dụ 1.7:***

DEL(R,msv='01',ht='nguyen van Anh',lop='12',14)

**Phép sửa (change – thay đổi)**

Phép sửa được dùng để thay đổi các giá trị của một hoặc nhiều thuộc tính trong một (hoặc nhiều) bộ của một quan hệ R.

Cú pháp:

change(r; B1=e1,... Bm=em; A1=d1, A2=d2, ... , An=dn)

Trong đó:

Bi và ei là thuộc tính và giá trị điều kiện

Ai là các thuộc tính cần sửa

di là các giá trị mới

* + Để lựa chọn các bộ cần được thay đổi, người sử dụng phải chỉ ra một điều kiện trên các thuộc tính.
  + Phép sửa là tổ hợp của phép chèn và phép xóa nên những sai sót của phép sửa cũng sẽ xảy ra tương tự phép chèn và phép xóa.

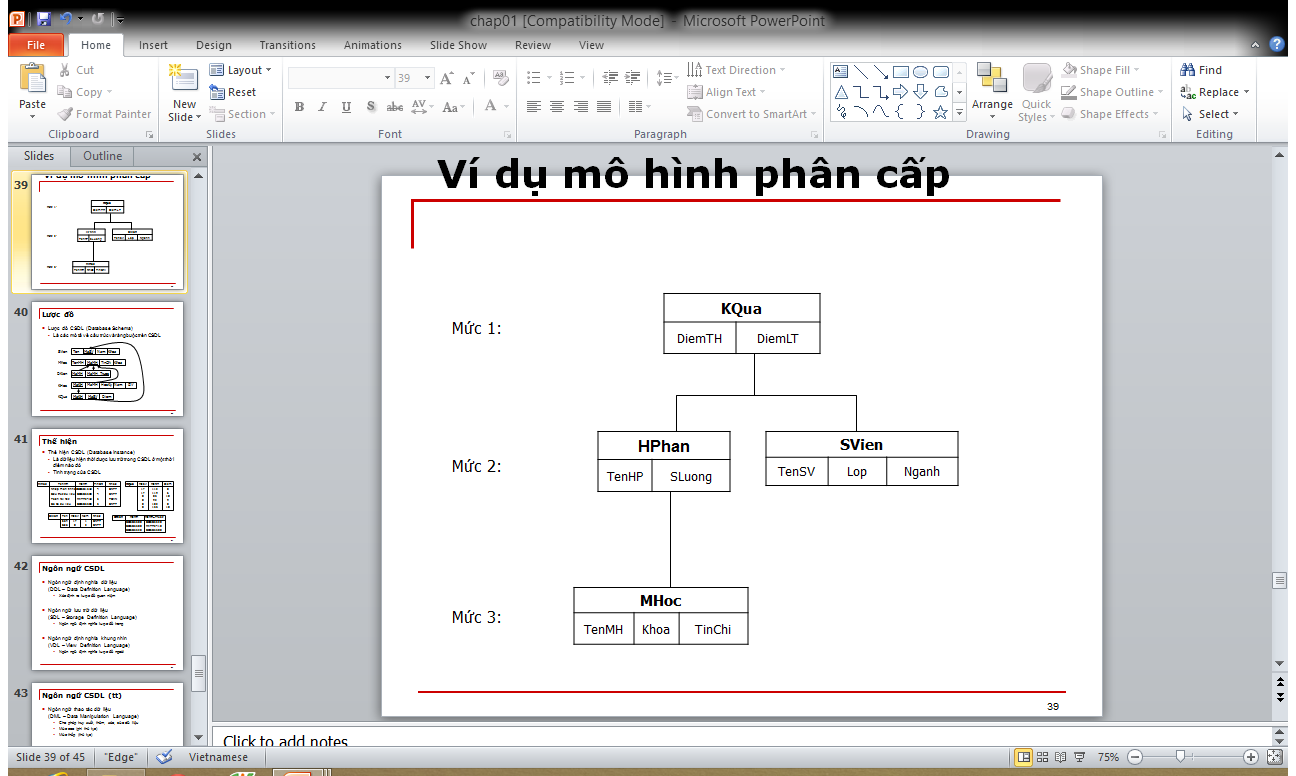
***Ví dụ 1.8***: Muốn sửa đổi td=16 trong quan hệ R(msv,ht,lop,td ) với sinh viên msv='01',ht='Nguyen Van Anh',lop='12' , td=14 ta viết

CH(R,msv='01',ht='nguyen van Anh',lop='12',16)

### 1.2.4. Các mô hình CSDL

**\* Mô hình phân cấp**

Mô hình phân cấp hay mô hình CSDL dạng cây được tổ chức theo cấu trúc từ trên xuống dưới giống như cây lộn ngược. Mỗi nút tương ứng với một kiểu dữ liệu, có thể có một hoặc nhiều trường, mô tả thực thể và một nhánh cây tạo nên một liên kết giữa kiểu dữ liệu này với kiểu dữ liệu khác. Mỗi nút đều có một nút cha và nhiều nút con, trừ nút gốc là không có cha.

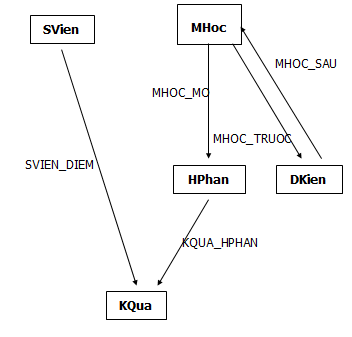
***Ví dụ 1.9:***

- Tuy nhiên, mô hình chỉ thể hiện được quan hệ 1-n, tức là mô tả được trường hợp nút cha có nhiều nút con như một phòng thì có thể có nhiều nhân viên hay một phòng có thể có nhiều dự án, còn trường hợp ngược lại thì không. Chẳng hạn nếu một dự án mà thuộc về nhiều phòng thì dự án phải được lưu ở nhiều nơi khác nhau. Điều này gây dư thừa dữ liệu và lãng phí không gian lưu trữ.

- Điểm nổi bật trong các thủ tục truy xuất đến một đối tượng trong mô hình phân cấp là đường dẫn đi từ gốc đến phần tử cần xét trong cây phân cấp.

**\* Mô hình mạng**

- Mô hình mạng được biểu diễn như một đồ thị có hướng. Mỗi nút có thể nối với một nút bất kỳ để biểu diễn một liên kết 1-n thông qua con trỏ liên kết. Sự khác nhau chính giữa hệ thống mạng và hệ thống phân cấp là mô hình mạng không ràng buộc về số và hướng của các liên kết thiết lặp giữa các nút.

***Ví dụ 1.10:***

**\* Mô hình quan hệ**

- Cơ sở dữ liệu quan hệ được xây dựng theo lý thuyết do E.F.Codd giới thiệu năm 1970. Thuật ngữ “quan hệ” là do bảng dữ liệu hai chiều được Codd gọi là bảng quan hệ. Mô hình quan hệ khác hẳn với các mô hình trước nó và từ năm 1980 đã trở thành mô hình được dùng rộng rãi để phát triển hệ quản trị CSDL.

- Theo mô hình quan hệ, dữ liệu được thể hiện trong bảng hai chiều, gồm các dòng và cột. Các bảng gọi là các “quan hệ”, các dòng gọi là các “bộ” và cột là “thuộc tính”. Theo cách nhìn của các mô hình trước thì mỗi dòng là một bản ghi, các thuộc tính cho biết ý nghĩa của các giá trị trong bản ghi.

\* **Mô hình thực thể**

Hiện nay mô hình dữ liệu quan hệ thường được dùng trong các hệ quản trị CSDL, đây là mô hình dữ liệu ở mức vật lý. Để thành lập được mô hình này, thường là phải dùng mô hình dữ liệu ở mức quan niệm để đặc tả, một trong những mô hình ở dạng đó là mô hình thực thể kết hợp (sau đó mới dùng một số quy tắc để chuyển hệ thống từ mô hình này về mô hình dữ liệu quan hệ)

Sau đây là các khái niệm của mô hình thực thể kết hợp.

Thực Thể (entity)

Thực thể là một sự vật tồn tại và phân biệt được, chẳng hạn sinh viên Nguyễn Văn Thành, lớp Cao Đẳng Tin Học 2A, môn học Cơ Sở Dữ Liệu, xe máy có biển số đăng ký 52-0549,… là các ví dụ về thực thể.

Thuộc tính (attribute)

Các đặc điểm riêng của thực thể gọi là các *thuộc tính.*

Chẳng hạn các thuộc tính của sinh viên Nguyễn Văn Thành là:mã số sinh viên, giới tính, ngày sinh, hộ khẩu thường trú, lớp đang theo học, …

(Trong giáo trình này, tên thuộc tính được viết bằng chữ in hoa)

Loại thực thể (entity type)

Là tập hợp các thực thể có cùng thuộc tính. Mỗi loại thực thể đều phải được đặt tên sao cho có ý nghĩa. Một loại thực thể được biểu diễn bằng một hình chữ nhật.

***Ví dụ 1.12:*** các sinh viên có mã sinh viên là ““02CĐTH019”, “02CĐTH519”, “02TCTH465”,… nhóm lại thành một loại thực thể, được đặt tên là Sinhvien chẳng hạn.

Tương tự trong ứng dụng quản lý điểm của sinh viên (sẽ được trình bày ngay sau đây) ta có các loại thực thể như Monhoc, Lop, Khoa,…

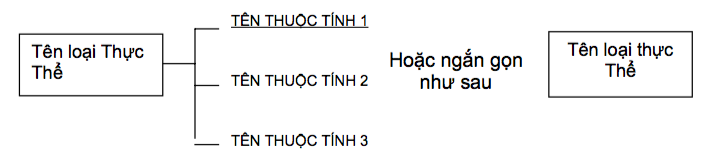
(Trong giáo trình này, tên của loại thực thể được in hoa ký tự đầu tiên, các ký tự còn lại viết thường).

Khoá (key)

Khoá của loại thực thể E là một hay một tập các thuộc tính của E có thể dùng để phân biệt hai thực thể bất kỳ của E.

***Ví dụ 1.13:*** khoá của loại thực thể Sinhvien là MASV, của Lớp là MALOP, của Khoa là MAKHOA, của Monhoc là MAMH,…

Cần chú ý rằng khi biểu diễn một hệ thống bằng mô hình thực thể kết hợp thì tên của các loại thực thể phải khác nhau. Trong danh sách các thuộc tính của một loại thực thể thì tập thuộc tính khoá thường được gạch dưới liền nét. Nếu một hệ thống có nhiều loại thực thể, để đơn giản hoá mô hình, người ta có thể chỉ nêu tên các loại thực thể; còn các thuộc tính của loại thực thể được liệt kê riêng.

**

##### *Ví dụ 1.14:*

Bài toán quản lý điểm của sinh viên được phát biểu sơ bộ như sau:

Mỗi sinh viên cần quản lý các thông tin như: họ và tên (HOTENSV), ngày tháng năm sinh(NGAYSINH), giới tính (NU), nơi sinh(NƠISINH), hộ khẩu thường trú (TINH). Mỗi sinh viên được cấp một mã số sinh viên duy nhất (MASV) để phân biệt với mọi sinh viên khác của trường, mỗi sinh viên chỉ thuộc về một lớp nào đó.

Mỗi lớp học có một mã số lớp (MALOP)duy nhất để phân biệt với tất cả các lớp học khác trong trường: có một tên gọi (TENLOP) của lớp, mỗi lớp chỉ thuộc về một khoa.

Mỗi khoa có một tên gọi (TENKHOA) và một mã số duy nhất (MAKHOA) để phân biệt với các khoa khác.

Mỗi môn học có một tên gọi (TENMH) cụ thể, được học trong một số đơn vị học trình (DONVIHT) )và ứng với môn học là một mã số duy nhất (MAMH) để phân biệt với các môn học khác.

Mỗi giảng viên cần quản lý các thông tin: họ và tên(HOTENGV), cấp học vị (HOCVI), thuộc một chuyên ngành (CHUYENNGANH) và được gán cho một mã số duy nhất gọi là mã giảng viên(MAGV) để phân biệt với các giảng viên khác. Mỗi giảng viên có thể dạy nhiều môn ở nhiều khoa, nhưng chỉ thuộc về sự quản lý hành chính của một khoa.

Mỗi sinh viên với một môn học được phép thi tối đa 3 lần, mỗi lần thi (LANTHI), điểm thi (DIEMTHI).

Mỗi môn học ở mỗi lớp học chỉ phân công cho một giảng viên dạy (tất nhiên là một giảng viên thì có thể dạy nhiều môn ở một lớp).

Với bài toán trên thì các loại thực thể cần quản lý như: Sinhviên, Mônhọc, Khoa, Lớp, Giảngviên.

Ví dụ với loại thực thể Sinhviên thì cần quản lý các thuộc tính như: MASV,HOTENSV, NGAYSINH,… và ta có thể biểu diễn như sau:

HOTENSV

NGAYSINH

MASV

Sinhvien

Mối kết hợp (relationship)

Mối kết hợp diễn tả sự liên hệ giữa các loại thực thể trong một ứng dụng tin học.

Ví dụ mối kết hợp giữa hai loại thực thể Sinhviên và Lop, mối kết hợp giữa Sinhviên với Mônhọc,...

Mối kết hợp được biểu diễn bằng một hình elip và hai bên là hai nhánh gắn kết với các loại thực thể (hoặc mối kết hợp) liên quan, tên mối kết hợp thường là: *thuộc, gồm , chứa,...*

Chẳng hạn giữa hai loại thực thể Lớp và Khoa có mối kết hợp “thuộc” như sau:

Sinh viên

Lop

thuộc

##### Bản số của mối kết hợp:

Bản số của một nhánh R trong mối kết hợp thể hiện số lượng các thực thể thuộc thực thể ở nhánh “bên kia” có liên hệ với một thực thể của nhánh R.

Mỗi bản số là một cặp số (min,max), chỉ số lượng tối thiểu và số lượng tối đa của thực thể khi tham gia vào mối kết hợp đó.

***Ví dụ 1.15:***

(1,1) (1,n)

Sinh viên

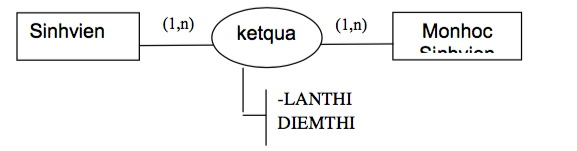
Lop

thuoc

Có nghĩa là: “mỗi sinh viên thuộc một và chỉ một lớp nên bản số bên nhánh Sinhviên là (1,1), mỗi lớp có 1 đến n sinh viên nên bản số bên nhánh Lop là (1,n)”.

Trong một số trường hợp đặc biệt, mối kết hợp có thể có các thuộc tính đi kèm và do đó chúng thường được đặt tên ý với nghĩa đầy đủ hơn.

Ví dụ giữa hai loại thực thể Monhoc và Sinhvien có mối kết hợp ketqua với ý nghĩa: “mỗi sinh viên ứng với mỗi lần thi của mỗi môn học có một kết quả điểm thi duy nhất”.



Khoá của mối kết hợp: là hợp của các khoá của các loại thực thể liên quan. Chẳng hạn như thuộc tính MAGV là khoá của loại thực thể Giangvien, MALOP là thuộc tính khoá của loại thực thể Lop, MAMH là thuộc tính khoá của loại thực thể Monhoc, do đó mối kết hợp phancong (giữa các loại thực thể Giangvien,Lop,Monhoc) có khoá là {MAGV,MAMH,MALOP} - phancong là mối kết hợp 3 ngôi.

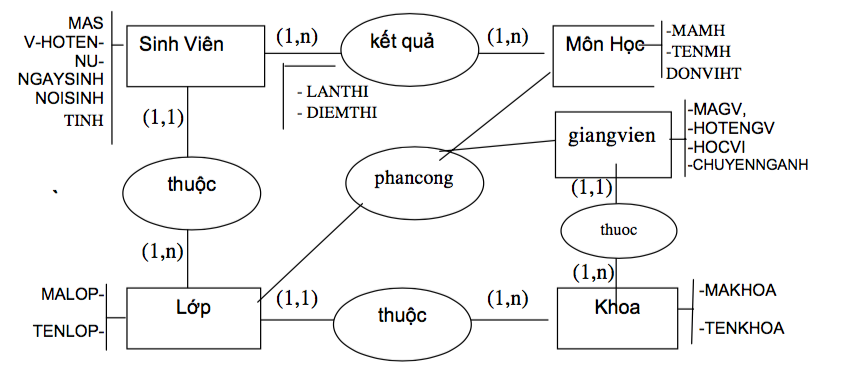
(Trong giáo trình này, tên của mối kết hợp được viết toàn bằng chữ thường). Việc thành lập mô hình thực thể kết hợp cho một ứng dụng tin học có thể tiến hành theo các bước sau:

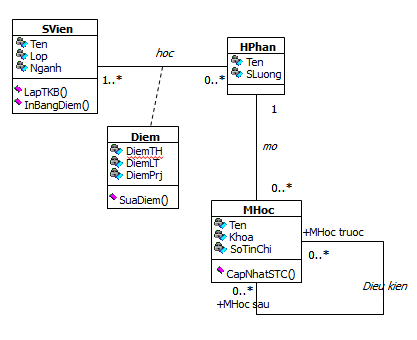
Bước 1: Xác định danh sách các loại thực thể

Bước 2: Xác định các mối kết hợp giữa các loại thực thể để phác thảo mô hình.

Bước 3: Lập bản số của các mối kết hợp.

***Ví dụ 1.16:*** Lập mô hình thực thể kết hợp cho bài toán quản lý điểm của sinh viên đã được nêu trong ví dụ 1.14



\* ***Mô hình hướng đối tượng***

* + - Sự ra đời: những năm 1990
    - Biểu diễn: sơ đồ lớp
    - Các khái niệm cơ bản:
      * Đối tượng
      * Thuộc tính
      * Phương thức
      * Lớp

**BÀI TẬP CHƯƠNG 1**

**Bài 1.1: Xây dựng các bảng CSDL quan hệ**

CSDL “THƯ VIỆN”, được mô tả như sau:

1) Mỗi cuốn sách trong thư viện có các thông tin về Mã sách, Tên sách Nhà xuất bản và Tác giả…

2) Một tác giả có thể viết nhiều cuốn sách. Một cuốn sách có thể có nhiều tác giả viết. Thông tin về tác giả gồm có Tên tác giả, Năm sinh, Quê quán.

3) Một nhà xuất bản xuất bản nhiều cuốn sách. Một cuốn sách do một nhà xuất bản xuất bản. Thông tin về Nhà xuất bản gồm có Tên, Địa chỉ và Số điện thoại.

4) Thư viện có những người mượn sách. Thông tin về những người mượn sách gồm có Số thẻ, Họ tên, Địa chỉ và Số điện thoại.

5) Thông tin về một lần mượn gồm có Ngày mượn và ngày trả.

**Bài 1.2:**

1. Viết 5 phép chèn dữ liệu vào 5 bảng dữ liệu đã xây dựng ở trên.
2. Viết 3 phép xóa với các điều kiện trên thuộc tính: Mã sách, Số thẻ, Tên tác giả.
3. Viết 2 phép sửa với các điều kiện trên thuộc tính: Tên nhà xuất bản, Tên người mượn

**Bài 1.3:** Dựa vào các phân tích sơ bộ dưới đây, hãy lập mô hình thực thể kết hợp (gồm loại thực thể, mối kết hợp, bản số, thuộc tính của loại thực thể, khoá của loại thực thể ) cho mỗi bài toán quản lý sau:

**Bài 1.3.1: Quản lý số lượng ngày công của các nhân viên**

Để quản lý việc phân công các nhân viên tham gia vào xây dựng các công trình. Công ty xây dựng ABC tổ chức quản lý như sau:

Cùng lúc công ty có thể tham gia xây dựng nhiều công trình, mỗi công trình có một mã số công trình duy nhất (MACT), mỗi mã số công trình xác định các thông tin như: Tên gọi công trình (TENCT), địa điểm(ĐIAĐIEM), ngày công trình được cấp giấy phép xây dựng (NGAYCAPGP), ngày khởi công (NGAYKC), ngày hoàn thành (NGAYHT).

Mỗi nhân viên của công ty ABC có một mã số nhân viên(MANV) duy nhất, một mã số nhân viên xác định các thông tin như: Họ tên (HOTEN), ngày sinh(NGSINH), phái (PHAI), địa chỉ (ĐIACHI), phòng ban, …

Công ty phân công các nhân viên tham gia vào các công trình, mỗi công trình có thể được phân cho nhiều nhân viên và mỗi nhân viên cùng lúc cũng có thể tham gia vào nhiều công trình. Với mỗi công trình một nhân viên có một số lượng ngày công (SLNGAYCONG) đã tham gia vào công trình đó.

Công ty có nhiều phòng ban(Phòng kế toán, phòng kinh doanh, phòng kỹ thuật, phòng tổ chức, phòng chuyên môn, Phòng phục vụ,…). Mỗi phòng ban có một mã số phòng ban(MAPB) duy nhất, một phòng ban ứng với một tên phòng ban(TENPB).

**Bài 1.3.2: Quản lý việc mượn/trả sách ở một thư viện**

Một thư viện tổ chức việc cho mượn sách như sau:

Mỗi quyển sách được đánh một mã sách (MASH) dùng để phân biệt với các quyển sách khác (giả sử nếu một tác phẩm có nhiều bản giống nhau hoặc có nhiều tập thì cũng xem là có mã sách khác nhau), mỗi mã sách xác định các thông tin khác như : tên sách (TENSACH), tên tác giả (TACGIA), nhà xuất bản (NHAXB), năm xuất bản (NAMXB).

Mỗi độc giả được thư viện cấp cho một thẻ thư viện, trong đó có ghi rõ mã độc giả (MAĐG), cùng với các thông tin khác như : họ tên (HOTEN), ngày sinh (NGAYSINH), địa chỉ (ĐIACHI), nghề nghiệp(NGHENGHIEP).

Cứ mỗi lượt mượn sách, độc giả phải đăng ký các quyển sách cần mượn vào một phiếu mượn, mỗi phiếu mượn có một số phiếu mượn (SOPM) khác nhau, mỗi phiếu mượn xác định các thông tin như: ngày mượn sách (NGAYMUON), mã độc giả. Các các quyển sách trong cùng một phiếu mượn không nhất thiết phải trả trong một lần. Mỗi quyển sách có thể thuộc nhiều phiếu mượn khác nhau (tất nhiên là tại các thời điểm khác nhau).

# CHƯƠNG 2. NGÔN NGỮ THAO TÁC DỮ LIỆU

## 2.1 Đại số quan hệ

### 2.1.1 Phép chiếu

***a. Mục đích:***

Phép chiếu trên một quan hệ thực chất là loại bỏ đi một số thuộc tính của quan hệ đó.

***b. Định nghĩa:***

Phép chiếu quan hệ R(U) trên tập con thuộc tính X ⊆ U, ký hiệu R[X], cho ta quan hệ P(X) như sau:

P(X) = R[X] = {t.X | t ∈ R}

R[X] được tính theo 2 bước sau:

+ Xoá các cột không thuộc X của bảng R.

+ Xoá bớt các dòng không giống nhau trong bảng kết quả P(X): chỉ giữ lại một dòng trong số các dòng giống nhau.

*Ví dụ 2.1*

Cho quan hệ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MONHOC | maM | tenM | soĐVHT | coĐA |
|  | 01 | Cơ sở dữ liệu | 5 | .f. |
|  | 02 | Cấu trúc dữ liệu | 4 | .f. |
|  | 03 | Phân tích hệ thống | 6 | .f. |
|  | 04 | Cấu trúc máy tính | 4 | .t. |

cho biết mã và tên của các môn học:

KQ = MONHOC[maM, tenM]

kết quả:

|  |  |  |
| --- | --- | --- |
| KQ | maM | tenM |
|  | 01 | Cơ sở dữ liệu |
|  | 02 | Cấu trúc dữ liệư |
|  | 03 | Phân tích hệ thống |
|  | 04 | Cấu trúc máy tính |

### 2.1.2 . Phép chọn

***a. Mục đích:***

Phép chọn là phép tính lọc ra một tập con các bộ của quan hệ R đã cho, thỏa biểu thức điều kiện cho trước.

***b. Định nghĩa:***

Cho quan hệ R trên tập thuộc tính U – ký hiệu R(U) và biểu thức điều kiện e (còn gọi là biểu thức lọc hay biểu thức chọn). Phép chọn trên quan hệ R theo điều kiện e, ký hiệu R(e) cho ta quan hệ P(U) như sau:

P(U) = R(e) = {t ∈ R | Sat(t,e)}

trong đó hàm logic Sat(t,e) kiểm tra bộ t thỏa điều kiện e được xác định như sau:

+ Thay mọi xuất hiện của thuộc tính A trong biểu thức chọn e bằng trị tương ứng của A trong bộ t, t.A, ta thu được một mệnh đề logic A.

+ Tính trị của b. Nếu là đúng (True) thì bộ t thỏa điều kiện e; ngược lại, nếu trị b là sai (False) thì bộ t không thỏa điều kiện e.

***c. Các phép toán:***

- Các toán hạng là hằng hoặc số hiệu thành phần.

- Các phép toán so sánh: <, =, >, ≠, ≤, ≥

- Các phép toán logic: &, ∧, AND (và), ∨, |, OR (hoặc), ¬, !, NOT (phủ định).

*Ví dụ 2.2*

Cho quan hệ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MONHOC | maM | tenM | soĐVHT | coĐA |
|  | 01 | Cơ sở dữ liệu | 5 | .f. |
|  | 02 | Cấu trúc dữ liệu | 4 | .f. |
|  | 03 | Phân tích hệ thống | 6 | .f. |
|  | 04 | Cấu trúc máy tính | 4 | .t. |

Chọn ra các môn có 4 đơn vị học trình và có đề án cho sinh viên thực hiện:

MON\_ĐA = MONHOC(soĐVHT = 4 ∧ coĐA)

kết quả:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MON\_ĐA | maM | tenM | soĐVHT | coĐA |
|  | 04 | Cấu trúc máy tính | 4 | .t. |

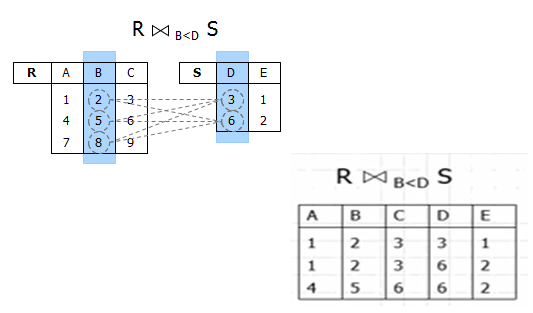
### 2.1.3. Phép kết nối

* + Phép kết
  + Kết tự nhiên (Natural join)
  + Kết có điều kiện tổng quát (Theta join)
  + Kết bằng (Equi join)
  + Được dùng để tổ hợp 2 bộ có liên quan từ 2 quan hệ thành 1 bộ
  + Ký hiệu R S
  + R(A1, A2, …, An) và S(B1, B2, …, Bm)
  + Kết quả của phép kết là một quan hệ Q mới
  + Có n + m thuộc tính Q(A1, A2, …, An, B1, B2, …, Bm)
  + Mỗi bộ của Q là tổ hợp của 2 bộ trong R và S, thỏa mãn một số điều kiện kết nào đó
    - Có dạng Ai θ Bj
    - Ai là thuộc tính của R, Bj là thuộc tính của S
    - Ai và Bj có cùng miền giá trị
    - θ là phép so sánh ≠, =, <, >, ≤, ≥

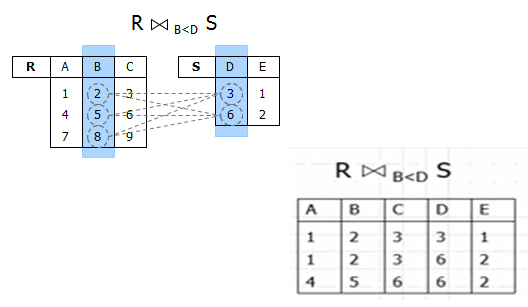
**Phân loại**

* + Kết theta (theta join) là phép kết có điều kiện
    - Ký hiệu R C S
    - C **gọi là điều kiện kết trên thuộc tính**
  + Kết bằng (equi join) khi C là điều kiện so sánh bằng
  + Kết tự nhiên (natural join)
    - Ký hiệu R S hay R \* S
    - R+ ∩ S+ ≠ ∅
    - Kết quả của phép kết bằng bỏ bớt đi 1 cột giống nhau

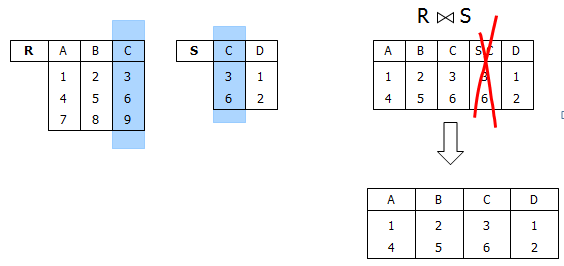
Chú *ý: 2 bảng phải có chung cột, chung tên, chung miền giá trị*

***Ví dụ 2.3***: ***phép kết theta***

***Ví dụ 2.4: phép kết bằng***



***Ví dụ 2.5: phép kết tự nhiên***



**Phân biệt các phép kết**

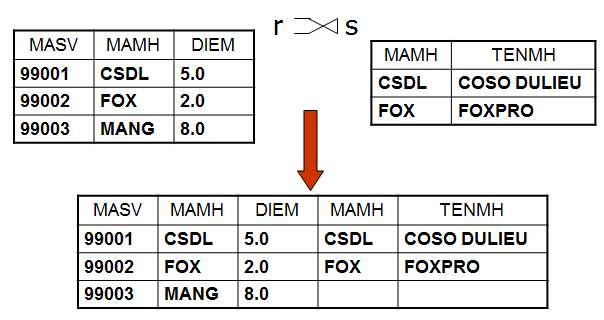
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Phép so sánh** | **Áp dụng trên** | **Tập kết quả** |
| **Theta join** | **Tất cả** | **Trên 2 thuộc tính cùng kiểu dữ liệu** |  |
| **Equijoin** | **phép bằng** | **Trên 2 thuộc tính cùng kiểu dữ liệu** |  |
| **Natural join** | **phép bằng** | **Trên 2 thuộc tính chung** | **Không lặp lại thuộc tính chung** |

**\* Phép kết ngoài - Outer join**

* + Bao gồm : Left/Right Outer Join, Full Outer Join
  + Phép kết Left Outer Join giữa r và s, cho phép các bộ của r không kết được với các bộ của s cũng được xuất hiện trong quan hệ kết quả.
  + Phép kết Right Outer Join giữa r và s, cho phép các bộ của s không kết được với các bộ của r cũng được xuất hiện trong quan hệ kết quả.
  + Phép kết Full Outer Join giữa r và s, cho phép tất các bộ của r và s không kết được với nhau cũng được xuất hiện trong quan hệ kết quả
  + Những giá trị tương ứng với các bộ trong quan hệ bị thiếu sẽ được gán trị Null
  + Ưu điểm: giữ được thông tin mà lẽ ra bị mất trong phép kết



***Ví dụ 2.6: Phép kết ngoài – Left Outer join***



### 2.1.4 Phép tập hợp

**2.1.4.1 Phép cộng (hợp)**

***a. Mục đích:***

Phép hợp của hai quan hệ thực chất là phép nối dữ liệu của hai quan hệ; điều kiện là hai quan hệ này phải có cùng số thuộc tính.

***b. Định nghĩa:***

Phép cộng hai quan hệ tương thích R(U) và S(U), ký hiệu R+S, cho ta quan hệ chứa các bộ của mỗi quan hệ thành phần.

P(U) = R+S = {t | t ∈ R ∨ t ∈ S}

*Ví dụ 2.4*

Cho 2 quan hệ sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BIENCHE | maNV | hotenNV | chucvu | donvi | mucLG |
|  | 001 | Lê Văn Sỹ | GĐ | BGĐ | 550 |
|  | 002 | Nguyễn Chí Thanh | PGĐ | BGĐ | 500 |
|  | 003 | Châu Văn Liêm | TP | HCTH | 480 |
|  | 004 | Nguyễn An Ninh | PP | HCTH | 475 |
|  | 005 | Phan Đình Phùng |  | HCTH | 333 |
|  | 006 | Ngô Gia Tự |  | TV | 290 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HOPĐONG | maNV | hotenNV | chucvu | donvi | mucLG |
|  | 011 | Lê Văn Tám |  | BV | 290 |
|  | 012 | Phạm Văn Đồng |  | BV | 255 |
|  | 013 | Nguyễn Văn Linh |  | ĐTTH | 310 |
|  | 014 | Kim Đồng |  | ĐTTH | 310 |

Trong cơ quan, có hai loại nhân viên: biên chế và hợp đồng. Danh sách chung toàn bộ của cơ quan là:

NHANVIEN = BIENCHE + HOPĐONG

kết quả:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NHANVIEN | maNV | hotenNV | chucvu | donvi | mucLG |
|  | 001 | Lê Văn Sỹ | GĐ | BGĐ | 550 |
|  | 002 | Nguyễn Chí Thanh | PGĐ | BGĐ | 500 |
|  | 003 | Châu Văn Liêm | TP | HCTH | 480 |
|  | 004 | Nguyễn An Ninh | PP | HCTH | 475 |
|  | 005 | Phan Đình Phùng |  | HCTH | 333 |
|  | 006 | Ngô Gia Tự |  | TV | 290 |
|  | 011 | Lê Văn Tám |  | BV | 290 |
|  | 012 | Phạm Văn Đồng |  | BV | 255 |
|  | 013 | Nguyễn Văn Linh |  | ĐTTH | 310 |
|  | 014 | Kim Đồng |  | ĐTTH | 310 |

**2.1.4.2 Phép trừ**

***a. Định nghĩa:***

Phép trừ hai quan hệ tương thích R(U) và S(U), ký hiệu R-S, cho ta quan hệ chứa các bộ của quan hệ R nhưng không có trong S.

P(U) = R-S - {t | t ∈ R, t ∉ S}

*Ví dụ 2.5*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NHANVIEN | maNV | hotenNV | chucvu | donvi | mucLG |
|  | 001 | Lê Văn Sỹ | GĐ | BGĐ | 550 |
|  | 002 | Nguyễn Chí Thanh | PGĐ | BGĐ | 500 |
|  | 003 | Châu Văn Liêm | TP | HCTH | 480 |
|  | 004 | Nguyễn An Ninh | PP | HCTH | 475 |
|  | 005 | Phan Đình Phùng |  | HCTH | 333 |
|  | 006 | Ngô Gia Tự |  | TV | 290 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LANHĐAO | maNV | hotenNV | chucvu | donvi | mucLG |
|  | 001 | Lê Văn Sỹ | GĐ | BGĐ | 550 |
|  | 002 | Nguyễn Chí Thanh | PGĐ | BGĐ | 500 |
|  | 003 | Châu Văn Liêm | TP | HCTH | 480 |
|  | 004 | Nguyễn An Ninh | PP | HCTH | 475 |

NHANVIEN là quan hệ chứa các bộ về tất cả các nhân viên trong cơ quan, còn LANHĐAO chỉ liên quan đến những người lãnh đạo. Tìm những nhân viên không là lãnh đạo trong cơ quan.

KQ = NHANVIEN – LANHĐAO

kết quả:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| KQ | maNV | hotenNV | chucvu | donvi | mucLG |
|  | 005 | Phan Đình Phùng |  | HCTH | 333 |
|  | 006 | Ngô Gia Tự |  | TV | 290 |

**2.1.4.3. Phép giao**

***a. Định nghĩa:***

Phép giao hai quan hệ tương thích R(U) và S(U), ký hiệu R&S, cho ta quan hệ chứa các bộ xuất hiện đồng thời trong cả hai quan hệ thành phần.

P(U) = R&S = {t | t ∈ R, t ∈ S}

*Ví dụ 2.6*

Cho 2 quan hệ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NHAHANG | ten | diachi | phuong | quan |
|  | Huy Hoàng | 504 Điện Biên Phủ | 14 | 3 |
|  | Tương Lai | 229 Ngô Chí Thanh | 6 | 5 |
|  | Sinh Đôi | 300 Lý Thái Tổ | 6 | 10 |
|  | Đông Nam Á | 230 Ngô Gia Tự | 10 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KHACHSAN | ten | diachi | phuong | quan |
|  | Viễn Đông | 2 Lê Lợi | 5 | 1 |
|  | Huy Hoàng | 504 Điện Biên Phủ | 14 | 3 |
|  | Ánh Hồng | 151 Đặng Dung | 2 | PN |
|  | Tương Lai | 229 Ngô Chí Thanh | 6 | 5 |

Tìm các nơi vừa là khách sạn, vừa là nhà hàng.

NH\_KS = NHAHANG & KHACHSAN

kết quả:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NH\_KS | ten | diachi | phuong | quan |
|  | Huy Hoàng | 504 Điện Biên Phủ | 14 | 3 |
|  | Tương Lai | 229 Ngô Chí Thanh | 6 | 5 |

**2.1.4.5. Phép chia**

***a. Định nghĩa:***

Cho hai quan hệ R(U) và S(V), phép chia quan hệ R cho S là một quan hệ gồm các bộ t có M=U-V thuộc tính sao cho:

P(M) = R : S = {t.M | t ∈ R, (t.M)\*S ⊆ R, M = U-V}

*Ví dụ 2.7:*

Cho 2 quan hệ sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DM | ngay | maNV | maH | TenH | DVT |
|  | 20/09/1994 | 013 | A01 | Dầu lửa | lít |
|  | 20/09/1994 | 013 | A02 | Xăng | lít |
|  | 20/09/1994 | 013 | B01 | Đường trắng | kg |
|  | 20/09/1994 | 013 | B02 | Đường vàng | kg |
|  | 21/09/1994 | 014 | A01 | Dầu lửa | lít |
|  | 21/09/1994 | 014 | A02 | Xăng | lít |
|  | 21/09/1994 | 014 | B01 | Đường trắng | kg |
|  | 21/09/1994 | 014 | B02 | Đường vàng | kg |
|  | 22/09/1994 | 015 | B01 | Đường trắng | kg |
|  | 22/09/1994 | 015 | B02 | Đường vàng | kg |

|  |  |  |  |
| --- | --- | --- | --- |
| HANG | maH | tenH | DVT |
|  | A01 | Dầu lửa | lít |
|  | A02 | Xăng | lít |
|  | B01 | Đường trắng | kg |
|  | B02 | Đường vàng | kg |

Liệt kê những ngày cùng mã nhân viên mà cả 4 mặt hàng đều có bán:

KQ = DM : HANG

kết quả:

|  |  |  |
| --- | --- | --- |
| KQ | ngay | maNV |
|  | 20/09/1994 | 013 |
|  | 21/09/1994 | 014 |

**Chú ý: Một số hàm tiện ích**

***a. Hàm Sum(R, A)***

Trả về tổng các giá trị số trong thuộc tính A của quan hệ R:

Sum(R,A) = ∑ (t.A | t є R)

***b. Hàm Avg(R, A)***

Trả về trung bình cộng các giá trị trong thuộc tính A của quan hệ R:

Avg(R,A) = Sum(R,A) / Card(R) nếu Card(R) ≠ 0.

***c. Hàm Max(R, A)***

Trả về giá trị lớn nhất trong thuộc tính A của quan hệ R.

***d. Hàm Min(R, A)***

Trả về giá trị nhỏ nhất trong thuộc tính A của quan hệ R.

***e. Attr(R)***

Trả về tập thuộc tính của quan hệ R.

***f. Hàm Card(R)***

Trả về số record có trong R.

***g. Hàm Sat(t,e)***

Kiểm tra bộ t thoả điều kiện e.

**h. Một vài ký hiệu khác cho các phép toán quan hệ**

|  |  |  |
| --- | --- | --- |
| Phép toán | Ký hiệu | Ký hiệu khác |
| chọn | R(e) | σe(R) |
| chiếu | R[X] | Πx(R) |
| kết nối tự nhiên | R \* S | R ⋈ S |
| cộng | R + S | R ∪ S |
| giao | R & S | R ∩ S |
| trừ | R – S | R \ S |
| chia | R : S | R ÷ S |

## 2.2 Ngôn ngữ tân từ SQL (Structured query language)

Mỗi hệ quản trị CSDL đều phải có ngôn ngữ giao tiếp giữa người sử dụng với cơ sở dữ liệu. Ngôn ngữ giao tiếp CSDL gồm các loại sau:

*Ngôn ngữ mô tả dữ liệu* (Data Definition Language –DDL): Cho phép khai báo cấu trúc các bảng của CSDL, khai báo các mối liên hệ của dữ liệu (relatíonship) và các quy tắc áp đặt lên các dữ liệu đó.

*Ngôn ngữ thao tác dữ liệu* (Data Manipullation Language- DML) cho phép người sử dụng có thể thêm (insert), xoá (delete), sửa (update) dữ liệu trong CSDL.

*Ngôn ngữ truy vấn dữ liệu (*hay ngôn ngữ hỏi đáp có cấu trúc(Structured Query Language-SQL)): Cho phép người sử dụng khai thác CSDL để truy vấn các thông tin cần thiết trong CSDL.

*Ngôn ngữ quản lý dữ liệu* (Data Control Language- DCL): Cho phép những người quản trị hệ thống thay đổi cấu trúc của các bảng dữ liệu, khai báo bảo mật thông tin và cấp quyền khai thác CSDL cho người sử dụng.

Những năm 1975-1976, IBM lần đầu tiên đưa ra hệ quản trị CSDL kiểu quan hệ mang tên SYSTEM–R với ngôn ngữ giao tiếp CSDL là SEQUEL (Structured English Query Language). Năm 1976 ngôn ngữ SEQUEL được cải tiến thành SEQUEL-2, khoảng năm 1978-1979 SEQUEL-2 được cải tiến và đổi tên thành ngôn ngữ truy vấn có cấu trúc (Structured Query Language). Cuối năm 1979 hệ quản trị CSDL được cải tiến thành SYSTEM-R\*. Năm 1986 viện tiêu chuẩn quốc gia Mỹ (American National Standards Institute –ANSI) đã công nhận và chuẩn hoá ngôn ngữ SQL và sau đó tổ chức tiêu chuẩn thế giới (International Standards Organization -ISO) cũng đã công nhân ngôn ngữ này. Đó là chuẩn SQL-86. tới này SQL đã qua 3 lần chuẩn hoá (1989,1992,1996) để

mở rộng các phép toán và tăng cường khả năng bảo mật và tính toàn vẹn dữ liệu.

Trong chương này chúng ta chỉ nghiên cứu về ngôn ngữ SQL.

Ngôn ngữ truy vấn SQL có tập lệnh khá phong phú để thao tác trên cơ sở dữ liệu. Chẳng hạn lệnh create để tạo các bảng quan hệ, lệnh update để cập nhật dữ liệu, lệnh delete để xoá dữ liệu, lệnh insert để thêm dữ liệu,… Trong chương này, chúng tôi chỉ trình bày với bạn đọc câu lệnh quan trọng nhất của SQL đó là câu lệnh hỏi - tìm kiếm dữ liệu SELECT. Kết quả của lệnh select là một quan hệ, quan hệ kết quả này có thể kết xuất ra màn hình, máy in, hoặc là trên các thiết bị lưu trữ thông tin khác. Để đơn giản trong cách trình bày, ta xem quan hệ để thực hiện câu truy vấn là quan hệ nguồn và quan hệ kết quả của truy vấn là quan hệ đích.

Mỗi câu lệnh SQL có thể được viết trên nhiều dòng và kết thúc lệnh bởi dấu chấm phẩy (;), tuy nhiên từ khoá, tên hàm, tên thuộc tính, tên bảng, tên đối tượng thì không được phép viết tách xuống hàng. Trong vận dụng thực tế, từ khoá, tên thuộc tính, tên bảng, tên đối tượng được viết in hoa hay chữ thường là như nhau.

### 2.2.1 Tạo, cập nhật, xóa dữ liệu

**2.2.1.1. Làm việc với CSDL**

**a. Tạo CSDL**

a. Cú pháp:

##### CREATE DATABASE <tên CSDL cần tạo>

*Ví dụ 2.8*

CLOSE ALL

CREATE DATABASE people

**b. Xoá một CSDL khỏi đĩa**

Cú pháp:

##### DELETE DATABASE <tên CSDL> | ? | [DELETETABLES] [RECYCLE]

Giải thích:

<tên CSDL>: tên CSDL cần xoá và phải đảm bảo rằng nó đã được đóng. <tên CSDL> có thể chứa cả đường dẫn đến CSDL.

?: Hiện ra hộp thoại Delete để từ đó ta có thể chọn tên CSDL cần xoá khỏi đĩa.

[DELETETABLES]: Xoá đồng thời vừa CSDL vừa các bảng chứa trong đó.

[RECYCLE]: CSDL sau khi xoá sẽ được đưa vào Recycle bin.

*Ví dụ 2.9*

CREATE DATABASES qlnv

CLOSE ALL

DELETE DATABASE qlnv

**c. Tạo project**

Cú pháp:

##### CREATE PROJECT <tên project cần tạo>

*Ví dụ 2.10*

CREATE PROJECT qlnv

**2.2.1.2. Làm việc với bảng**

**a.** Tạo bảng

Cú pháp:

***CREATE TABLE | DBF <tên tập tin cần tạo>***

***(<tên trường 1> <kiểu>[(<độ dài> [,<số chữ số thập phân])] [, <tên trường 2>…])***

Ý nghĩa:

Tạo một tập tin CSDL (.DBF) với tên cột, kiểu dữ liệu,… được chỉ định trong cú pháp câu lệnh.

*Ví dụ 2.11:*

CREATE TABLE salary(empl\_id chara(5), first\_name chara(10), last\_name chara(10), salary number, dept chara(4), job\_title chara(3))

**b. Xóa bảng**

Cú pháp:

##### DROP TABLE <tên bảng> | <tên tập tin> | ? [RECYCLE]

Giải thích:

<tên bảng>: tên của bảng cần xoá khỏi CSDL hiện hành và xoá luôn khỏi ổ đĩa.

<tên tập tin>: tên của một bảng tự do cần xoá khỏi đĩa.

?: Cho hiện ra một hộp thoại Remove để từ đó ta có thể chọn ra một bảng để xoá.

[RECYCLE]: bảng không bị xoá ngay mà được đặt trong Recycle Bin.

Lưu ý:

- Khi lệnh DROP TABLE được phát ra, mọi ràng buộc toàn vẹn (RBTV) trên bảng cũng bị xoá theo. DROP TABLE cũng tác động đến các bảng khác trong CSDL hiện hành nếu các bảng đó có các RBTV hoặc các liên kết có dính dáng đến bảng bị xoá. Các RBTV và các liên kết nói trên sẽ không có hiệu lực nữa.

- Bất kỳ bảng nào bị xoá bằng lệnh này đều không thể phục hồi lại. Ngay cả khi lệnh SET SAFETY là ON, chúng ta vẫn không được nhắc nhở gì trước khi bảng bị xoá.

*Ví dụ 2.12:*

##### ***DROP TABLE salary***

**c. Sửa cấu trúc bảng**

Thêm thuộc tính vào bảng đang tồn tại sử dụng lệnh ALTER TABLE với cú pháp:

***ALTER TABLE <tên bảng> ADD <tên trường> <kiểu>***

Xoá bỏ một thuộc tính khỏi bảng đang tồn tại sử dụng lệnh ALTER TABLE với cú pháp:

***ALTER TABLE <tên bảng> DROP <tên trường cần xoá>***

Thay đổi kiểu dữ liệu của một thuộc tính:

***ALTER TABLE <tên bảng> CHANGE COLUMN <tên trường> TO <kiểu mới>***

**d. Thêm một bảng tự do vào CSDL hiện hành**

Cú pháp:

##### ADD TABLE <tên bảng> | ? [NAME <tên dài>]

Giải thích:

<tên bảng>: tên bảng cần thêm vào CSDL.

?: hiện ra một hộp thoại Open để từ đó có thể chọn ra một bảng thêm vào CSDL.

[NAME <tên dài>]: tên dài của bảng, chứa tối đa 128 ký tự và có thể được dùng để thay thế cho tên ngắn.

Lưu ý:

- Sau khi bảng được thêm vào CSDL, chúng ta có thể thực hiện cùng các thao tác như trên bất kỳ bảng nào khác.

- Nó không còn là bảng tự do mà thuộc vào một CSDL. Tuy nhiên, bất kỳ bảng nào đang ở trong CSDL đều có thể trở thành tự do nếu có lệnh REMOVE TABLE.

- Bảng mà chúng ta muốn thêm vào:

+ Phải là một tập tin DBF hợp lệ.

+ Không thể trùng tên với một bảng đã có trong CSDL đang mở hay đã tồn tại rồi trong một CSDL khác.

*Ví dụ 2.13:*

ADD TABLE salary

**2.2.1.3. CÁC LỆNH CẬP NHẬT DỮ LIỆU**

**(DML – Data Manipulation Language)**

**1. Sửa dữ liệu**

a. Cú pháp:

***UPDATE [<tên CSDL!>] <tên bảng>***

***SET <tên cột 1> = <expr1> [, <tên cột 2> = expr2>]***

***WHERE <điều kiện>***

b. Giải thích:

SET <tên cột> = <expr>: cột đang xét sẽ được thay bằng trị của biểu thức tương ứng.

<điều kiện>: những dòng thoả điều kiện trong mệnh đề WHERE sẽ được cập nhật. Còn nếu không có mệnh đề WHERE thì mọi dòng trong các cột đã được đặc tả bởi SET sẽ được cập nhật cùng một giá trị.

c. Lưu ý:

- Lệnh này chỉ có thể cập nhật các mẫu tin của một bảng duy nhất.

d. Ví dụ 2.14: Tăng lương cho những nhân viên nữ lên 1%.

UPDATE salary

SET salary = salary\*1.01

WHERE sex = .F.

**2.2.1.4. Xoá một số hoặc tất cả các dòng trong bảng quan hệ**

Cú pháp:

***DELETE FROM <tên bảng>***

***WHERE <điều kiện>***

. Giải thích:

Chỉ xoá khỏi bảng được chỉ định bởi <tên bảng> những dòng thoả <điều kiện>.

*Ví dụ 2.15*: Xoá nhân viên mang mã số 12345.

DELETE FROM salary

WHERE empl\_id = ‘12345’

**2.2.1.5. Thêm một hoặc nhiều dòng mới vào bảng quan hệ**

a. Cú pháp:

***INSERT INTO tên\_bảng (danh sách các cột)***

***VALUES (các giá trị)***

hoặc

***INSERT INTO tên\_bảng (danh sách các cột)***

***FROM ARRAY mảng | FROM MEMVAR***

b. Ý nghĩa:

Thêm một hoặc nhiều mẫu tin mới vào cuối tập tin CSDL được chỉ định bởi tên\_bảng. Giá trị của những mẫu tin mới này được liệt kê trong VALUES hoặc chứa trong mảng, biến nhớ.

*Ví dụ 2.16:*

Cho quan hệ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S | S# | SNAME | STATUS | CITY |
|  | S1 | Smith | 20 | London |
|  | S2 | Jones | 10 | Paris |
|  | S3 | Blacke | 30 | Paris |

- Sử dụng cú pháp 1:

INSERT INTO S

VALUES(“S4”, “Marry”, 15, “New York”)

- Sử dụng cú pháp 2:

DIME mang(4)

mang(1) = “S5”

mang(2) = “George”

mang(3) = “18”

mang(4) = “London”

INSERT INTO S

FROM ARRAY mang

### 2.2.2 Truy Vấn

**2.2.2.1 Cấp quyền truy xuất**

a. Cú pháp:

***GRANT <danh sách quyền> [(<danh sách thuộc tính>)]***

***ON <danh sách bảng>***

***TO <danh sách người dùng> [WITH GRANT OPTION]***

b. Giải thích:

<danh sách quyền>: Gồm có các quyền sau:

- SELECT: Gọi, đọc dữ liệu, tạo truy vấn.

- UPDATE: Cập nhật dữ liệu.

- DELETE: Xoá bộ.

- INSERT: Thêm bộ.

- INDEX: Tạo chỉ mục.

- ALTER: Hiệu chỉnh cấu trúc.

- ALL: Tất cả các quyền.

Gán <danh sách quyền> sử dụng các bảng trong <danh sách bảng> cho <danh sách người dùng>. Nếu muốn hạn chế thuộc tính sử dụng thì khai báo thuộc tính trong <danh sách thuộc tính>.

[WITH GRANT OPTION]: cho phép người dùng được giao quyền của mình cho người khác.

*Ví dụ .17:*

- Gán quyền SELECT, INSERT, UPDATE bảng SALARY cho người dùng Minh, Hung:

GRANT SELECT, INSERT, UPDATE

ON Salary

TO Minh, Hung

WITH GRANT OPTION

- Gán quyền ALTER bảng JOB cho người dùng Lan:

GRANT ALTER

ON job

TO Lan

**2.2.2.2. Hủy bỏ quyền truy xuất**

a. Cú pháp:

***REVOKE <danh sách quyền> [(<danh sách thuộc tính>)]***

***ON <danh sách bảng>***

***FROM <danh sách người dùng>***

b. Giải thích:

Tước bỏ <danh sách quyền> sử dụng <danh sách bảng> của <danh sách người dùng>.

*Ví dụ 2.18*

Hủy bỏ quyền SELECT, INSERT bảng SALARY của người dùng Minh:

REVOKE SELECT, INSERT

ON Salary

FROM Minh

**2.2.2.3 CÁC LỆNH TRUY VẤN DỮ LIỆU**

**1. Câu lệnh tìm kiếm đơn giản**

- Cú pháp:

***SELECT [<bí danh>. ]<trường> [ AS <đặt tên mới cho cột>] [,[<bí danh>.]<trường> [ AS <đặt tên mới cho cột>]…]***

FROM <table> [<bí danh>] [,<table> [<bí danh>]…]

***[[INTO <nơi kết xuất>] | [TO FILE <file> [ADDITIVE] | TO PRINTER | TO SCREEN]]***

- Giải thích:

Nếu chúng ta phát ra lệnh SET TALK ON và bắt đầu thực thi câu lệnh SELECT, Foxpro sẽ hiển thị thời gian thực hiện truy vấn và số mẫu tin có được trong kết quả. Số mẫu tin này sẽ được chứa trong biến hệ thống \_TALLY.

***<trường>:*** Có thể là một trong những mục sau:

+ Tên của một trường trong tập tin CSDL được liệt kê trong điều khoản FROM.

+ Là một hằng mà sẽ xuất hiện trên mỗi dòng của kết quả truy vấn.

+ Là một biểu thức mà có thể là tên của một hàm tự tạo.

Mỗi <trường> mà chúng ta chỉ định trong điều khoản SELECT sẽ tạo thành một cột trong kết quả truy vấn. Nếu có nhiều hơn một <trường> cùng tên thì chúng ta phải sử dụng đến <bí danh> và dấu chấm phân cách giữa <bí danh> và <trường>. <bí danh> là tên hiệu của tập tin CSDL được chỉ định trong điều khoản FROM.

Những hàm (field functions) sau đây có sẳn trong Foxpro, có thể thao tác trên một <trường> trong điều khoản SELECT mà <trường> này phải là một trường trong tập tin CSDL hoặc là một biểu thức có liên quan đến một trường trong tập tin CSDL:

+ AVG(<trường>): Tính trung bình cột dữ liệu kiểu số.

+ COUNT(<trường>): Đếm số mục chọn trong cột. COUNT(\*) là đếm số dòng dữ liệu xuất ra của truy vấn.

+ MIN(<trường>): Trả về giá trị nhỏ nhất trong <trường>.

+ MAX(<trường>): Trả về giá trị lớn nhất trong <trường>.

+ SUM(<trường>): Tính tổng cột dữ liệu kiểu số.

***[AS <đặt tên mới cho cột>]:*** Chỉ định tựa đề cho cột trong kết quả xuất ra của truy vấn. Điều này rất có lợi khi <trường> là một biểu thức hoặc chứa một hàm field function và chúng ta muốn ghi cột với một tên có ý nghĩa.

***FROM <table> [<bí danh>] [,<table> [<bí danh>]…]:*** Điều khoản FROM này liệt kê những tập tin CSDL chứa dữ liệu mà câu truy vấn cần truy lục. Nếu một trong những tập tin mà chúng ta liệt kê không được mở trong vùng làm việc trước đó hoặc không tồn tại trong thư mục hiện hành hoặc đường dẫn Foxpro thì hộp thoại Open xuất hiện. Vì vậy, chúng ta có thể chỉ định vị trí của tập tin qua hộp thoại này.

***[<bí danh>]:*** Là tên tạm thời cho tập tin CSDL. Nếu chúng ta có chỉ định bí danh cục bộ thì chúng ta phải sử dụng nó thay cho tên table trong suốt câu lệnh SELECT.

***[INTO <nơi kết xuất>]:*** Điều khoản này quyết định nơi mà kết quả truy vấn được lưu trữ. Nó có thể là một trong những mục chọn sau:

+ INTO ARRAY <mảng>: Lưu trữ kết quả truy vấn trong một mảng có tên là <mảng>.

+ INTO CURSOR <tên tập tin CSDL tạm thời>: Kết quả truy vấn sẽ được lưu trữ trên bộ nhớ RAM. Sau khi lệnh SELECT đã thi hành xong, thì tập tin tạm này ở trong tình trạng đang mở và hiện dịch nhưng chỉ đọc, chúng ta có thể thao tác trên tập tin này giống như bất kỳ tập tin CSDL nào khác. Điểm khác biệt là một khi chúng ta đã đóng tập tin tạm này lại thì nó cũng bị xóa.

+ INTO DBF <tên tâp tin CSDL> | INTO TABLE <tên tập tin CSDL>: Lưu trữ kết quả truy vấn trong một tập tin CSDL có tên <tên tập tin CSDL>. Nếu tập tin này đã tồn tại rồi thì sẽ hiển thị cảnh báo nếu như SET SAFETY ON.

***TO FILE <file> [ADDITIVE]:*** Kết quả truy vấn sẽ được chứa trong một tập tin văn bản (<file>.TXT). Nếu có thêm từ khoá ADDITIVE thì kết quả sẽ được chép nối đuôi với dữ liệu trong tập tin <file>.

***TO PRINTER:*** Đưa kết quả truy vấn ra máy in.

***TO SCREEN:*** Đưa kết quả truy vấn ra màn hình.

*- Ví dụ 2.19*

SELECT first\_name, last\_name FROM salary TO SCREEN

**2. Loại bỏ các phần tử trùng nhau**

- Cú pháp:

***SELECT [ALL | DISTINCT] [<bí danh>.]<trường> [AS <đặt tên mới cho cột>] [,[<bí danh>.]<trường> [AS <đặt tên mới cho cột>]…]***

***FROM <table> [<bí danh>] [,<table> [<bí danh>]…]***

- Giải thích:

***[ALL | DISTINCT]:*** Điều khoản SELECT chỉ định những trường, những hằng, và những biểu thức mà sẽ được hiển thị trong kết quả truy vấn. Mặc định tất cả những dòng (ALL) trong kết quả truy vấn đều được hiển thị. Tuy nhiên, nếu có thêm từ khoá DISTINCT thì sẽ loại trừ những dòng giống nhau trong kết quả truy vấn.

*- Ví dụ 2.20:*

SELECT dept FROM salary

SELECT DISTINCT dept FROM salary

**3. Tìm kiếm tất cả các phần tử**

- Cú pháp:

**SELECT \***

**FROM <table> [<bí danh>] [,<table> [<bí danh>]…]**

- Giải thích:

Hiển thị tất cả các cột của các tập tin CSDL tham gia vào truy vấn.

*- Ví dụ 2.21*:

SELECT \* FROM salary

**4. Tìm kiếm theo điều kiện**

- Cú pháp:

***SELECT [<bí danh>.]<trường> [AS <đặt tên mới cho cột>] [,[<bí anh>.]<trường> [AS <đặt tên mới cho cột>]…]***

***FROM <table> [<bí danh>] [,<table> [<bí danh>]…]***

***[WHERE <điều kiện nối kết> [AND <điều kiện nối kết>…] [AND | OR <điều kiện lọc> [AND | OR <điều kiện lọc>…]]]***

- Ý nghĩa:

WHERE cho phép truy lục dữ liệu từ nhiều tables và thỏa một điều kiện chọn lựa nào đó.

- Giải thích:

***<điều kiện nối kết>:*** Chỉ định những trường để liên kết các tables được liệt kê trong điều khoản FROM. Nếu như có nhiều hơn một table tham gia truy vấn thì chúng ta nên chỉ định điều kiện liên kết cho mỗi table.

+ Nếu có hai tables tham gia truy vấn mà chúng ta không chỉ định <điều kiện nối kết> thì cứ một record trong table đầu tiên sẽ liên kết với tất cả các record trong table thứ hai. Điều này có thể đưa đến kết quả truy vấn rất dài mà không đáp ứng được yêu cầu.

+ Nhiều <điều kiện nối kết> phải được cách nhau bởi toán tử AND hoặc OR. Mỗi <điều kiện nối kết> có dạng như sau:

*<field1> <comparision> <field2>*

Trong đó: <field1> là tên của một trường từ table 1

<field2> là tên của một trường từ table khác và <comparision> là một trong những toán tử sau:

|  |  |
| --- | --- |
| Operator | Comparision |
| =  <>, !=, #  >  >=  <  <= | Bằng  Khác nhau  Lớn hơn  Lớn hơn bằng  Nhỏ hơn  Nhỏ hơn bằng |

*Ví dụ 2.22*:

SELECT s.first\_name, s.last\_name, d.department

FROM salary s, department d

WHERE s.dept = d.deptnumb

*Ví dụ 2.23*:

SELECT salary.first\_name, salary.last\_name, job.job\_name, department.name

FROM salary, job, department

WHERE salary.dept = department.deptnumb AND

salary.job\_title = job.job\_title

***<điều kiện lọc>:*** Chỉ định tiêu chuẩn mà những records phải thỏa tiêu chuẩn này mới được chứa trong kết quả truy vấn. Một truy vấn có thể bao gồm nhiều <điều kiện lọc> và chúng cách nhau bởi toán tử AND hoặc OR. Chúng ta cũng có thể sử dụng toán tử NOT để đảo ngược giá trị của biểu thức logic hoặc sử dụng hàm EMPTY( ) để kiểm tra một trường rỗng. <điều kiện lọc> có thể là một trong những dạng sau:

*(1): <field1> <comparision> <field2>*

*(2): <field> <comparision> <expression>*

*(3): <field> <comparision> ALL (<subquery>) (Xem phần sau)*

*(4): <field> <comparision> ANY | SOME (<subquery>) (Xem phần sau)*

*(5): <field> [NOT] BETWEEN <start\_range> AND <end\_range>*

Kiểm tra giá trị của <field> có nằm (hoặc không nằm) trong khoảng chỉ định là từ <start\_range> tới <end\_range> hay không.

*(6): <field> [NOT] IN (<tâp hợp giá trị>)*

Kiểm tra giá trị của field có thuộc (hoặc không thuộc) vào một trong những giá trị được liệt kê trong <tập hợp giá trị>.

*(7): <field> [NOT] IN (<subquery>) (Xem phần sau)*

*(8): <field> [NOT] LIKE <exprC>*

Kiểm tra xem <field> có giống với <exprC> hay không. Chúng ta có thể sử dụng ký tự đại diện % hoặc \_ như một phần của <exprC>. Dấu % đại diện cho một chuỗi ký tự chưa biết, còn dấu \_ đại diện cho một ký tự chưa biết.

*- Ví dụ 2.24:*

(2): SELECT \* FROM salary WHERE salary > 30000

(7): SELECT first\_name, last\_name

FROM salary

WHERE salary IN (30000, 34000, 45000)

(5): SELECT \* FROM salary

WHERE salary BETWEEN 20000 AND 30000

(9): SELECT \* FROM salary

WHERE first\_name LIKE “%N%”

**5. Tìm kiếm có sắp xếp**

- Cú pháp:

***SELECT [ALL | DISTINCT] [<bí danh>.]<trường> [AS <đặt tên mới cho cột>] [,[<bí danh>.]<trường> [AS <đặt tên mới cho cột>]…]***

***FROM <table> [<bí danh>] [,<table> [<bí danh>]…]***

***ORDER BY <trường> [ASC | DESC] [, <trường> [ASC | DESC]…]***

- Giải thích:

Điều khoản ORDER BY này dùng để sắp xếp kết quả truy vấn dựa trên dữ liệu của một hoặc nhiều cột. Mỗi <trường> sắp xếp phải tương ứng với một cột trong kết quả truy vấn và có thể là một trong những mục sau:

+ Một mục chọn trong điều khoản SELECT hoặc là một trường trong table được liệt kê ở điều khoản FROM.

+ Một biểu thức số chỉ định vị trí của cột trong kết quả truy vấn (cột bên trái nhất là 1).

Chúng ta có thể chỉ định từ khoá DESC nếu như chúng ta muốn kết quả truy vấn sắp xếp theo chiều giảm dần. Còn từ khoá ASC là sắp xếp theo chiều tăng dần và nó cũng là giá trị mặc định.

*- Ví dụ 2.25:*

SELECT first\_name, last\_name, salary

FROM salary

ORDER BY salary

**6. Truy vấn con**

- Cú pháp:

***SELECT <các mục chọn>***

***FROM <danh sách các table tham gia truy vấn>***

***WHERE <field> <comparision> (SELECT … FROM .. . )***

- Giải thích:

<field> <comparision> (SELECT … FROM .. . ) có thể là một trong những dạng sau:

*(1): <field> [NOT] IN (<subquery>)*

Record được đưa vào kết quả truy vấn khi record đó có <field> phải thuộc một trong những giá trị do <subquery> trả về.

*Ví dụ 2.25*: Tìm những nhân viên không làm ở phòng kế toán nhưng có lương bằng với một trong những nhân viên làm ở phòng kế toán.

SELECT first\_name, last\_name

FROM salary

WHERE dept <> ‘ACCT’ AND

salary IN

(SELECT salary FROM salary WHERE dept = ‘ACCT’)

*(2): <field> <operator> ALL (<subquery>)*

Record được đưa vào kết quả truy vấn khi record đó có <field> phù hợp với tất cả những giá trị do <subquery> trả về.

Ví dụ: Tìm những nhân viên không làm ở phòng kế toán nhưng có lương cao hơn tất cả những nhân viên làm trong phòng kế toán.

SELECT first\_name, last\_name

FROM salary

WHERE dept <> ‘ACCT’ AND

salary > ALL

(SELECT max(salary) FROM salary WHERE dept=‘ACCT’)

*(3): <field> <operator> ANY | SOME (<subquery>)*

Record được đưa vào kết quả truy vấn khi record đó có <field> phù hợp với ít nhất một giá trị do <subquery> trả về.

*Ví dụ 2.26*: Tìm những nhân viên không làm ở phòng kế toán nhưng có lương cao hơn ít nhất là một nhân viên làm trong phòng kế toán.

SELECT first\_name, last\_name

FROM salary

WHERE dept <> ‘ACCT’ AND

salary > SOME

(SELECT salary FROM salary WHERE dept = ‘ACCT’)

**BÀI TẬP CHƯƠNG 2**

C©u 2.1:

R ( A B C D ) s ( C D )

a b c d c d

a b e f e f

b c e f

e d c d

e d e f

a b d e

a. Tính R x s

b. Tính R chia s

C©u 2.2:

R ={ A , B , C , D }

( A B C D )

A1 b1 c1 d1

A1 b1 c1 d2

A2 b2 c2 d2

A2 b2 c3 d3

1. Thực hiện chiếu 2 trường A, C
2. Lấy ra các bản ghi có B=b2
3. Hiện thị 2 trường A, C của các bản ghi có C=c3

C©u 2.3

R ( A B C ) s ( A B C )

A1 b1 c1 A1 b1 c1

A2 b1 c2 A2 b2 c2

A2 b2 c1

1. Tính R giao s
2. Tính R U S
3. Tính R – s

Câu 2.4 SQL

I. Tạo cơ sở dữ liệu quản lý điểm

* + 1. Tạo bảng Môn Học ( Mamon, tenmon, sotrinh, sotiet)

- Yêu cầu: mamon làm khóa chính

+ số tiết >10

* + 1. Tạo Bảng Sinh Viên(masv, hoten,ngaysinh,socmt, phai, hocbong)

- Yêu cầu: masv làm khóa chính

+ Socmt: là duy nhất

3. Tạo Bảng Kết Quả(masv,mamon,diem)

- Yêu cầu: Masv, mamon làm khóa phụ,

Điểm từ 0 đến 10

4. Tạo Bảng Kết Quả học lại(masv,mamon,diem)

- Yêu cầu: Masv, mamon làm khóa phụ,

Điểm từ 0 đến 10

II. Viết lệnh thực hiện yêu cầu sau

1. Xóa bảng Kết Quả học lại

2. Bổ xung cột Nơi sinh vào bảng sinh viên ( HD: Alter)

3. Xoá cột số tiết trong bảng Môn học

4. Chuyển kiểu cột điểm trong bảng Ketqua thành kiểu Float

5. Thêm mới 1 môn học với giá trị (‘AV’,’Anh Văn’,4)

6. Thêm mới 1 sinh viên(‘SV01’,NTT,25/01/82,’12233444’,Nữ,250000)

7. Tăng học bổng của toàn bộ SV thêm 50000

8. Cập nhật cột Thành Tiền = SL \* DG trong bảng Hoadon

9. Cập nhật số tiết môn Cơ Sở Dữ Liệu thành 90 tiết

10. Sửa thông tin sinh viên có mã SV01 gồm : hoten(MinhChau),Phai(Nu)

11. Tăng học bổng của sinh viên nữ khoa tin học thêm 10%

12. Tăng học bổng cho toàn bộ SV nữ và các SV nam ở khoa tin học

13. Xoá bỏ toàn sinh viên của khoa Cơ Khí

14. Xoá bỏ kết qủa môn Cơ sở dữ liệu của sinh viên tên là ‘Thu’

15. Xoá bỏ các sinh viên không đạt học bổng của khoa Anh Văn

16. Liệt kê kết quả học tập của sinh viên có mã SV01 gồm : masv, hoten, ngaysinh, tenmon, diem

17. Hiển thị sinh viên với giới tính là Nam hoặc Nu

18. Kết quả học tập môn Cơ sở dữ liệu. Xếp loại thep điểm thi

19. Liệt kê mã môn và điểm của sinh viên có mã A01.Sắp xếp điểm giảm dần

20. Hiển thị kết quả học tập của sinh viên gồm : mã, họ tên và ĐTB

21.Cho biết masv,hotenm diemtrungbinh của các sinh viên có điểm trung bình lớn hơn 7 điểm.

# CHƯƠNG 3: RÀNG BUỘC TOÀN VẸN DỮ LIỆU

## 3.1. Khái niệm ràng buộc toàn vẹn

Trong mỗi CSDL luôn tồn tại nhiều mối liên hệ giữa các thuộc tính, giữa các bộ; sự liên hệ này có thể xảy ra trong cùng một quan hệ hoặc trong các quan hệ của một lược đồ CSDL. Các mối liên hệ này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thoả mãn ở mọi thời điểm. Những điều kiện bất biến đó được gọi là *ràng buộc toàn vẹn.*. Trong thực tế ràng buộc toàn vẹn là các quy tắc quản lý được áp đặt trên các đối tượng của thế giới thực. Chẳng hạn mỗi sinh viên phải có một mã sinh viên duy nhất, hai thí sinh dự thi vào một trường phải có số báo danh khác nhau, một sinh viên dự thi một môn học không quá 3 lần,…

Nhiệm vụ của người phân tích thiết kế là phải phát hiện càng đầy đủ các ràng buộc toàn vẹn càng tốt và mô tả chúng một cách chính xác trong hồ sơ phân tích thiết kế - đó là một việc làm rất quan trọng. Ràng buộc toàn vẹn được xem như là một công cụ để diễn đạt ngữ nghĩa của CSDL. Một CSDL được thiết kế cồng kềnh nhưng nó thể hiện được đầy đủ ngữ nghĩa của thực tế vẫn có giá trị cao hơn rất nhiều so với một cách thiết kế gọn nhẹ nhưng nghèo nàn về ngữ nghĩa vì thiếu các ràng buộc toàn vẹn của CSDL.

Công việc kiểm tra ràng buộc toàn vẹn thường được tiến hành vào thời điểm cập nhật dữ liệu ( thêm, sửa, xoá). Những ràng buộc toàn vẹn phát sinh phải cần được ghi nhận và xử lý một cách tường minh (thường là bởi một hàm chuẩn hoặc một đoạn chương trình).

Ràng buộc toàn vẹn và kiểm tra sự vi phạm ràng buộc toàn vẹn là hai trong số những vấn đề quan trọng trong quá trình phân tích thiết kế cơ sở dữ liệu, nếu không quan tâm đúng mức đến những vấn đề trên, thì có thể dẫn đến những hậu quả nghiêm trọng về tính an toàn và toàn vẹn dữ liệu , đặc biệt là đối với những cơ sở dữ liệu lớn.

## 3.2. Các yếu tố của ràng buộc toàn vẹn

Mỗi ràng buộc toàn vẹn có bốn yếu tố: điều kiện, bối cảnh, bảng tầm ảnh hưởng và hành động phải cần thực hiện khi phát hiện có ràng buộc toàn vẹn bị vi phạm:

### 3.2.1. Điều kiện của RBTV

Điều kiện của ràng buộc toàn vẹn là sự mô tả, và biểu diễn hình thức nội dung của nó

Điều kiện của một ràng buộc toàn vẹn R có thể được biểu diễn bằng ngôn ngữ tự nhiên, ngôn ngữ đại số quan hệ, ngôn ngữ mã giả, ngôn ngữ truy vấn SQL,… ngoài ra điều kiện của ràng buộc toàn vẹn cũng có thể được biểu diễn bằng phụ thuộc hàm

Sau đây là một số ràng buộc toàn vẹn trên lược đồ CSDL ***QLSV*** .

* Mỗi lớp học phải có một mã số duy nhất để phân biệt với các lớp học khác trong trường.
* Mỗi lớp học phải thuộc về một khoa của trường.
* Mỗi sinh viên có một mã số sinh viên duy nhất, không trùng với bất cứ sinh viên nào trong trường.
* Mỗi học viên phải đăng ký vào một lớp học trong trường. Mỗi học viên chỉ được thi tối đa 3 lần cho mỗi môn học.
* Tổng số học viên của một lớp phải lớn hơn hoặc bằng số lượng đếm được của một lớp tại một thời điểm nào đó.

### 3.2.2. Bối cảnh RBTV

Bối cảnh của ràng buộc toàn vẹn là những quan hệ mà ràng buộc đó có hiệu lực hay nói một cách khác, đó là những quan hệ cần phải được kiểm tra khi tiến hành cập nhật dữ liệu. Bối cảnh của một ràng buộc toàn vẹn có thể là một hoặc nhiều quan hệ.

Chẳng hạn với ràng buộc toàn vẹn R trên thì bối cảnh của nó là quan hệ Sinhvien

***Ví dụ 3.1***

* Ràng buộc R1 có bối cảnh là quan hệ SINHVIEN.
* Ràng buộc R2 có **bối cảnh** là quan hệ NHANVIEN và PHONGBAN

### 3.2.3. Tầm ảnh hưởng RBTV

* Các thao tác cập nhật dữ liệu (thêm / xóa / sửa)
  + ******Có tác động lên các quan hệ trong bối cảnh của ràng buộc toàn vẹn
  + Cần phải được kiểm tra lại điều kiện ràng buộc.
* Xác định yếu tố Tầm ảnh hưởng bằng cách:
* Xây dựng là bảng 2 chiều
* Xác định các thao tác ảnh hưởng (+) và thao tác không ảnh hưởng (-) lên các quan hệ nằm trong bối cảnh.

### 3.2.4. Hành động khi RBTV bị vi phạm

**Hành động (phản ứng) gồm 2 phần**

* Thôngbáo: thông báo cho người dùng biết dữ liệu vi phạm RBTV nào và cần sửa lại như thế nào
* Xử lý: Đưa ra phương án xử lý khi RBTV bị vi phạm, có thể từ chối thao tác hoặc tiếp tục cho hiệu chỉnh dữ liệu

## 3.3. Phân loại ràng buộc toàn vẹn

Trong quá trình phân tích thiết kế CSDL, người phân tích phải phát hiện tất cả các ràng buộc toàn vẹn tiềm ẩn trong CSDL đó. Việc phân loại các ràng buộc toàn vẹn là rất có ích, nó nhằm giúp cho người phân tích có được một định hướng để phát hiện các ràng buộc toàn vẹn, tránh bỏ sót. Các ràng buộc toàn vẹn có thể được chia làm hai loại chính như sau:

**Thứ nhất**: Ràng buộc toàn vẹn có phạm vi là một quan hệ bao gồm: Ràng buộc toàn vẹn miền giá trị, ràng buộc toàn vẹn liên thuộc tính, ràng buộc toàn vẹn liên bộ.

Thứ hai: Ràng buộc toàn vẹn có phạm vi là nhiều quan hệ bao gồm: Ràng buộc toàn vẹn phụ thuộc tồn tại, ràng buộc toàn vẹn liên bộ - liên quan hệ, ràng buộc toàn vẹn liên thuộc tính - liên quan hệ.

Để minh hoạ cho phần lý thuyết của chương này, chúng ta xét ví dụ sau đây:

##### *Ví dụ 3.2*

Cho một CSDL C dùng để quản lý việc đặt hàng và giao hàng của một công ty. Lược đồ CSDL C gồm các lược đồ quan hệ như sau:

Q1: Khach (MAKH, TENKH, DIACHIKH, DIENTHOAI)

***Tân từ:***

Mỗi khách hàng có một mã khách hàng (MAKH) duy nhất, mỗi MAKH xác định tên khách hàng (TENKH), địa chỉ (DIACHIKH), số điện thoại (DIENTHOAI).

Q2: Hang(MAHANG,TENHANG,QUYCACH, DVTINH)

***Tân từ*:**

Mỗi mặt hàng có một mã hàng (MAHANG) duy nhất, mỗi MAHANG xác

định tên hàng (TENHANG), quy cách hàng (QUYCACH), đơn vị tính (DVTINH).

Q3: Dathang(SODH,MAHANG, SLDAT, NGAYDH, MAKH)

***Tân từ:***

Mỗi mã số đặt hàng (SODH) xác định một ngày đặt hàng (NGAYDH) và mã khách hàng tương ứng (MAKH). Biết mã số đặt hàng và mã mặt hàng thì biết được số lượng đặt hàng(SLDAT). Mõi khách hàng trong một ngày có thể có nhiều lần đặt hàng

Q4: Hoadon(SOHD, NGAYLAP, SODH, TRIGIAHD, NGAYXUAT)

***Tân từ:***

Mỗi hoá đơn tổng hợp có một mã số duy nhất là SOHD, mỗi hoá đơn bán hàng có thể gồm nhiều mặt hàng. Mỗi hoá đơn xác định ngày lập hoá đơn

(NGAYLAP), ứng với số đặt hàng nào (SODH). Giả sử rằng hoá đơn bán hàng theo yêu cầu của chỉ một đơn đặt hàng có mã số là SỌDH và ngược lại , mỗi đơn đặt hàng chỉ được giải quyết chỉ trong một hoá đơn. Do điều kiện khách quan có thể công ty không giao đầy đủ các mặt hàng cũng như số lượng từng mặt hàng như yêu cầu trong đơn đặt hàng nhưng không bao giờ giao vượt ngoài yêu cầu. Mỗi hóa đơn xác định một trị giá của nhưng các mặt hàng trong hoá đơn (TRIGIAHD) và một ngày xuất kho giao hàng cho khách (NGAYXUAT)

Q5: Chitiethd (SOHD, MAHANG, GIABAN, SLBAN)

***Tân từ****:*

Mỗi SOHD, MAHANG xác định giá bán (GIABAN) và số lượng bán (SLBAN) của một mặt hàng trong một hoá đơn.

Q6: Phieuthu(SOPT, NGAYTHU, MAKH, SOTIEN)

***Tân từ:***

Mỗi phiếu thu có một số phiếu thu (SOPT) duy nhất, mỗi SOPT xác định một ngày thu (NGAYTHU) của một khách hàng có mã khách hàng là MAKH và số tiền thu là SOTIEN. Mỗi khách hàng trong một ngày có thể có nhiều số phiếu thu.

3.3.1. Ràng buộc toàn vẹn có bối cảnh là một quan hệ

*3.3.1.1. Ràng buộc toàn vẹn liên bộ:*

+ *Ràng buộc toàn vẹn về khoá chính:*

Đây là một trường hợp dặc biệt của Ràng Buộc toàn Vẹn liên bộ, RBTV này rất phổ biến và thường được các hệ quản trị CSDL tự động kiểm tra.

##### *Ví dụ 3.3:*

Với r là một quan hệ trên lược đồ quan hệ Khach ta có ràng buộc toàn vẹn sau:

R1: t1, t2r; t1. MAKH t2. MAKH

Cuối

R1

Khach

Thêm

Sửa

Xoá

+

-

-

+ *Ràng buộc toàn vẹn về tính duy nhất*

Ví dụ: mỗi phòng ban phải có một tên gọi duy nhất

**+** Ngoài ra nhiều khi ta còn gặp những RBTV khác chẳng hạn như RBTV sau trong quan hệ sau đây.

***Ví dụ 3.4***: KETQUA(MASV,MAMH,LANTHI,DIEM)

Mỗi sinh viên chỉ được đăng thi mỗi môn tối đa là 3 lần.

*3.3.1.2. Ràng Buộc Toàn Vẹn Về Miền Giá Trị*

Ràng buộc toàn vẹn có liên quan đến miền giá trị của các thuộc tính trong một quan hệ. Ràng buộc này thường gặp. Thông thường các hệ quản trị CSDL đã tự động kiểm tra (một số) ràng buộc loại này.

##### *Ví dụ 3.5:*

Với r là một quan hệ của Hoadon ta có ràng buộc toàn vẹn sau

R3: t r

1. TRIGIAHD > 0

Cuối

R3

Hoadon

Thêm Sửa

+

Xoá

+

-

*3.3.1.3. Ràng Buộc Toàn Vẹn Liên Thuộc Tính*

Ràng buộc toàn vẹn liên thuộc tính (một quan hệ) là mối liên hệ giữa các thuộc tính trong một lược đồ quan hệ.

***Ví dụ 3.6***

Với r là một quan hệ của Hoadon ta có ràng buộc toàn vẹn sau:

R4: t r

t.NGAYLAP <= t.NGAYXUAT

Cuối

R4

Hoadon

Thêm

Sửa

Xoá

+

+

-

3.3.2. Ràng buộc toàn vẹn có bối cảnh là nhiều quan hệ

*3.3.2.1. Ràng buộc toàn vẹn về khoá ngoại:*

Ràng buộc toàn vẹn về khoá ngoại còn được gọi là ràng buộc toàn vẹn phụ thuộc tồn tại. Cũng giống như ràng buộc toàn vẹn về khoá nội, loại ràng buộc toàn vẹn này rất phổ biến trong các CSDL.

***Ví dụ 3.7***

R2. dathang[MAKH] khach[MAKH]

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Sửa | Xoá |
| dathang | + | + | - |
| Khach | - | - | + |

*3.3.2.2. Ràng Buộc Toàn Vẹn Liên Thuộc Tính Liên Quan Hệ*

Ràng buộc loại này là mối liên hệ giữa các thuộc tính trong nhiều lược đồ quan hệ.

##### *Ví dụ 3.8*

Với r,s lần lượt là quan hệ của Dathang và Hoadon. Ta có ràng buộc toàn vẹn R5 như sau:

R5: t1 r, t2 s

Nếu t1.SODH=t2.SODH thì t1.NGAYDH <= t2.NGAYXUAT

Cuối

|  |  |  |  |
| --- | --- | --- | --- |
| R5 | Thêm | Sửa | Xoá |
| Dathang | + | - | - |
| Hoandon | + | + | - |

*3.3.2.3. Ràng Buộc Toàn Vẹn Liên Bộ Liên Quan Hệ*

Ràng buộc loại này là mối liên hệ giữa các bộ trong một lược đồ cơ sở dữ liệu. Chẳng hạn như tổng số tiền phải trả trong mỗi hoá đơn (chitiethd) phải bằng TRỊ GIÁ HOÁ ĐƠN của hoá đơn đó trong quan hệ Hoadon. Hoặc số lượng học viên trong một lớp phải bằng SOHOCVIEN của lớp đó.

Ngoài ra còn có một số loại RBTV khác như :RBTV về thuộc tính tổng hợp, RBTV do tồn tại chu trình ,RBTV về giá trị thuộc tính theo thời gian.

**BÀI TẬP CHƯƠNG 3**

**Bài 3.1.Việc tổ chức kỳ thi tốt nghiệp của một khoa như sau:**

Mỗi thí sinh có một Mã số sinh viên duy nhất (MASV), mỗi MASV xác định được các thông tin: họ và tên (HOTEN), ngày sinh (NGAYSINH), nơi sinh, nữ,phái, dân tộc.

Mỗi lớp có một mã lớp (MALOP) duy nhất , mỗi mã lớp xác định các thông tin: tên lớp (TENLOP), mỗi lớp chỉ thuộc sự quản lý của một khoa nào đó. Mỗi khoa có một mã khoa duy nhất (MAKHOA), mỗi mã khoa xác định tên khoa (TENKHOA).

Mỗi thí sinh đều phải dự thi tốt nghiệp ba môn. Mỗi môn thi có một mã môn thi (MAMT) duy nhất, mỗi mã môn thi xác định các thông tin: tên môn thi (TENMT), thời gian làm bài – được tính bằng phút (PHUT), ngày thi (NGAYTHI), buổi thi (BUOITHI), môn thi này là môn lý thuyết hay thực hành (LYTHUYET). Chú ý rằng, nếu một môn học được cho thi ở nhiều hệ thì được đặt MAMT khác nhau (chẳng hạn cả trung cấp và cao đẳng ngành công nghệ thông tin đều thi môn Cơ Sở Dữ Liệu), để diễn tả điều này, mỗi mã môn học cần phải được ghi chú (GHICHU) để cho biết môn thi đó dành cho khối nào trung cấp, hay cao đẳng). Mỗi thí sinh ứng với một môn thi có một điểm thi (DIEMTHI) duy nhất, điểm thi được chấm theo thang điểm 10 và có lấy điểm lẻ đến 0.5. Một thí sinh được coi là đậu tốt nghiệp nếu điểm thi của tất cả các môn của thí sinh đó đều lớn hơn hoặc bằng 5.

Trong một phòng thi có thể có thí sinh của nhiều lớp. Trong một kỳ thi, mỗi thí sinh có thể thi tại những phòng thi (PHONGTHI) khác nhau, chẳng hạn một thí sinh thi tốt nghiệp ba môn là Cơ sở dữ liệu, Lập trình C và Visual Basic thì môn Cơ Sở Dữ Liệu và Lập Trình C thi tại phòng A3.4, còn môn thực hành Visual Basic thi tại phòng máy H6.1

Qua phân tích sơ bộ trên, ta có thể lập một lược đồ cơ sở dữ liệu như sau: THISINH(MASV, HOTEN, NGAYSINH, MALOP)

LOP(MALOP,TENLOP)

MONTHI(MAMT, TENMT, LYTHUYET, PHUT, NGAYTHI, BUOITHI, GHICHU) KETQUA(MASV, MAMT, DIEMTHI)

1. Tìm khoá cho mỗi lược đồ quan hệ trên.
2. Hãy phát biểu các ràng buộc toàn có trong cơ sở dữ liệu trên.

**Bài 3.2. Cho lược đồ cơ sở dữ liệu**

Sinhvien(MASV, HOTENSV, NU, NGAYSINH, NOISINH, TINH, MALOP) Lop(MALOP,TENLOP,MAKHOA)

Khoa(MAKHOA,TENKHOA) Monhoc(MAMH,TENMH,DONVIHT)

Giangvien(MAGV,HOTENGV,HOCVI,CHUYENNGANH,MAKHOA) Ketqua(MASV, MAMH, LANTHI,DIEMTHI)

Phancong(MALOP,MAMH,MAGV)

Hãy phát biểu các ràng buộc toàn vẹn có trong lựơc đồ cơ sở dữ liệu trên.

**Bài 3.3. Lược đồ CSDL quản lý giáo vụ**

HOCVIEN (MAHV, HO, TEN, NGSINH, GIOITINH, NOISINH, MALOP)

LOP (MALOP, TENLOP, TRGLOP, SISO, MAGVCN)

KHOA (MAKHOA, TENKHOA, NGTLAP, TRGKHOA)

MONHOC (MAMH, TENMH, TCLT, TCTH, MAKHOA)

DIEUKIEN (MAMH, MAMH\_TRUOC)

GIAOVIEN(MAGV,HOTEN,HOCVI,HOCHAM,GIOITINH,NGSINH,NGVL,

HESO, MUCLUONG, MAKHOA)

GIANGDAY(MALOP,MAMH,MAGV,HOCKY, NAM,TUNGAY,DENNGAY)

KETQUATHI (MAHV, MAMH, LANTHI, NGTHI, DIEM, KQUA)

**Xác định: điều kiện, bối cảnh, tầm ảnh hưởng**

R1: Giới tính của học viên chỉ là Nam hoặc Nữ

R2:Ngày bắt đầu (TUNGAY) giảng dạy một môn học cho một lớp luôn nhỏ hơn ngày kết thúc (DENNGAY)

R3: Tất cả các học viên phải có mã số phân biệt với nhau

R4: Các giáo viên có cùng học vị, cùng hệ số lương thì mức lương sẽ bằng nhau

R5: Học viên thi một môn học nào đó thì môn học đó phải có trong danh sách các môn học

R6: Ngày giáo viên giảng dạy một môn học phải lớn hơn hoặc bằng ngày giáo viên đó vào làm.

R7: Ngày thi một môn học phải lớn hơn ngày kết thúc học môn học đó.

R8: Sỉ số của một lớp là số lượng học viên thuộc lớp đó

# CHƯƠNG 4. CHUẨN HÓA VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU

## 4.1 Phụ thuộc hàm

**4.*1.1 Định nghĩa***

* Phụ thuộc hàm (Functional Dependancy) là một công cụ dùng để biểu diễn một cách hình thức các ràng buộc toàn vẹn.
* Phụ thuộc hàm là khái niệm quan trong nhất trong việc thiết kế cơ sở dữ liệu.
* Phụ thuộc hàm ứng dụng chủ yếu trong việc giải quyết các bài toán như: tìm khóa, tìm phủ tối thiểu, xác định dạng chuẩn…

Cho R(A1,…,An) là một sơ đồ quan hệ với tập thuộc tính U={A1,…,An}. X và Y là tập con của U. Ta nói X 🡪 Y (đọc là: X xác định hàm Y hoặc Y phụ thuộc hàm vào X) nếu với mỗi cặp bộ u, v của R sao cho:

u[X] = v[X] ⇒ u[Y] = v[Y]

***4.1.2 Cách xác định các phụ thuộc hàm cho lược đồ quan hệ***

Cách duy nhất để xác định đúng các phụ thuộc thích hợp cho một lược đồ quan hệ là xem xét nội dung tân từ của lược đồ quan hệ đó (Ý nghĩa của các thuộc tính trong LĐQH).

*Ví dụ 4.1*: Trong quan hệ Sinhvien(Masv,Hoten,Phai,Ngaysinh,Quequan,diachi)

**Phụ thuộc hàm:**

Masv → hoten

Masv,hoten → Ngaysinh,Quequan

Masv → Quequan,diachi

**Không phải phụ thuộc hàm**

Hoten→ Ngaysinh,Quequan

Ngaysinh → Masv

***4.1.3 Phân loại phụ thuộc hàm***

* Phụ thuộc hàm sơ cấp/ phi sơ cấp (không sơ cấp) (Phụ thuộc hàm toàn phần/bộ phận)

X🡪Y phi sơ cấp (bộ phận) vào X nếu ∃Z⊂X sao cho: Z🡪Y

Ngược lại: X🡪Y sơ cấp (toàn phần)

(Y phụ thuộc hàm sơ cấp vào X)

*Ví dụ 4.2:*

- MaSV,Ngaysinh🡪Tuoi ⇨ phi sơ cấp (vì Ngaysinh🡪Tuoi)

- AB🡪C ⇨ phi sơ cấp (vì A🡪C)

* **Phụ thuộc hàm trực tiếp/ gián tiếp**

*X🡪Y gián tiếp (bắc cầu) nếu ∃ Z sao cho: X🡪Z & Z🡪Y*

*Ngược lại: X🡪Y trực tiếp (Y phụ thuộc hàm trực tiếp vào X)*

*Ví dụ 4.3:*

**PHT gián tiếp**

- MaSV🡪Ngaysinh, Ngaysinh🡪Tuoi ⇒ MaSV🡪Tuoi

- AB🡪C, C🡪D ⇒ AB🡪D

* **Phụ thuộc hàm chính quy**

*X🡪Y chính quy nếu Y chỉ có một thuộc tính (Y phụ thuộc hàm chính quy vào X)*

*Ví dụ 4.4:*

- MaSV🡪Ten ⇨ chính quy

- AB🡪C ⇨chính quy

- BC🡪EM ⇨ không chính quy

## 4.2 Các mệnh đề và tiên đề Amstrong

***4.2.1.*** ***Định nghĩa hệ tiên đề Armstrong***:

Gọi R(U) là lược đồ quan hệ với U = {A­1,…,An} là tập các thuộc tính. X, Y, Z, W ⊆ U. Hệ tiên đề Armstrong bao gồm:

F1) Tính phản xạ:

Y ⊆ X ⇒ X 🡪 Y

F2) Tính bắc cầu:

X 🡪 Y, Y 🡪 Z ⇒ X 🡪

F3) Tính mở rộng hai vế:

X 🡪 Y ⇒ (Z ⊆ U) XZ 🡪 YZ

**4.2.2. Định nghĩa tính đầy đủ và tính chặt chẽ của một hệ tiên đề:**

Cho tập các phụ thuộc hàm F:

- Một hệ tiên đề S cho các phụ thuộc hàm được gọi là đầy đủ nếu các luật của S cho phép suy diễn ra mọi phụ thuộc hàm trong F+.

- Một hệ tiên đề S cho các phụ thuộc hàm được gọi là chặc chẽ nếu các luật của S không cho phép suy diễn từ F bất kỳ phụ thuộc hàm nào không thuộc F+.

**4.2.3 Bổ đề 1:**

Hệ tiên đề Armstrong là đúng. Có nghĩa là F là tập các phụ thuộc hàm đúng trên quan hệ R. Nếu X 🡪 Y là một phụ thuộc hàm được suy dẫn từ F nhờ hệ tiên đề Armstrong thì X 🡪 Y là đúng trên quan hệ R.

**4.2.4 Bổ đề 2:**

Từ hệ tiên đề Armstrong suy ra được một số luật sau đây:

F4) Cộng tính ở vế phải:

X 🡪 Y, X 🡪 Z ⇒ X 🡪 YZ

F5) Tính tựa bắc cầu:

X 🡪 Y, YZ 🡪 W ⇒ XZ 🡪 W

F6) Luật tách:

X 🡪 Y, Z ⊆ Y ⇒ X 🡪 Z

**4.2.5 Bổ đề 3:**

X 🡪 Y suy dẫn từ hệ tiên đề Armstrong khi và chỉ khi Y ⊆ X+ (X+ là bao đóng của tập các thuộc tính - giới thiệu ở mục sau).

**4.2.6 Định lý 1:**

Hệ tiên đề Armstrong là chặt chẽ và đầy đủ.

*Ví dụ 4.5***:** Cho tập phụ thuộc hàm F = {A 🡪 B, B 🡪 CD} ta chứng minh phụ thuộc hàm AC 🡪 CD được suy diễn logic từ F.

Thật vậy:

F3: A 🡪 B ⇒ AC 🡪 BC

F3: B 🡪 CD ⇒ BC 🡪 CD

F2: AC 🡪 BC, BC 🡪 CD ⇒ AC 🡪 CD

## 4.3 Bao đóng (CLOSURE)

***4.3.1. Định nghĩa suy diễn theo logic***

Giả sử F là tập phụ thuộc hàm trên lược đồ quan hệ R(U) và X, Y là các tập con thuộc tính của R (X, Y ⊆ U). Ta nói rằng F suy diễn logic phụ thuộc hàm X 🡪 Y hay phụ thuộc hàm X 🡪 Y được suy diễn logic từ F, ký hiệu:

F |== X 🡪 Y

Nếu mọi quan hệ r thoả các phụ thuộc hàm trong F cũng thoả phụ thuộc hàm X 🡪 Y.

Nói cách khác, X 🡪 Y được suy diễn logic từ F nếu từ F áp dụng một số lần các tính chất F1, F2, F3 ta thu được X 🡪 Y.

Ví dụ 4.5:

Cho tập phụ thuộc hàm F = {A 🡪 B, B 🡪 CD} ta chứng minh phụ thuộc hàm AC 🡪 CD được suy diễn logic từ F.

Thật vậy:

F3: A 🡪 B ⇒ AC 🡪 BC

F3: B 🡪 CD ⇒ BC 🡪 CD

F3: AC 🡪 BC, BC 🡪 CD ⇒ AC 🡪 CD

***4.3.2 Bao đóng của tập các phụ thuộc hàm***

***a. Định nghĩa:***

Cho tập phụ thuộc hàm F trên tập thuộc tính U. Bao đóng của F, ký hiệu là F+, là tập nhỏ nhất các phụ thuộc hàm trên U thoả:

F+ = {X 🡪 Y | F |== X 🡪 Y}

***b. Định nghĩa khác cho bao đóng của tập phụ thuộc hàm:***

F+ là tập các phụ thuộc suy diễn từ F nhờ hệ tiên đề Armstrong. Tức nó phải thoả hai tính chất sau:

F+ ⊇ F

Khi áp dụng các tính chất F1, F2, F3 cho F+ ta không thu được phụ thuộc hàm nào nằm ngoài F+.

***c. Tính chất:***

(1): Tính phản xạ: F+ ⊇ F

(2): Tính đơn điệu: F ⊆ G ⇒ F+ ⊆ G+

(3): Tính lũy đẳng: (F+)+ = F+

(4): (FG)+ ⊇ F+G+

(5): (F+G)+ = (FG+)+ = (FG)+

*Ví dụ: 4.6* Cho lược đồ quan hệ Q(ABCDEGH) và tập phụ thuộc hàm:

F = {B → A; DA→ CE; D → H; GH→ C; AC→ D }

Khi đó bao đóng của phụ thuộc hàm F là F+

F+ = { B → A; DA→ CE; D → H; GH→ C; AC→ D ; *BC* → *AC; BC* → *D; DA* → *AH; DG* → *C;BC* → *AD;….}*

***4.3.3. Bao đóng của tập thuộc tính***

***a. Các loại thuộc tính***

*Thuộc tính đơn (simple attribute) là thuộc tính không bị phân rã thành nhiều thuộc tính khác.*

*Thuộc tính phức hợp (composite attribute) là thuộc tính bị phân rã thành nhiều thuộc tính khác.*

*Thuộc tính đơn trị (single-valued attribute) là thuộc tính chỉ chứa một giá trị.*

*Thuộc tính đa trị (multivalued attribute) là thuộc tính chứa nhiều giá trị khác nhau thuộc một miền trị.*

***b. Định nghĩa bao đóng***

Cho tập phụ thuộc hàm F trên tập thuộc tính U và X ⊆ U. Bao đóng của tập thuộc tính X (đối với F), ký hiệu X+, là tập sau:

X+ = {A | X 🡪 A ∈ F+}

***c. Định nghĩa khác cho bao đóng của tập thuộc tính:***

X+ là tập các thuộc tính A sao cho X 🡪 A có thể suy diễn được từ F bằng hệ tiên đề Armstrong.

***d. Tính chất:***

(1): Tính phản xạ: X+ ⊇ X

(2): Tính đơn điệu: X ⊆ Y ⇒ X+ ⊆ Y+

(3): Tính lũy đẳng: (X+)+ = X+

(4): (XY)+ ⊇ X+Y+

(5): (X+Y)+ = (XY+)+ = (XY)+

(6): X 🡪 Y ⇒ Y ⊆ X+

(7): X 🡪 Y ⇒ Y+ ⊆ X+

(8): X 🡪 X+ và X+ 🡪 X

(9): X+ = Y+ ⇒ X 🡪 Y, Y 🡪 X

**e. Thuật toán tìm bao đóng**

Tính toán bao đóng của tập phụ thuộc hàm F+ rất tốn kém thời gian vì F+ có thể rất lớn ngay khi bản thân F nhỏ. Xét tập hợp:

F = {A 🡪 B1, A 🡪 B2,…, A 🡪 Bn}

Khi đó F+ bao gồm tất cả các phụ thuộc hàm dạng A 🡪 Y, trong đó Y là tập con của:

{B1, B2,…, Bn}

Bởi vì có đến 2n tập con như vậy nên danh sách F+ sẽ rất lớn dù cho n có nhỏ đi nữa.

Nhưng việc tính bao đóng của tập thuộc tính X+ không tốn kém lắm, thời gian tính toán tỉ lệ thuận với độ dài của tất cả phụ thuộc hàm trong F. Theo kết luận từ các định lý việc xác định X 🡪 Y ∈ F+ tương đương với việc tính bao đóng X+.

Sau đây là thuật toán tính bao đóng tập thuộc tính X+:

*Giải thuật:*

Input: Một tập hữu hạn các thuộc tính U, một tập các phụ thuộc hàm trên U, và một tập X ⊆ U.

Output: X+ thoả F.

Method:

(i): Đặt X(0) = X, F0 = F

(ii): Biết X(k-1), tính X(k) như sau:

- Đặt X(k) = X(k+1)

- Lần lượt duyệt qua các phụ thuộc hàm Y 🡪 Z trong F0, với Y⊂ X(i). Ta mở rộng X(k) = X(k) ∪ Z và đặt F0 = F0 \ {Y 🡪 Z}

Nếu X(k) = U hoặc F0 = ∅, thì kết thúc.

(iii): Nếu X(k) = X(k-1) thì thuật toán dừng. Ngược lại, tăng k lên 1 và quay lại bước (ii).

*Ví dụ 4.7:*

Cho tập phụ thuộc hàm F:

AB 🡪 C D 🡪 EG

C 🡪 A BE 🡪 C

BC 🡪 D CG 🡪 BD

ACD 🡪 B CE 🡪 AG

Đặt X = BD. Áp dụng giải thuật trên để xác định bao đóng X+.

- Đặt X(0) = X = BD, F0 = F

- Tính X(1): Ta tìm các phụ thuộc hàm có phía bên trái là B, D, hoặc BD, có:

D 🡪 EG

nên ta thêm E và G vào X(0) để có:

X(1) = BDEG

Tương tự có:

X(2) = BCDEG

X(3) = ABCDEG = U 🡪 thoát.

Vậy: (BD)+ = ABCDEG

*Ví dụ 4.8 :* Cho lược đồ quan hệ Q(A,B,C,D,E,G,H) và tập phụ thuộc hàm:

F={B→A; DA→CE; D→H; GH→ C; AC→D}.

Tìm bao đóng của X = {AC} trên F

X(0) = {A,C} , {có AC→D}

X(1) = {A,C,D}, { có AD→CE}

X(2) = {A,C,D,E}, {có D→H}

X(3) = {A,C,D,E,H}

X+= X(3)

## 4.4. Sự tương đương của phụ thuộc hàm

Định Nghĩa: *Hai tập phụ thuộc hàm F và G là tương đương (Equivalent) nếu F+ = G+*

Ký hiệu F≈ G

Mệnh đề 1: F ⊂ G+ ⇔ F+ ⊂ G+

Mệnh đề 2: Mỗi tập phụ thuộc hàm F tương đương với tập phụ thuộc hàm G gồm các phụ thuộc hàm mà vế phải chỉ có 1 thuộc tính.

Kiểm tra tính tương đương của 2 tập phụ thuộc hàm:

B1: Với mỗi Y🡪Z∈F & Z⊆ Y+ (trên G) thì Y🡪Z∈G+

Nếu ∀t∈F & f∈G+ thì F+ ⊆ G+

B2: Tương tự,nếu ∀t∈G & f∈F+ thì G+⊆ F+

B3: Nếu F+⊆G+ và G+ ⊆ F+ thì F ≈ G

**Hay**

**Thuật toán** xác định F và G có tương đương không?

***Bước 1***: Với mỗi phụ thuộc hàm X→Y của F ta xác định xem X→Y có là thành viên của G không

***Bước 2***: Với mỗi phụ thuộc hàm X→Y của G ta xác định xem X→Y có là thành viên của F không

Nếu cả hai bước trên đều đúng thì F ≈ G

*Ví dụ 4.9:*

Cho lược đồ quan hệ Q(ABC) hai tập phụ thuộc hàm:

F={ A→B; A→C; B→A; C→A; B→C}

G={ A→B; C→A; B→C}

**F có tương đương với G không?**

**Tìm các thuộc tính VT của F: A, B, C**

A+G =ABC ⊃ A, C

BG+= BCA ⊃ A, C ⬄

C+G =CAB ⊃ A

**Tìm các thuộc tính VT của G: A, C, B**

A+F =ABC ⊃ B

BF+= BAC ⊃ C ⬄

C+F =CAB ⊃ A

*Vậy 2 tập phụ thuộc hàm F, G tương đương*

## 4.5. Phụ thuộc hàm tối thiếu

***4.5.1. Phủ, phụ thuộc hàm tương đương:***

Cho lược đồ quan hệ R và các tập phụ thuộc hàm F và G trên R ta nói:

F *phủ phụ thuộc hàm* G nếu G+ ⊂ F+

F *tương đương phụ thuộc hàm* G nếu G+ = F+

Để xác định phụ thuộc hàm Y 🡪 Z ∈ G+ hay không ta sử dụng thuật toán tính bao đóng tập thuộc tính để tính Y+ đối với G và kiểm tra xem Z ⊂ Y+ hay không.

***Mệnh đề:*** F ⊂ G+ ⇔ F+ ⊂ G+

***Mệnh đề:*** Mỗi tập phụ thuộc hàm F tương đương với tập phụ thuộc hàm G gồm các phụ thuộc hàm mà vế phải chỉ có 1 thuộc tính.

***4.4.2. Phủ tối tiêu:***

Để tối ưu hơn nữa việc thiết kế lược đồ CSDL quan hệ ta yêu cầu mạnh hơn đối với tập phụ thuộc hàm tương đương.

***Định nghĩa***: Tập phụ thuộc hàm F gọi là phụ thuộc hàm tối thiểu nếu nó thoả mãn các điều kiện sau:

(1): Vế phải của mỗi phụ thuộc hàm trong F chỉ có 1 thuộc tính.

(2): Mọi phụ thuộc hàm X 🡪 A ∈ F là quan trọng, tức là tập phụ thuộc hàm có từ F bằng sự loại bỏ phụ thuộc hàm X 🡪 A:

F \ {X 🡪 A}

không tương đương với F.

(3): Với mỗi phụ thuộc hàm X 🡪 A ∈ F, mọi thuộc tính B ∈ X đều quan trọng, tức là tập phụ thuộc hàm có từ F bằng việc thay phụ thuộc hàm X 🡪 A bởi phụ thuộc hàm (X \{B}) 🡪 A:

(F \ {X 🡪 A}) ∪ {X \ {B} 🡪 A}

không tương đương với F.

Nhận xét: Điều kiện (2) đảm bảo không có phụ thuộc hàm dư thừa, điều kiện (3) đảm bảo không có thuộc tính ở vế trái dư thừa.

***Thuật toán tìm phủ tối thiểu***:

- *Input*: Tập phụ thuộc hàm F.

- *Output*: Tập phụ thuộc hàm tối thiểu G tương đương với F.

- *Method*:

(1): Phân rã vế phải tất cả phụ thuộc hàm của F và gọi G là tập tất cả các phụ thuộc hàm thu được.

(2): Loại các phụ thuộc hàm dư thừa trong G: Không tồn tại X 🡪 A nào trong F mà tập F - {X 🡪 A} tương đương với F.

(3): Loại các thuộc tính dư thừa ở vế trái của các phụ thuộc hàm trong G: Không tồn tại X 🡪 A trong F mà Z ⊂ X để cho:

(F - {X 🡪 A}) ∪ {Z 🡪 A}

tương đương với F.

***Ví dụ 4.10:***

Cho lược đồ R = (A, B, C, D, E, G) và tập phụ thuộc hàm F gồm các phụ thuộc hàm sau:

AB 🡪 C D 🡪 EG

C 🡪 A BE 🡪 C

BC 🡪 D CG 🡪 BD

ACD 🡪 B CE 🡪 AG

Ta áp dụng thuật toán tính phủ tối thiểu để tính phủ tối thiểu của F.

(1): Phân rã vế phải các phụ thuộc hàm trong F

Tập G thu được gồm các phụ thuộc hàm:

AB 🡪 C BE 🡪 C

C 🡪 A CG 🡪 B

BC 🡪 D CG 🡪 D

ACD 🡪 B CE 🡪 A

D 🡪 E CE 🡪 G

D 🡪 G

(2): Loại các phụ thuộc hàm dư thừa và thuộc tính dư thừa

\* Loại các phụ thuộc hàm dư thừa:

- Loại CG 🡪 B, vì nó suy ra từ C 🡪 A, ACD 🡪 B và CG 🡪 D bằng các phép kéo theo như sau:

C 🡪 A ⇒ CG 🡪 AG (qui tắc mở rộng hai vế)

⇒ CG 🡪 A (qui tắc phân rã)

CG🡪A & CG🡪D & CG 🡪 C ⇒ CG 🡪 ACD (qui tắc hợp)

CG 🡪 ACD & ACD 🡪 B ⇒ CG 🡪 B (qui tắc bắc cầu)

Như vậy ta loại ra khỏi G.

- Loại CE 🡪 A, vì nó suy ra từ C 🡪 A.

Kết thúc bước này các phụ thuộc hàm còn lại như sau:

AB 🡪 C BE 🡪 C

C 🡪 A

BC 🡪 D CG 🡪 D

ACD 🡪 B

D 🡪 E CE 🡪 G

D 🡪 G

\* Loại thuộc tính dư thừa:

Trong phụ thuộc hàm ACD 🡪 B, thuộc tính A dư thừa vì C 🡪 A. Như vậy ta thay ACD 🡪 B bởi CD 🡪 B.

Cuối cùng ta nhận được phủ tối thiểu:

AB 🡪 C BE 🡪 C

C 🡪 A

BC 🡪 D CG 🡪 D

CD 🡪 B

D 🡪 E CE 🡪 G

D 🡪 G

## 4.6 Khóa và giải thuật tìm khóa

**4.6.1. Khái niệm khoá**

Cho R={A1,A2,A3…An}. Cho K là tập con của R

Cho r là quan hệ trên R. t1,t2 là hai bộ bất kỳ

Khi đó: K gọi là khoá nếu như t1 ≠t2 thì t1[k] ≠ t2[k]

K là siêu khoá khi: ∃K’ ⊆ K

Khái niệm theo phụ thuộc hàm

Cho R={A1,A2,A3…An}. Cho K ⊆ R

Cho F là tập phụ thuộc hàm

Khi đó : K là khoá nếu:

(1): K 🡪 R

(2): ∃K’ ⊆ K

**4.6.2 Giải thuật tìm khoá**

***Cách 1: Ứng dụng: cho phép xác định được khoá của một quan hệ dựa trên tập phụ thuộc hàm***

\* Bài toán:

Cho một lược đồ R có các thuộc tính {A1,A2,...An} và tập Phụ thuộc hàm F.

Yêu cầu: Tìm khoá của R dựa trên F

**Bước 1**:

gán K = Q+

**Bước 2:**

A là một thuộc tính của K, đặt K’ = K - A. Nếu K’+= Q+ thì gán K = K' thực hiện lại **bước 2**

***Nếu muốn tìm các khóa khác (nếu có) của lược đồ quan hệ, ta có thể thay đổi thứ tự loại bỏ các phần tử của K.***

*Ví dụ 4.11* Cho lược đồ quan hệ R và tập phụ thuộc hàm F như sau:

* + R(A,B,C,D,E)
  + F={AB→C, AC → B, BC → DE} tìm khóa K

B1: K=R+ ⇒ K=ABCDE

B2:(K\A)+ ⇒(BCDE)+=BCDE ≠ R+ ⇒ K=ABCDE

B3:(K\B)+ ⇒(ACDE)+= ABCDE = R+ ⇒ K=ACDE

B4: (K\C)+ ⇒(ADE)+ = ADE ≠ R+ ⇒ K=ACDE

B5: (K\D)+ ⇒ (ACE)+ = ACEBD=R+ ⇒ K=ACE

B6: (K\E)+ ⇒(AC)+ = ACBDE =R+ ⇒ **K=AC**

***Cách 2: phương pháp đường chạy nhị phân.***

Sau đây là thuật toán tìm tất các các kháo của lược đồ quan hệ

- Gọi A là tập thuộc tính mà không có ở vế phải của bất kỳ phụ thuộc hàm nào. Tập thuộc tính đó ký hiệu là: N

- Gọi B là tập thuộc tính mà không có ở vế trái của bất kỳ phụ thuộc hàm nào. Tập thuộc tính đó ký hiệu là: D

Tập thuộc tính còn lại có cả 2 vế của các phụ thuộc hàm ta ký hiệu: N

🡺 Nhận xét: Nếu K là khóa thì K chứa tất cả các thuộc tính nguồn và không chứa bất kỳ thuộc tính đích nào.

**Method:**

Bước 1: Xây dựng 2v tập con của L: L1,L2… bằng phương pháp đường chạy nhị phân.

Bước 2: Xây dựng tập K chứa các siêu khóa

🡪 K = Rỗng

🡪 Với mọi Li: Xi = N hợp Ni

🡪 Tính X+F. Nếu X+F= U thì K= K hợp Xi

Bước 3 : Loại bỏ dần các siêu khóa lớn

*Ví dụ 4.12 :*

Cho R(ABCDEG) với các phụ thuộc hàm sau

F={ AE🡪C, CG🡪A, BD🡪G, GA🡪E}. Xác định tất khóa của R

Hướng dẫn

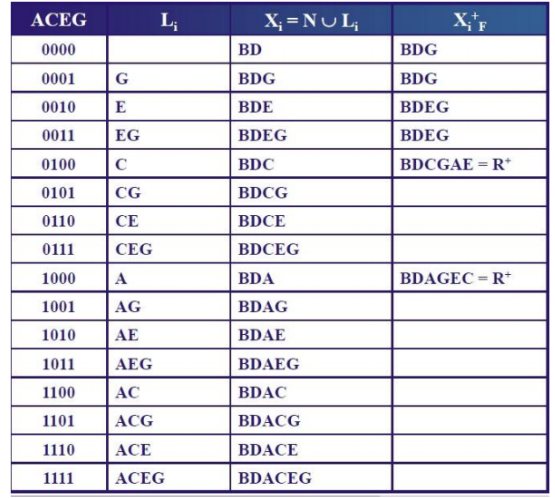
Ta có

N ={B,D}

D = Rỗng

L ={A,C,E,G}

Xây dựng tập thuộc tính nhị phân bằng phương pháp chạy nhị phân



KL: 2 siêu khóa: BDC, BDA

Có 2 khóa: BDC, BDA

## 4.7. Chuẩn hóa lược đồ quan hệ

**4.7.1 Khái niệm chuẩn hóa**

Là quá trình tách một lược đồ quan hệ thành một tập các lược đồ con, sao cho quá trình tách là không mất thông tin và các lược đồ con là tối ưu hơn lược đồ ban đầu theo nghĩa: hạn chế dư thừa dữ liệu, thuận lợi cho các quá trình tạo lập, cập nhật và các thao tác tìm kiếm khác.

**4.7.1.1. Đặt vấn đề**

Khi thiết kết CSDL quan hệ ta thường đứng trước vấn đề lựa chọn giữa các lược đồ quan hệ: lược đồ nào tốt hơn? Tại sao? Mục này sẽ nghiên cứu một số tiêu chuẩn đánh giá lược đồ quan hệ và các thuật toán giúp chúng ta xây dựng được lược đồ CSDL quan hệ có cấu trúc tốt.

Có thể nói tổng quát một lược đồ quan hệ có cấu trúc tốt là lược đồ không chứa đựng các vấn đề liệt kê sau đây:

**4.7.1.2 Các vấn đề nảy sinh đối với một CSDL không tốt**

***a.*** ***Dư thừa dữ liệu***

Là sự trùng lặp thông tin trong CSDL.

Ví dụ: Xét lược đồ quan hệ NHACC(Ten\_NCC, Hang, DonGia, Diachi\_NCC).

Nếu một nhà cung cấp cung cấp nhiều mặt hàng thì địa chỉ nhà cung cấp phải lặp lại nhiều lần 🡪 kéo theo dư thừa dữ liệu.

Ngoài việc gây lãng phí dung lượng lưu trữ, sự dư thừa dữ liệu có thể gây ra những hậu quả nghiêm trọng đối với dữ liệu khi người dùng cập nhật dữ liệu làm cho dữ liệu không tương thích, thiếu nhất quán.

Ví dụ: Xét lại lược đồ NHACC trên. Ta có thể sửa địa chỉ một nhà cung cấp tại một bộ nào đó mà không sửa ở một bộ khác gây ra địa chỉ không nhất quán của cùng một nhà cung cấp.

***b. Dị thường do thêm dữ liệu***

Không thể chèn bộ mới vào quan hệ nếu không có đầy đủ dữ liệu.

Ví dụ: Ta không thể ghi nhận địa chỉ một nhà cung cấp nếu nhà cung cấp đó không cung cấp mặt hàng nào cả vì Ten\_NCC, Hang tạo thành một khoá cho quan hệ.

***c. Dị thường do xóa dữ liệu***

Ngược lại với vấn đề c, khi ta xoá hết các mặt hàng do một hãng cung cấp, ta không thể theo dõi được địa chỉ của hãng đó.

Trong ví dụ trên, vấn đề này sẽ không còn nữa khi ta thay quan hệ NHACC bằng hai quan hệ:

NCC\_ĐC(TÊN\_NCC, ĐC\_NCC)

NCC\_HG(TÊN\_NCC, TÊN\_HG, GIA)

Tuy nhiên vẫn còn tồn tại nhiều vấn đề, ví dụ vẫn còn một bất lợi trong cách phân chia sơ đồ trên: khi muốn tìm địa chỉ hãng cung cấp mặt hàng bột ngọt Vedan, ta phải làm một kết nối, trong khi chỉ với quan hệ NHACC ta chỉ đơn giản thực hiện phép chọn và chiếu. Như vậy làm sao xác định sự thay thế như trên là có lợi? Có xẩy ra các vấn đề như đã bàn ở trên trong hai quan hệ này không? Làm thế nào tìm được một sự thay thế tốt đẹp cho một sơ đồ quan hệ tồi tệ?

***4.7.2 Dạng chuẩn 1 NF*** **(1NF- Fist normal form)**

***a. Định nghĩa:***

Một sơ đồ quan hệ R được xem là ở dạng chuẩn thứ nhất nếu mọi thuộc tính của R đều khác trống, phụ thuộc hàm vào khoá và không được mang giá trị kép (tức không thể chia được thành các thành phần nhỏ hơn và có ý nghĩa).

***b. Ví dụ:***

Cho sơ đồ quan hệ:

CUNG\_UNG(MaNSX, MaH, SL, VonNSX, TP, Nuoc)

với các phụ thuộc hàm:

MaNSX, MaH 🡪 SL (a)

MaNSX 🡪 VonNSX (b)

MaNSX 🡪 TP (c)

MaNSX 🡪 Nuoc (d)

TP 🡪 Nuoc (e)

Sơ đồ này thoả dạng chuẩn thứ nhất (1NF).

**c.** **Quy tắc chuyển bảng từ dạng chưa chuẩn về chuẩn 1 NF**

Nguyên tắc chung: Loại bỏ thuộc tính lặp hoặc đa trị

**Các bước thực hiện:**

* Tách nhóm thuộc tính lặp/ đa trị sang một bảng mới.
* Khóa của bảng mới là khóa của bảng ban đầu và khóa nhóm lặp.
* Bảng còn lại là bảng gồm có khóa và các thuộc tính còn lại.
  + Ví dụ: cho quan hệ R( **A1**, A2, A3, {A4, A5, A6})
* A1 là khóa chính, {A4, A5, A6} thuộc tính lặp
* Quan hệ R được tách thành:
  + R1(**A1**, A2, A3)
  + R2 (**A1**, A4, A5, A6)

**4.7.3 Dạng chuẩn thứ hai (2NF)**

***a. Định nghĩa***

Cho lược đồ quan hệ R, lược đồ R được gọi là ở dạng chuẩn 2 (ký hiệu là 2 NF) nếu nó ở **dạng chuẩn 1NF** và **các thuộc tính không khóa của R là phụ thuộc hàm đầy đủ vào khóa chính.**

***Ví dụ 4.13:***

Sơ đồ quan hệ SAIP (khoá SI) với các phụ thuộc hàm: SI 🡪 P và S 🡪 A đã vi phạm dạng chuẩn thứ 2. Do khoá là SI nên A là không nguyên tố.

**b**. **Thuật toán kiểm tra dạng chuẩn 2NF**

**Bước 1:** Tìm tất cả các khóa của R

**Bước 2:** Với tất cả các khóa K, tìm bao đóng của tất cả các tập con thật sự S của K.

**Bước 3:** Nếu có bao đóng S+ **chứa** thuộc tính **không khóa** thì R không đạt chuẩn 2NF ngược lại R đạt được dạng chuẩn 2NF.

*Ví dụ 4.14* Cho lược đồ quan hệ R(A, B, C, D) và tập phụ thuộc hàm: F={AB→C, B→D}.

Hỏi R có đạt chuẩn 2NF hay không?

**Bước 1: Tìm tất cả các khóa của R**

Hai khóa AB

**Bước 2: Tìm bao đóng của tất cả các tập con của khóa**

B+, A+ ta có B+=BD, A+=A

**Bước 3: Liệt kê thuộc tính không khóa: D**

Thấy tập con B+ chứa thuộc tính không khóa D

=> R không đạt chuẩn 2NF.

**c. Quy tắc chuẩn hóa từ 1NF-2NF**

Bước 1: Loại bỏ các thuộc tính không khóa phụ thuộc vào một bộ phận khóa chính và tách thành ra một bảng riêng, khóa chính của bảng là bộ phận khóa mà chúng phụ thuộc vào.

Bước 2: Các thuộc tính còn lại lập thành một quan hệ, khóa chính cuả nó là khóa chính ban đầu

*Ví dụ 4.15:*

Cho lược đồ quan hệ R(A, B, C, D) và tập phụ thuộc hàm: F={AB→C; B→D} không đạt chuẩn 2NF

R1( B,D) F1={ B→D}

R2( A,B,C) F2={AB→C;}

**4.7.4 Dạng chuẩn thứ ba(3NF)**

**a. Định nghĩa**

Cho lược đồ quan hệ R, lược đồ R được gọi là ở dạng chuẩn 3 (ký hiệu là 3NF) nếu thỏa mãn:

* Dạng chuẩn 2 NF
* Mọi thuộc tính không khóa không phụ thuộc hàm bắc cầu vào khóa chính.

(tức là tất cả các thuộc tính phải được suy ra trực tiếp từ khóa).

Hay

Cho lược đồ quan hệ R lược đồ R được gọi là ở dạng chuẩn 3, (ký hiệu là 3NF), nếu như mọi phụ thuộc hàm X→ A Î F+ với A Ï X thì:

- Hoặc X là siêu khóa.

- Hoặc A là thuộc tính khóa.

*Ví dụ 4.16:*

Cho lược đồ quan hệ R(ABCD)   
F=(AB → C ; D → B; C → ABD)

K1=[AB]; K2=[AD];K3=[C] là các khoá,

R không có thuộc tính không khoá nên R đạt chuẩn 3NF

*Ví dụ 4.17:*

Cho lược đồ quan hệ sau. R(NGPM) F={NGP→M; M→P}

Khóa R {NGP}, {NGM}

NGP → M có vế trái là siêu khoá   
M → P có vế phải là thuộc tính khoá.

R đạt chuẩn 3 NF

**b. Thuật toán kiểm tra chuẩn 3 NF**

**Bước 1:** Tìm tất cả khóa của R.

**Bước 2:** Từ F tạo tập phụ thuộc hàm tương đương F1 có vế phải một thuộc tính.

**Bước 3:** Nếu mọi phụ thuộc hàm X→ AÎ F1 với A Ï X đều có

* Hoặc X là siêu khóa
* hoặc A là một thuộc tính khóa

->R đạt chuẩn 3NF ngược lại R không đạt chuẩn 3NF

*Ví dụ 4.18*

Cho lược đồ quan hệ R(A, B, C, D)

F={ AB→C; D→B; C→ABD }. Hỏi lược đồ có ở dạng chuẩn 3NF?

Giải:

**B1:** Khóa chính là: AB,C, AD

=> THUỘC TÍNH KHÓA LÀ A, B, C, D

**B2:** Phân rã vế phải PTH về giá trị nguyên tố

F1= {AB->C, D->B, C->A, C->B, C->D}

**B3:** Do vế phải của PTH đều là thuộc tính khóa

-> lược đồ đạt chuẩn 3NF

*Ví dụ 4.19*

Cho lược đồ quan hệ Q(A, B, C, D)

F={ AB→C; AB→D; C→ABD }. Hỏi lược đồ có ở dạng chuẩn 3NF?

Giải:

B1:Khóa của lược đồ là: AB, C

Thuộc tính không khóa là D, thuộc tính khóa là A, B, C

B2: phân rã vp về giá trị nguyên tố:

F1= {AB->C, AB->D, C->ABD}

B3: Do vế trái của PTH là siêu khóa

-> LƯỢC ĐỒ ĐẠT CHUẨN 3NF

**c. Quy tắc chuẩn hóa từ 2NF - 3NF**

**Bước 1:** Loại bỏ các thuộc tính phụ thuộc bắc cầu ra khỏi quan hệ và tách chúng thành một quan hệ riêng có khóa chính là thuộc tính bắc cầu

**Bước 2:** Các thuộc tính còn lại lập thành một quan hệ có khóa chính là quan hệ ban đầu.

*Ví dụ 4.20*: cho R=(ABCD)

F={AB 🡪 C,C 🡪 D)

R không phải chuẩn 3NF

Tách R(ABCD)

R1(CD);F={C 🡪 D} Khoá là C

R2(ABC); F={AB 🡪 C} Khoá AB

*Ví dụ 4.21*: Sơ đồ quan hệ NSX ở ví dụ trên không thỏa dạng chuẩn 3 vì có chứa phụ thuộc hàm truyền:

TP 🡪 Nuoc

Ta có thể tách NSX thành 2 sơ đồ quan hệ:

NSX(MaNSX, VonNSX, TP)

TP(TP, Nuoc)

thì 2 sơ đồ này thỏa 3NF.

**4.7.5 Dạng chuẩn thứ tư (BCNF)**

Một sơ đồ quan hệ R được gọi là thỏa dạng chuẩn Boyce – Codd nếu với mọi phụ thuộc hàm không tầm thường đều có vế trái là siêu khoá.

*Ví dụ 4.22:*

Sơ đồ quan hệ CSZ (khoá là CS và SZ) với các phụ thuộc hàm: CS 🡪 Z và Z 🡪 C thỏa dạng chuẩn 3 nhưng không thỏa Boyce – Codd vì phụ thuộc hàm Z 🡪 C, Z không là siêu khoá.

*Ví dụ 4.23:*

Xét lược đồ:

LOPHOC(Lop, MonHoc, GiaoVien)

với 2 phụ thuộc hàm:

GiaoVien 🡪 Monhoc

Lop, MonHoc 🡪 GiaoVien

Lược đồ có 2 khoá:

K1 = Lop, MonHoc và

K2 = Lop, GiaoVien

nên tất cả các thuộc tính đều là thuộc tính khoá 🡪 lược đồ ở dạng chuẩn 3. Tuy nhiên lược đồ không ở dạng chuẩn Boyce – Codd vì phụ thuộc hàm

GiaoVien 🡪 MonHoc

không thỏa yêu cầu vế trái phải là siêu khoá.

## 4.8 Kiểm tra phép tách quan hệ không làm mất thông tin

**4.8.1. Định nghĩa:**

Nếu R là một sơ đồ quan hệ tách thành các sơ đồ R1, R2,…, Rk và F là tập phụ thuộc hàm, ta nói phép tách có kết nối không mất thông tin nếu với mọi quan hệ r của R thoả F:

r = πR1(r) \* πR2(r) \*…\* πRk(r)

**4.8.2. Bổ đề:**

Cho R là một sơ đồ quan hệ, ρ = {R1, R2,…, Rk} là một phép tách R, r là một quan hệ trên R. Đặt:

mρ(r) = πR1(r) \* πR2(r) \*…\* πRk(r)

Ta có:

(i): r ⊆ mρ(r)

(ii): Nếu s = mρ(r) thì πRi(s) = ri­

(iii): mρ(s) = mρ(r)

**4.8.3. Kiểm tra kết nối không mất thông tin:**

***a. Giải thuật:***

Input: Một sơ đồ quan hệ R = (A­1,…,An), một tập phụ thuộc hàm F và một phép tách

ρ = (R1, R2,…, Rk)

Output: Một quyết định xem có là một phép tách với một kết nối không mất thông tin.

Method:

- Ta xây dựng một bảng k dòng và n cột, cột j tương ứng thuộc tính thứ j và dòng i tương ứng sơ đồ quan hệ Ri. Tại vị trí giao của dòng i và cột j ta đặt ký hiệu aj nếu Aj có trong Ri, ký hiệu bij nếu ngược lại.

- Lặp lại việc xét từng phụ thuộc hàm X 🡪 Y trong F, cho đến khi không còn thay đổi được bảng. Mỗi lần xét X 🡪 Y ta tìm các dòng mang trị bằng nhau trên tập thuộc tính X. Nếu tìm được hai dòng như vậy, làm cho các ký hiệu trên hai dòng đó giống nhau tại các thuộc tính của Y (nếu một ký hiệu là aj­ ta cho ký hiệu thứ hai cũng là aj; nếu chúng là bij và blj ta cho chúng cùng là bij hoặc cùng là blj.

- Nếu sau khi sửa bảng theo cách như trên, ta thấy có dòng nào đó có dạng a1, a2,…, ak thì kết nối là không mất thông tin, ngược lại là mất.

*Ví dụ 4.24*

Ta hãy xét phép tách SAIP thành SA và SIP. Các phụ thuộc hàm gồm S 🡪 A và SI 🡪 P, và bảng ban đầu là:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | S | A | I | P |
| SA | a1 | a2 | b13 | b14 |
| SIP | a1 | b22 | a3 | a4 |

Xét S 🡪 A, ta thấy hai dòng có giá trị giống nhau ở cột S, vì vậy giá trị ở cột A cần thay đổi thành a2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | S | A | I | P |
| SA | a1 | a2 | b13 | b14 |
| SIP | a1 | a2 | a3 | a4 |

Vì bảng có một dòng chứa toàn aj nên phép tách này là phép tách không mất thông tin.

***c. Định lý 1:***

Thuật toán trên xác định đúng nếu phép tách là không mất mát thông tin.

***d. Định lý 2:***

\* Phát biểu:

Nếu ρ = (R1, R2) là một phép tách sơ đồ quan hệ R và F là tập phụ thuộc hàm, thì có một phép tách không mất thông tin thoả F nếu và chỉ nếu:

(R1 ∩ R2) 🡪 (R1 – R2)

hoặc

(R1 ∩ R2) 🡪 (R2 – R1)

*Chú ý*: Các PTH này không cần thuộc F, chỉ cần thuộc F+.

\* Áp dụng:

Trong khi giải thuật trên có thể áp dụng cho một phép tách thành một số bất kỳ các sơ đồ quan hệ, định lý 2 cho một cách kiểm tra đơn giản hơn đối với phép tách R thành hai sơ đồ quan hệ.

*Ví dụ 4.25*

Giả thiết R = ABC và F = {A🡪 B}.

Như vậy, phép tách R thành AB và AC là phép tách không mất thông tin vì

AB ∩ AC = A

AB – AC = B

và ta có: A 🡪 B

**BÀI TẬP CHƯƠNG 4**

**4.1. Tìm phủ tối thiểu của lược đồ quan hệ sau**

1. Cho phụ thuộc hàm F={AB->C, C->A, BC->D, ACD->B, D->EG, BE->C, CG-> BD, CE->AG }. Tìm phủ tối thiểu của F
2. Cho quan hệ R(CZD) với các phụ thuộc hàm F={CD->Z, và CZ->C. }. Tìm phủ tối thiểu của F
3. Cho lược đồ quan hệ R(ABCDEGH) và phụ thuộc hàm F={B->AC, A->BDE, AB->GH, ABD->CE}.}. Tìm phủ tối thiểu của F
4. Cho lược đồ R(XYWUST) và phụ thuộc hàm F={XY->W, XW->U, XYW->ST, X->Y

**4.2 Tìm khóa của các lược đồ sau:**

1. Cho lược đồ quan hệ R và tập phụ thuộc hàm F={AB->E, AG->I, BE->I, E->G, GI->H}.
2. Cho lược đồ quan hệ R(CTHRSG) với Phụ thuộc hàm tối thiểu G={C->T, HR->C, CS->G và HS->R}.
3. Cho lược đồ quan hệ R(ABCDEGH) và phụ thuộc hàm F={B->AC, A->BDE, AB->GH, ABD->CE}.
4. Tìm khoá của R(N,M,P,Q,R,T,U,W) Với :

F=[M->W;MR->T,T->R,QR->T,M->U;MT->P,NP->Q]?

1. Tìm khoá của R(A,B,C,D,E,F,G,H,I,J) với F={AB->C,BD->EF,AD->GH,A->I,H->J}?
2. Cho R(A,B,C,D,E,F,G,H,I,J) VớI f={AB->C,A->DE,B->F,F->GH,D->IJ}

**4.3 Kiểm tra việc tách có mất mát thông tin hay không?**

1. Cho quan hệ R(ABCD) và phụ thuộc hàm F={A->B,AC->D}. Phân rã thành R1(AC), R2(BCD). Hỏi phép tách trên có bảo toàn thông tin hay không?

2.Cho quan hệ R(ABCD) và phụ thuộc hàm F={A->B,AC->D}. Phân rã thành R1(AB), R2(ACD). Hỏi phép tách trên có bảo toàn thông tin hay không

3: Cho l­ợc đồ R(BACSQD) và F={S->D,C->B,CS->Q,B-> A}

R1(SD),R2(CB),R3(CSQ),R4(BA).Phéptáchtrêncómấtmátthông tin không ?

4. Cho l­ợc đồ R(IFCDE) và tậpcácphụthuộchàm F={I->C,F->C,C->D,DE->C,CE->I}.Tách R(IFCDE) thành R1(CI),R2(CD),R3(FE),R4(FC),R5(IE) và,phép tách nàycómất mát thông tin không ?

**4.4. Cho lược đồ cơ sở dữ liệu với 3 lược đồ quan hệ sau:**

**Nhanvien(MANV,HOTEN,NGAYSINH,ĐIACHI, MAPB,MACV)**

Mỗi nhân viên có một mã nhân viên (MANV) duy nhất, mỗi MANV xác định các thông tin: họ và tên (HOTEN), ngày sinh (NGAYSINH) - NGÀY SINH là dạng ngày tháng năm, địa chỉ (ĐIACHI), mã phòng ban(MAPB) và mã chức vụ (MACV).

**Phongban (MAPB,TENPB)**

Mỗi phòng ban có một mã phòng ban (MAPB) duy nhất, mỗi MAPB xác định tên phòng ban (TENPB).

**Chucvu(MACV,TENCV,PHUCAP)**

Mỗi chức vụ có một mã chức vụ (MACV) duy nhất, mỗi MACV xác định tên chức vụ (TENCV), phu cấp chức vụ(PHUCAP)

a.Tìm khóa cho mỗi lược đồ quan hệ trên.

b. Tìm tất cả các phụ thuộc hàm của lược đồ quan hệ trên

c. Xác định lược đồ quan hệ trên thuộc chuẩn mấy

**4.5. Mô tả cơ sở dữ liệu quản lý của phòng giáo dục như sau:**

Một phòng giáo dục quận muốn lập một hệ thống thông tin để quản lý việc làm thi tốt nghiệp phổ thông cơ sở (lớp 9). Công việc làm thi được tổ chức như sau:

Lãnh đạo phòng giáo dục thành lập nhiều hội đồng thi (mỗi hội đồng thi gồm một trường hoặc một số trường gần nhau). Mỗi hội đồng thi có một mã số duy nhất (MAHĐT), một hội đồng thi xác định tên hội đồng thi(TENHĐT), họ tên chủ tịch hội đồng(TENCT), địa chỉ (ĐCHĐT),điện thoại(ĐTHĐT).

Mỗi hội đồng thi được bố trí cho một số phòng thi, mỗi phòng thi có một số hiệu phòng(SOPT) duy nhất, một phòng thi xác định địa chỉ phòng thi (ĐCPT). Số hiệu phòng thi được đánh số khác nhau ở tất cả các hội đồng thi.

Giáo viên của các trường trực thuộc phòng được điều động đến các hội đồng để coi thi, mỗi trường có thể có hoặc không có thí sinh dự thi, mỗi trường có một mã trường duy nhất (MATR), mỗi mã trường xác định tên trường(TENTR),địa chỉ (ĐCTR), loại hình đào tạo (LHĐT) (Công lập, chuyên, bán công, dân lập,…). Các giáo viên của một trường có thể làm việc tại nhiều hội đồng thi. Một giáo viên có một mã giáo viên(MAGV), một mã giáo viên xác định tên giáo viên (TENGV), chuyên môn giảng dạy (CHUYENMON), chức danh trong hội đồng thi(CHUCDANH).

Các thí sinh dự thi có một số báo danh duy nhất(SOBD), mỗi số báo danh xác định tên thí sinh(TENTS), ngày sinh (NGSINH), giới tính (PHAI), mỗi thí sinh được xếp thi tại một phòng thi nhất định cho tất cả các môn, mỗi thí sinh có thể có chứng chỉ nghề (CCNGHE) hoặc không (thuộc tính CCNGHE kiểu chuổi, CCNGHE=”x” nếu thí sinh có chứng chỉ nghề và CCNGHE bằng rổng nếu thí sinh không có chứng chỉ nghề).Thí sinh của cùng một trường chỉ dự thi tại một hội đồng thi.

Mỗi môn thi có một mã môn thi duy nhất (MAMT), mỗi mã môn thi xác định tên môn thi(TENMT), buổi thi (BUOI), ngày thi (NGAY). Giả sử toàn bộ các thí sinh trong hội đồng thi đó đều thi chung một số môn do sở giáo dục quy định (có thể thay đổi tuỳ theo năm). Mỗi môn thi được tổ chức trong một buổi của một ngày nào đó. Ứng với mỗi môn thi một thí sinh có một điểm thi duy nhất(ĐIEMTHI)

Dựa vào phân tích ở trên, giả sử ta có lược đồ cơ sở dữ liệu sau: HĐ(MAHĐT,TENHĐT, TENCT, ĐCHĐT,ĐTHĐT)

PT(SOPT,ĐCPT,MAHĐT)

TS(SOBD, TENTS,NGSINH,PHAI,CCNGHE, MATR,SOPT)

MT(MAMT,TENMT,BUOI,NGAY) GV(MAGV,TENGV,CHUYENMON,CHUCDANH,MAHĐT,MATR) TR(MATR,TENTR,ĐCTR,LHĐT)

KQ(SOBD,MAMT,ĐIEMTHI)

**YÊU CẦU**

1. Hãy xác định khóa cho mỗi lược đồ quan hệ trên.
2. Liệt kê các phụ thuộc hàm xuất hiện trên lược đồ quan hệ
3. Xác định lược đồ quan hệ trên ở dạng chuẩn mấy? Giải thích

**TÀI LIỆU THAM KHẢO**

🙥🙧

* Lê Tiến Vương, nhập môn Cơ sở dữ liệu quan hệ, NXB Khoa học và Kỹ thuật.
* Nguyễn Đăng Tỵ - Đỗ Phúc, giáo trình Cơ sở dữ liệu quan hệ, NXB Đại học Quốc Gia TP Hồ Chí Minh.
* Trịnh Minh Tuấn, thiết kế cơ sở dữ liệu, NXB Đại học Quốc Gia TP Hồ Chí Minh.
* Đỗ Trung Tuấn. *Cơ sở dữ liệu*. Nhà Xuất bản Giáo dục, 1998.