

## GIỚI THIỆU

Cơ sở dữ liệu được sử dụng rộng khắp trong ngành công nghệ thông tin nói riêng và lĩnh vực kinh doanh nói chung. Mỗi ngày, chúng ta sử dụng cơ sở dữ liệu cả trực tiếp và gián tiếp từ giao dịch ngân hàng, đặt vé du lịch, website tìm kiếm việc làm, mua sắm trực tuyến cho đến hầu hết mọi giao dịch khác, đều được lưu trữ và phục vụ bởi cơ sở dữ liệu.

Đi kèm sự phát triển như vũ bão của công nghệ là sự ra đời của các tiêu chuẩn trong công nghệ cơ sở dữ liệu. Kết quả là sự ra đời của vô số sản phẩm thương mại, Mỗi sản phẩm lại mang một tầm nhìn riêng của nhà cung cấp phần mềm. Ngoài ra, một số mô hình cơ sở dữ liệu khác nhau cũng ra đời, trong đó nổi bật nhất là mô hình cơ sở dữ liệu quan hệ.

Thách thức lớn nhất khi cài đặt cơ sở dữ liệu là thiết kế chính xác cấu trúc của cơ sở dữ liệu đó. Nếu không hiểu rõ vấn đề cần giải quyết, thiếu kinh nghiệm tổ chức các dữ liệu cần thiết một cách tối ưu, cơ sở dữ liệu tạo ra sẽ trở nên cồng kềnh, khó quản lý, khó kiểm soát.

Tập bài giảng này phục vụ giảng viên trong công tác giảng dạy, đồng thời là tài liệu nghiên cứu của sinh viên, trình bày những vấn đề cốt lõi nhất của môn cơ sở dữ liệu. Các bài học được trình bày ngắn gọn, nhiều ví dụ. Cuối mỗi chương đều có bài tập để sinh viên luyện tập.

Việc biên soạn chắc chắn còn nhiều khiếm khuyết. Rất mong được sự đóng góp ý kiến của các đồng nghiệp, các em sinh viên và độc giả để chất lượng ngày càng hoàn thiện.

Chân thành cảm ơn.

Đà Nẵng, tháng 8 năm 2017

**Người biên soạn**

# MỤC LỤC

<b>GIỚI THIỆU .....</b>	<b>I</b>
<b>MỤC LỤC .....</b>	<b>II</b>
<b>CHƯƠNG 1. TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU .....</b>	<b>7</b>
<b>I. CÁC KHÁI NIỆM CƠ BẢN.....</b>	<b>7</b>
1. Cơ sở dữ liệu (Database) .....	7
1.1. Một số ví dụ về cơ sở dữ liệu .....	7
1.2. Định nghĩa .....	8
1.3. Ưu điểm của cơ sở dữ liệu.....	8
1.4. Những vấn đề mà CSDL cần phải giải quyết.....	8
1.5. Các đối tượng sử dụng CSDL.....	9
2. Hệ quản trị cơ sở dữ liệu (Database Management System).....	9
3. Mô hình dữ liệu (Data Model).....	10
<b>II. MÔ HÌNH THỰC THỂ KẾT HỢP.....</b>	<b>11</b>
1. Thực Thể (Entity) .....	11
2. Thuộc tính (Attribute).....	11
3. Loại thực thể (Entity type).....	11
4. Khoá (Key).....	11
5. Quan hệ (Relationship).....	13
5.1. Bản số của quan hệ.....	13
5.2. Thuộc tính của quan hệ.....	13
5.3. Khoá của quan hệ.....	14
6. Các bước thành lập mô hình thực thể kết hợp .....	14
<b>III. BÀI TẬP .....</b>	<b>15</b>
<b>CHƯƠNG 2. MÔ HÌNH DỮ LIỆU QUAN HỆ.....</b>	<b>20</b>
<b>I. CÁC KHÁI NIỆM CƠ BẢN.....</b>	<b>20</b>
1. Thuộc tính (Attribute).....	20
1.1. Kiểu dữ liệu (Data type) .....	20

1.2. Miền giá trị ( <i>Domain of values</i> ) .....	20
2. Lược đồ quan hệ ( <i>Relation schema</i> ) .....	21
3. Quan hệ ( <i>Relation</i> ) .....	22
4. Bộ ( <i>Tuple</i> ) .....	22
5. Siêu khoá – Khoá ( <i>Super key- Key</i> ) .....	22
II. CHUYỂN MÔ HÌNH THỰC THỂ KẾT HỢP SANG MÔ HÌNH DỮ LIỆU QUAN HỆ .....	23
1. Quy tắc 1 .....	23
2. Quy tắc 2 .....	24
3. Quy tắc 3 .....	24
4. Quy tắc 4 .....	24
III. CÁC PHÉP TOÁN ĐẠI SỐ TRÊN QUAN HỆ .....	25
1. Đại số quan hệ .....	25
2. Các phép toán tập hợp .....	26
2.1. Phép Hợp 2 quan hệ ( <i>Union</i> ) .....	26
2.2. Phép Giao 2 quan hệ ( <i>Intersection</i> ) .....	26
2.3. Phép Trừ 2 quan hệ ( <i>Minus</i> ) .....	27
2.4. Tích Decac của 2 quan hệ ( <i>Cartesian Product</i> ) .....	27
2.5. Phép chia 2 quan hệ .....	28
3. Các phép toán quan hệ .....	29
3.1. Phép Chiếu ( <i>Projection</i> ) .....	29
3.2. Phép Chọn ( <i>Selection</i> ) .....	29
3.3. Phép $\theta$ - kết .....	29
3.4. Phép Kết Tự Nhiên ( <i>Natural join</i> ) .....	30
4. Ví dụ về ngôn ngữ đại số quan hệ .....	30
IV. BÀI TẬP .....	31
<b>CHƯƠNG 3. NGÔN NGỮ SQL .....</b>	<b>33</b>
I. TỔNG QUAN .....	33

1. Lịch sử phát triển .....	33
2. SQL là ngôn ngữ chuẩn cho các hệ quản trị cơ sở dữ liệu quan hệ.....	33
3. Các thành phần cơ bản của SQL .....	35
3.1. Kiểu dữ liệu.....	35
3.2. Các toán tử.....	35
4. Các quy ước biểu diễn câu lệnh SQL .....	36
II. CÁC LỆNH ĐỊNH NGHĨA DỮ LIỆU (DDL) .....	36
1. Tạo một cơ sở dữ liệu .....	36
2. Tạo một bảng - CREATE .....	37
2.1. Cú pháp.....	37
2.2. Các loại ràng buộc trong bảng.....	37
3. Sửa đổi cấu trúc – ALTER .....	40
4. Xoá đối tượng – DROP .....	40
III. CÁC LỆNH XỬ LÝ CẬP NHẬT DỮ LIỆU (DML).....	41
1. Thêm hàng – INSERT .....	41
2. Xóa hàng – DELETE.....	41
3. Sửa đổi giá trị của một hàng – UPDATE .....	42
IV. CÁC LỆNH KIỂM SOÁT DỮ LIỆU (DCL).....	42
1. Lệnh cấp quyền – GRANT.....	42
2. Lệnh thu hồi quyền – REVOKE.....	44
V. CÁC LỆNH TRUY VẤN DỮ LIỆU (DQL) .....	44
1. Cú pháp tổng quát.....	44
2. Các thành phần cơ bản của câu lệnh.....	45
3. Các loại biểu thức .....	46
3.1. Truy vấn đơn giản – Mệnh đề SELECT .....	46
3.2. Truy vấn có điều kiện – Mệnh đề WHERE.....	47
3.3. Truy vấn có sắp xếp – Mệnh đề ORDER BY.....	49
3.4. Câu lệnh truy vấn lồng nhau .....	50

3.5. Các hàm tính toán.....	55
3.6. Gom nhóm dữ liệu – mệnh đề GROUP BY.....	55
VI. BÀI TẬP .....	57
<b>CHƯƠNG 4. RÀNG BUỘC TOÀN VỆN.....</b>	<b>62</b>
I. RÀNG BUỘC TOÀN VỆN .....	62
1. Khái niệm ràng buộc toàn vẹn.....	62
2. Các yếu tố của ràng buộc toàn vẹn.....	62
2.1. Điều kiện .....	62
2.2. Bối cảnh .....	63
2.3. Bảng tâm ảnh hưởng .....	63
2.4. Hành động cần phải có khi phát hiện có RBTV bị vi phạm:.....	64
II. PHÂN LOẠI CÁC RÀNG BUỘC TOÀN VỆN.....	64
1. Ràng buộc toàn vẹn có bối cảnh là một quan hệ.....	66
1.1. Ràng buộc toàn vẹn liên bộ .....	66
1.2. Ràng buộc toàn vẹn về miền giá trị.....	67
1.3. Ràng buộc toàn vẹn liên thuộc tính.....	67
2. Ràng buộc toàn vẹn có bối cảnh là nhiều quan hệ .....	67
2.1. Ràng buộc toàn vẹn về khóa ngoại.....	67
2.2. Ràng buộc toàn vẹn liên thuộc tính liên quan hệ.....	68
2.3. Ràng buộc toàn vẹn liên bộ liên quan hệ .....	68
III. BÀI TẬP .....	68
<b>CHƯƠNG 5. LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU .....</b>	<b>72</b>
I. CÁC VẤN ĐỀ GẶP PHẢI KHI TỔ CHỨC DỮ LIỆU .....	72
II. PHỤ THUỘC HÀM .....	72
1. Định nghĩa phụ thuộc hàm.....	72
2. Cách xác định phụ thuộc hàm cho lược đồ quan hệ.....	73
3. Một số tính chất của phụ thuộc hàm – hệ luật dẫn Armstrong .....	73

III. BAO ĐÓNG CỦA TẬP PHỤ THUỘC HÀM VÀ BAO ĐÓNG CỦA TẬP THUỘC TÍNH.....	74
1. Bao đóng của tập phụ thuộc hàm F .....	74
2. Bao đóng của tập thuộc tính X .....	75
3. Bài toán thành viên .....	76
4. Thuật toán tìm bao đóng của một tập thuộc tính.....	76
IV. KHÓA CỦA LƯỢC ĐỒ QUAN HỆ - MỘT SỐ THUẬT TOÁN TÌM KHÓA	77
1. Định nghĩa khóa của quan hệ .....	77
2. Thuật toán tìm một khóa của một lược đồ quan hệ.....	77
3. Thuật toán tìm tất cả các khóa của một lược đồ quan hệ .....	78
V. PHỦ TỐI THIỂU .....	80
1. Tập phụ thuộc hàm tương đương .....	80
2. Phủ tối thiểu .....	80
2.1. <i>Phụ thuộc hàm có vế trái dư thừa</i> .....	80
2.2. <i>Tập phụ thuộc hàm có vế phải một thuộc tính</i> .....	81
2.3. <i>Tập phụ thuộc hàm không dư thừa</i> .....	81
3. Thuật toán tìm phủ tối thiểu.....	82
VI. DẠNG CHUẨN CỦA LƯỢC ĐỒ QUAN HỆ .....	82
1. Một số khái niệm liên quan đến các dạng chuẩn.....	82
2. Dạng chuẩn 1 (First Normal Form) .....	82
3. Dạng chuẩn 2 (Second Normal Form) .....	83
4. Dạng chuẩn 3 (Third Normal Form) .....	84
5. Dạng chuẩn BC (Boyce Codd Normal Form).....	86
VII. BÀI TẬP .....	88
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>91</b>

## CHƯƠNG 1. TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

### I. CÁC KHÁI NIỆM CƠ BẢN

#### 1. Cơ sở dữ liệu (Database)

##### 1.1. Một số ví dụ về cơ sở dữ liệu

**Ví dụ 1.1:** Trong một công ty, chúng ta quản lý nhân sự theo bảng sau:

Mã nhân viên	Họ tên	Giới tính	Năm sinh	Trình độ đào tạo	Lương
001	Nguyễn Văn A	Nam	1970	Đại học	300000
002	Nguyễn Kim Anh	Nữ	1971	Trung cấp	210000
003	Trần Văn ánh	Nam	1969	Đại học	500000
004	Trần Bình	Nam	1965	PTS	450000
.....					
120	Trần Thị yến	Nữ	1967	PTS	455000

Mỗi người là một mối quan hệ giữa các **thuộc tính**: Mã nhân viên, Họ tên, Giới tính, Năm sinh, Trình độ đào tạo, Lương.

Bảng trên được gọi là **bảng dữ liệu**, mỗi người sử dụng lại có cách khai thác dữ liệu trong bảng trên theo mục đích riêng, chẳng hạn:

- Một nhân viên cần tra cứu mức lương của mình → anh ta quan tâm tới dòng chứa thông tin của riêng mình, và tất nhiên buộc phải có thông tin tại cột lương.
- Giám đốc muốn biết công ty có bao nhiêu người → đếm số dòng trên bảng.
- Phòng tài chính dự chi tiền lương hàng tháng → tính tổng cột lương

Điều này có nghĩa là các dữ liệu của bảng trên độc lập với các xử lý tác động lên chúng.

Dữ liệu trên được tổ chức thành các cột và các hàng. Các cột hình thành nên bộ khung của bảng dữ liệu, gọi là lược đồ (**scheme**). Mỗi cột gọi là một thuộc tính (**attribute**) hay là trường (**field**). Mỗi dòng gọi là một thể hiện (**instance**) hay là bản ghi (**record**) của bảng dữ liệu.

Các thao tác thường áp dụng lên bảng dữ liệu là: tạo/thêm/ xóa/ sửa một bản ghi - CRUD (Create, Read, Update, Delete).

### **1.2. Định nghĩa**

Trong công ty, có nhiều bảng dữ liệu tương tự như bảng thông tin nhân sự bên trên. Thông thường khi thiết kế và xây dựng các hệ thống tin quản lý, chúng ta lưu trữ các bảng dữ liệu trong các file để xử lý.

Một cơ sở dữ liệu là một hệ thống các file dữ liệu, mỗi file này có cấu trúc bản ghi khác nhau, nhưng về mặt nội dung có quan hệ với nhau.

Cơ sở dữ liệu (CSDL) là một hệ thống các thông tin có cấu trúc được lưu trữ trên các thiết bị như băng từ, đĩa từ,... để có thể thỏa mãn yêu cầu khai thác đồng thời của nhiều người sử dụng.

CSDL gắn liền với đại số, logic toán và một số lĩnh vực khác.

### **1.3. Ưu điểm của cơ sở dữ liệu**

- Giảm sự trùng lặp 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.

### **1.4. Những vấn đề mà CSDL cần phải giải quyết**

- Tính chủ quyền của dữ liệu: tính chủ quyền của dữ liệu được thể hiện ở phương diện an toàn dữ liệu, khả năng biểu diễn các mối liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu. Điều này có nghĩa là người khai thác CSDL phải có nhiệm vụ cập nhật các thông tin mới nhất của CSDL.
- Tính bảo mật và quyền khai thác thông tin của người sử dụng: do có nhiều người được phép khai thác dữ liệu một cách đồng thời, nên cần thiết phải có một cơ chế bảo mật và phân quyền hạn khai thác CSDL. Các hệ điều hành đa người dùng hay hệ điều hành mạng cục bộ đều có cung cấp cơ chế này.
- Tranh chấp dữ liệu: nhiều người được phép truy nhập cùng một lúc vào tài nguyên dữ liệu của CSDL với những mục đích khác nhau, do đó cần thiết phải có một cơ chế



lập lịch khi truy nhập dữ liệu. Cơ chế lập lịch có thể được thực hiện bằng việc cấp quyền trước/sau cho từng người khai thác.

– Đảm bảo an toàn dữ liệu khi có sự cố: việc quản lý dữ liệu tập trung có thể làm tăng khả năng mất mát hoặc sai lệch thông tin khi có sự cố như mất điện đột xuất, hay một phần đĩa lưu trữ CSDL bị hư,... một số hệ điều hành mạng có cung cấp dịch vụ sao lưu ảnh đĩa cứng, tự động kiểm tra và khắc phục lỗi khi có sự cố. Tuy nhiên, bên cạnh dịch vụ của hệ điều hành, để đảm bảo CSDL luôn ổn định, một CSDL nhất thiết phải có một cơ chế khôi phục dữ liệu khi có các sự cố bất ngờ xảy ra.

### ***1.5. Các đối tượng sử dụng CSDL***

- Những người sử dụng CSDL không chuyên về lĩnh vực tin học và CSDL.
- Các chuyên viên CSDL biết khai thác CSDL Những người này có thể xây dựng các ứng dụng khác nhau, phục vụ cho các mục đích khác nhau trên CSDL.
- Những **người quản trị CSDL**, đó là những người hiểu biết về tin học, về các hệ quản trị CSDL và hệ thống máy tính. Họ là người tổ chức CSDL, do đó họ phải nắm rõ các vấn đề kỹ thuật về CSDL để có thể phục hồi CSDL khi có sự cố. Họ là những người cấp quyền hạn khai thác CSDL, do vậy họ có thể giải quyết được các vấn đề tranh chấp dữ liệu nếu có.

## **2. Hệ quản trị cơ sở dữ liệu (Database Management System)**

Một hệ quản trị cơ sở dữ liệu là một *phần mềm* quản lý và điều hành các file dữ liệu. Nói chung một hệ quản trị cơ sở dữ liệu thường có những đặc tính sau :

- Có tính độc lập với các công cụ lưu trữ.
- Có tính độc lập với các chương trình phần mềm của người sử dụng (có nghĩa là các ngôn ngữ lập trình khác nhau có thể được dùng trong hệ này).
- Có khả năng tại một thời điểm truy nhập vào nhiều nơi trong hệ này.
- Có khả năng khai thác tốt tiềm năng của máy.
- Người dùng với kiến thức tối thiểu cũng có thể sử dụng được hệ này,
- Bảo đảm an toàn dữ liệu và bảo mật dữ liệu.
- Thuận lợi và mềm dẻo trong việc bổ sung, loại bỏ, thay đổi dữ liệu.

- Giảm bớt sự dư thừa dữ liệu trong lưu trữ.

Hiện nay trên thị trường phần mềm đã có những hệ quản trị CSDL hỗ trợ được nhiều tiện ích như: MS Access, Visual Foxpro, SQL Server Oracle, ...

### 3. Mô hình dữ liệu (Data Model)

Trong quá trình thiết kế và xây dựng các hệ quản trị cơ sở dữ liệu, người ta tiến hành xây dựng các mô hình dữ liệu. Mô hình dữ liệu phải *thể hiện được các mối quan hệ bản chất của các dữ liệu* mà các dữ liệu này phản ánh các mối quan hệ và các thực thể trong thế giới hiện thực. Có thể thấy mô hình dữ liệu phản ánh khía cạnh cấu trúc logic mà không đi vào khía cạnh vật lý của các cơ sở dữ liệu. Khi xây dựng các mô hình dữ liệu cần phân biệt các thành phần cơ bản sau :

- **Thực thể** (Entity): Đó là đối tượng có trong thực tế mà chúng ta cần mô tả các đặc trưng của nó.
- **Thuộc tính** (Attribute): Đó là các dữ liệu thể hiện các đặc trưng của thực thể.
- **Quan hệ** (Relationship): Đó là các mối quan hệ logic của các thực thể.

Sau đây chúng ta sẽ điểm qua lịch sử phát triển của các mô hình dữ liệu:

- Vào những năm sáu mươi, thế hệ đầu tiên của CSDL ra đời dưới dạng mô hình thực thể kết hợp, mô hình mạng và mô hình phân cấp.
- Vào những năm bảy mươi, thế hệ thứ hai của CSDL ra đời. Đó là mô hình dữ liệu quan hệ do EF. Codd phát minh. Mô hình này có cấu trúc logic chặt chẽ. Đây là mô hình đã và đang được sử dụng rộng khắp trong công tác quản lý trên phạm vi toàn cầu. Việc nghiên cứu mô hình dữ liệu quan hệ nhằm vào lý thuyết chuẩn hoá các quan hệ là một công cụ quan trọng trong việc phân tích thiết kế các hệ CSDL hiện nay. Mục đích của nghiên cứu này nhằm bỏ đi các phần tử không bình thường của quan hệ khi thực hiện các phép cập nhật, loại bỏ các phần tử dư thừa.
- Sang thập kỷ tám mươi, mô hình CSDL thứ ba ra đời, đó là mô hình cơ sở dữ liệu hướng đối tượng, mô hình cơ sở dữ liệu phân tán, mô hình cơ sở dữ liệu suy diễn,.

Hiện nay, các hệ quản trị CSDL thường dùng mô hình dữ liệu quan hệ, 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, diễn hình nhất là mô hình thực thể kết hợp. Sau đó

dùng một số quy tắc để chuyển hệ thống từ mô hình thực thể kết hợp về mô hình dữ liệu quan hệ - các quy tắc này sẽ được nói đến trong mục **II-Chương 2**.

## **II. MÔ HÌNH THỰC THỂ KẾT HỢP**

Đây là mô hình dữ liệu tiêu biểu nhất để thiết kế (bước đầu) một ứng dụng tin học. Sau đây là các khái niệm của mô hình thực thể kết hợp.

### **1. 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.

### **2. 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)*

### **3. 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 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,.

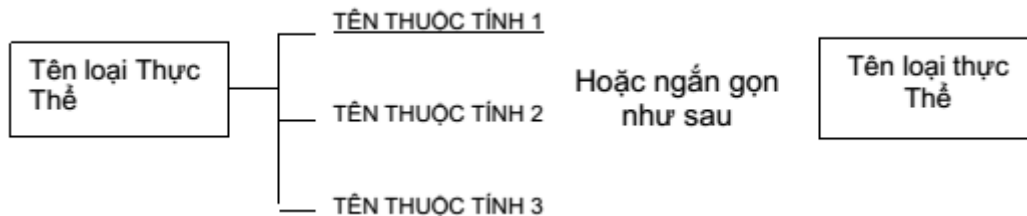
*(Trong giáo trình này, tên của loại thực thể được in hoa ký tự đầu tiên, các ký tự còn lại viết thường).*

### **4. 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 khoá của loại thực thể Sinhvien là MASV, của Lớp là MALOP, của Khoa là MAKHOA, của Monhoc là MAMH,... Các mã này sẽ là đơn nhất (không trùng nhau).

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 1.2:** Bài toán quản lý điểm của sinh viên.

Mỗi sinh viên cần quản lý các thông tin như: họ và tên (HOTENSV), ngày tháng năm sinh(NGAYSINH), giới tính (NU), nơi sinh (NOISINH), 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

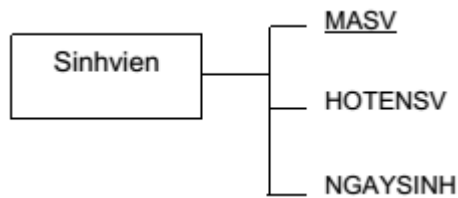
Mỗi khoa có một tên gọi (TENKHOA) và một mã số duy nhất (MAKHOA) để phân biệt với các khoa khác.

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 thuộc về sự quản lý hành chính của một khoa, nhưng có thể dạy nhiều môn ở nhiều khoa khác.

**Xác định các thực thể:** Sinhviên, Môn học, Khoa, Lớp, Giảngviên

**Biểu diễn thực thể:** Ví dụ với thực thể Sinhviên biểu diễn như sau:



## 5. Quan hệ (Relationship)

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

Ví dụ quan hệ giữa hai loại thực thể Sinhviên và Lớp, quan hệ giữa Sinhviên với Môn học,...

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

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

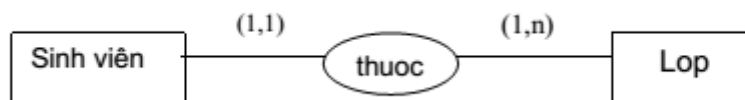


### 5.1. Bản số của quan hệ

Bản số của một nhánh R trong quan hệ 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 quan hệ đó.

Ví dụ

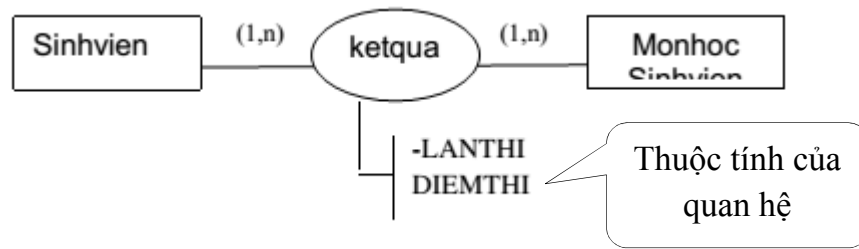


Có nghĩa là: “mỗi sinh viên thuộc một và chỉ một lớp nên bản số bên nhánh Sinhviên là (1,1), mỗi lớp có 1 đến n sinh viên nên bản số bên nhánh Lớp là (1,n)”

### 5.2. Thuộc tính của quan hệ

Trong một số trường hợp đặc biệt, quan hệ 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 Môn học và Sinhviên có quan hệ ketqua với ý nghĩa: “mỗi sinh viên ứng với mỗi lần thi của mỗi môn học có một kết quả điểm thi duy nhất”.



### 5.3. Khoá của quan hệ

Là hợp của các khoá của các loại thực thể liên quan.

Chẳng hạn như thuộc tính MAGV là khoá của loại thực thể Giảngvien, MALOP là thuộc tính khoá của loại thực thể Lớp, MAMH là thuộc tính khoá của loại thực thể Môn học, do đó quan hệ “phancong” (giữa các loại thực thể Giảngvien, Lớp, Môn học) có khoá là {MAGV, MAMH, MALOP} - phancong là quan hệ 3 ngôi.

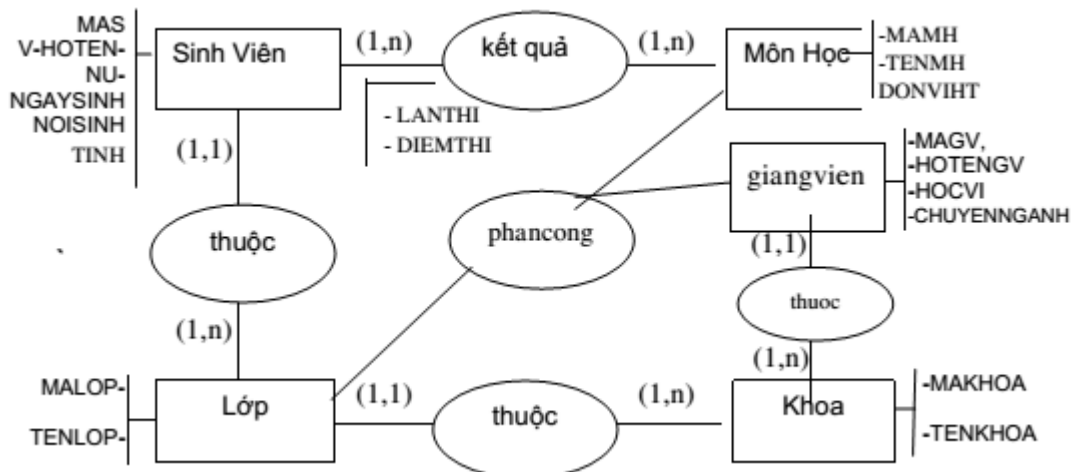
(Trong giáo trình này, tên của quan hệ được viết bằng chữ thường).

## 6. Các bước thành lập mô hình thực thể kết hợp

Việc thành lập mô hình thực thể kết hợp cho một ứng dụng tin học có thể tiến hành theo các bước sau:

- B1. Xác định danh sách các loại thực thể.
- B2. Xác định các quan hệ giữa các loại thực thể để phác thảo mô hình.
- B3. Lập bản số của các quan hệ.

**Ví dụ 1.3:** lập mô hình thực thể kết hợp cho bài toán quản lý điểm của sinh viên đã được nêu trong ví dụ 1.2.



### **III. BÀI TẬP**

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

#### **Bài tập 1.1. 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(ĐIADIEM), 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 tập 1.2. 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 (MADG), cùng với các thông tin khác như: họ tên (HOTEN), ngày sinh (NGAYSINH), địa chỉ (DIACHI), nghề nghiệp (NGHENGHIEP).

Cứ mỗi lượt mượn sách, độc giả phải đăng ký các quyền 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 quyền sách trong cùng một phiếu mượn không nhất thiết phải trả trong một lần. Mỗi quyền sách có thể thuộc nhiều phiếu mượn khác nhau (tất nhiên là tại các thời điểm khác nhau).

### **Bài tập 1.3. Quản lý lịch dạy của giáo viên**

Để quản lý lịch dạy của các giáo viên và lịch học của các lớp, một trường tổ chức như sau:

Mỗi giáo viên có một mã số giáo viên (MAGV) duy nhất, mỗi MAGV xác định các thông tin như: họ và tên giáo viên (HOTEN), số điện thoại (DTGV). Mỗi giáo viên có thể dạy nhiều môn cho nhiều khoa nhưng chỉ thuộc sự quản lý hành chánh của một khoa nào đó.

Mỗi môn học có một mã số môn học (MAMH) duy nhất, mỗi môn học xác định tên môn học (TENMH). Ứng với mỗi lớp thì mỗi môn học chỉ được phân cho một giáo viên.

Mỗi phòng học có một số phòng học (PHONG) duy nhất, mỗi phòng có một chức năng (CHUCNANG); chẳng hạn như phòng lý thuyết, phòng thực hành máy tính, phòng nghe nhìn, xưởng thực tập cơ khí,...

Mỗi khoa có một mã khoa (MAKHOA) duy nhất, mỗi khoa xác định các thông tin như: tên khoa (TENKHOA), điện thoại khoa (DTKHOA).

Mỗi lớp có một mã lớp (MALOP) duy nhất, mỗi lớp có một tên lớp (TENLOP), sĩ số lớp (SISO). Mỗi lớp có thể học nhiều môn của nhiều khoa nhưng chỉ thuộc sự quản lý hành chính của một khoa nào đó.

Hàng tuần, mỗi giáo viên phải lập lịch báo giảng cho biết giáo viên đó sẽ dạy những lớp nào, ngày nào (NGAYDAY), môn gì?, tại phòng nào, từ tiết nào (TUTIET) đến tiết nào (ĐENTIET), tựa đề bài dạy (BAIDAY), những ghi chú (GHICHU) về các tiết dạy này, đây là giờ dạy lý thuyết (LYTHUYET) hay thực hành - giả sử nếu LYTHUYET=1 thì đó



là giờ dạy thực hành và nếu LYTHUYET=2 thì đó là giờ lý thuyết, một ngày có 16 tiết, sáng từ tiết 1 đến tiết 6, chiều từ tiết 7 đến tiết 12, tối từ tiết 13 đến 16.

Một số yêu cầu của hệ thống này như:: Lập lịch dạy trong tuần của các giáo viên. Tổng số tiết dạy của các giáo viên theo từng môn cho từng lớp,....

#### **Bài tập 1.4. Quản lý học viên ở một trung tâm tin học**

Trung tâm tin học KTDN thường xuyên mở các lớp tin học ngắn hạn và dài hạn. Mỗi lớp ngắn hạn có một hoặc nhiều môn học (chẳng hạn như lớp Tin học văn phòng thì có các môn : Word, Power Point, Excel, còn lớp lập trình Pascal thì chỉ học một môn Pascal). Các lớp dài hạn (chẳng hạn như lớp kỹ thuật viên đồ hoạ đa truyền thông, lớp kỹ thuật viên lập trình, lớp kỹ thuật viên phần cứng và mạng,... ) thì có thể học nhiều học phần và mỗi học phần có thể có nhiều môn học.

Mỗi học viên có một mã học viên (MAHV) duy nhất và chỉ thuộc về một lớp duy nhất (nếu học viên cùng lúc học nhiều lớp thì ứng với mỗi lớp, học viên đó có một MAHV khác nhau). Mỗi học viên xác định họ tên (HOTEN), ngày sinh (NGAYSINH), nơi sinh (NOISINH), phái nam hay nữ(PHAI), nghề nghiệp (NGHENGHIEP) - nghề nghiệp là SINH VIÊN, GIÁO VIÊN, KỸSƯ, HỌC SINH, BUÔN BÁN,...

Trung tâm KTDN có nhiều lớp, mỗi lớp có một mã lớp duy nhất (MALOP), mỗi lớp xác định các thông tin: tên lớp (TENLOP), thời khoá biểu, ngày khai giảng (NGAYKG), học phí (HOCPhi).

Chú ý rằng tại một thời điểm, trung tâm có thể mở nhiều lớp cho cùng một chương trình học. Với các lớp dài hạn thì ngày khai giảng được xem là ngày bắt đầu của mỗi học phần và HOC PHÍ là học phí của mỗi học phần, với lớp ngắn hạn thì HOC PHÍ là học phí của toàn khoá học đó.

Trung tâm có nhiều môn học, mỗi môn học có mã môn học (MAMH) duy nhất, mỗi môn học xác định tên môn học (TENMH), số tiết lý thuyết (SOTIETLT), số tiết thực hành (SOTIETTH).

Mỗi học viên ứng với mỗi môn học có một điểm thi (DIEMTHI) duy nhất. Mỗi lần đóng học phí, học viên sẽ được trung tâm giao cho một phiếu biên lai thu tiền, mỗi biên lai có một số biên lai duy nhất để quản lý.

Một số yêu cầu của hệ thống này như::Lập danh sách những học viên khai giảng khoá ngày nào đó. Lập danh sách các học viên của một lớp ? Cho biết số lượng học viên của mỗi lớp khai giảng khoá ngày nào đó ?

### **Bài tập 1.5. Quản lý coi thi tuyển sinh**

Một hội đồng coi thi tuyển sinh có nhiều điểm thi, mỗi điểm thi được đặt tại một trường nào đó. Các điểm thi (DIEMTHISO) được đánh số là điểm thi số 1, điểm thi số 2, điểm thi số 3,... Mỗi điểm thi xác định địa chỉ (DIACHIDIEMTHI). Ví dụ: điểm thi số 1, đặt tại trường PTTH Nguyễn Thị Minh Khai, điểm thi số 2 đặt tại trường PTTH Bùi Thị Xuân,...

Mỗi thí sinh có một số báo danh (SOBD) duy nhất, mỗi số báo danh xác định các thông tin: họ và tên (HOTEN), ngày sinh (NGAYSINH), phái (PHAI), hộ khẩu thường trú (TINH), đối tượng dự thi (DOITUONG), ngành đăng ký thi, khu vực của thí sinh (KHUVUC), số hiệu phòng thi. Ví dụ: thí sinh Vũ Mạnh Cường, có số báo danh là 02978, sinh ngày 12/12/1984, phái nam, hộ khẩu thường trú tại Chợ Gạo - Tiền Giang, thuộc khu vực 1, đối tượng là 5B, đăng ký dự thi vào ngành có mã ngành là 01, thi tại phòng thi 0178, điểm thi số 1.

Mỗi ngành có một mã ngành (MANGANH) duy nhất, mỗi mã ngành xác định tên ngành (TENNGANH)

Mỗi điểm thi có nhiều phòng thi – mỗi phòng thi (PHONGTHI) được đánh số khác nhau ở tất cả các điểm thi. Trong một phòng thi, danh sách các thí sinh được sắp xếp theo thứ tự alphabet (do đó trong một phòng thi có thể có thí sinh của nhiều ngành khác nhau). Mỗi phòng thi có thêm cột ghi chú (GHICHU) - ghi thêm các thông tin cần thiết như phòng thi đó nằm tại dãy nhà nào. Ví dụ phòng thi 0060 nằm ở dãy nhà H lầu 2 - điểm thi số 1 - trường PTTH Bùi Thị Xuân.

Mỗi môn thi có một mã môn thi duy nhất (MAMT), mỗi mã môn thi biết các thông tin như: tên môn thi (TENMT), ngày thi (NGAYTHI), buổi thi (BUOITHI), thời gian làm bài thi được tính bằng phút (PHUT). Thời gian làm bài thi của các môn tối thiểu là 90 phút và tối đa là 180 phút (tùy theo kỳ tuyển sinh công nhân, trung cấp, cao đẳng hay đại học)

Mỗi ngành có một mã ngành, chẳng hạn ngành Công Nghệ Thông Tin có mã ngành là 01, ngành Công Nghệ Hoá Thực Phẩm có mã ngành là 10,...

Mỗi đơn vị có cán bộ tham gia vào kỳ thi có một mã đơn vị duy nhất (MADONVI), mã đơn vị xác định tên đơn vị (TENDONVI). Nếu là cán bộ, công nhân viên của trường thì đơn vị là khoa/phòng quản lý cán bộ đó, nếu là giáo viên từ các trường khác thì ghi rõ tên đơn vị đó. Chẳng hạn cán bộ Nguyễn Thanh Liêm đơn vị Khoa Công Nghệ Thông Tin, cán bộ coi thi Nguyễn Thị Tuyết Mai, đơn vị trường PTTH Ngôi Sao - Quận 1,...

Mỗi cán bộ coi thi chỉ làm việc tại một điểm thi nào đó. Mỗi cán bộ có một mã số duy nhất (MACANBO), mỗi MACANBO xác định các thông tin khác như: họ và tên (HOTENCB), đơn vị công tác, chức vụ (CHUCVU) được phân công tại điểm thi, chẳng hạn chức vụ là điểm trưởng, điểm phó, giám sát, thư ký, cán bộ coi thi, phục vụ,... Ví dụ cán bộ Nguyễn Văn Thanh đơn vị Khoa Công Nghệ Thông Tin, làm nhiệm vụ thi tại điểm thi số 1, chức vụ là giám sát phòng thi.

## CHƯƠNG 2. MÔ HÌNH DỮ LIỆU QUAN HỆ

### I. CÁC KHÁI NIỆM CƠ BẢN

Mô hình dữ liệu quan hệ (Relational Data Model)- gọi tắt là mô hình quan hệ, do EF.Codd đề xuất năm 1970. Nền tảng lý thuyết của nó là khái niệm lý thuyết tập hợp trên các quan hệ, tức là tập của các bộ giá trị.

Mô hình dữ liệu quan hệ là mô hình được nghiên cứu nhiều nhất, và thực tiễn đã cho thấy rằng nó có cơ sở lý thuyết vững chắc nhất. Mô hình dữ liệu này cùng với mô hình thức thể kết hợp đang được sử dụng rộng rãi trong việc phân tích và thiết kế CSDL hiện nay.

Dưới đây trình bày các khái niệm của mô hình dữ liệu quan hệ.

#### 1. Thuộc tính (Attribute)

Thuộc tính là các đặc điểm riêng của một đối tượng (đối tượng được hiểu như là một loại thực thể ở mô hình thực thể-kết hợp), mỗi thuộc tính có một tên gọi và phải thuộc về một kiểu dữ liệu nhất định.

##### 1.1. Kiểu dữ liệu (Data type)

Các thuộc tính được phân biệt qua tên gọi và phải thuộc một kiểu dữ liệu nhất định (số, chuỗi, ngày tháng, logic, hình ảnh,...). Kiểu dữ liệu ở đây có thể là kiểu vô hướng hoặc là kiểu có cấu trúc. Nếu thuộc tính có kiểu dữ liệu là vô hướng thì nó được gọi là thuộc tính đơn hay thuộc tính nguyên tố, nếu thuộc tính có kiểu dữ liệu có cấu trúc thì ta nói rằng nó không phải là thuộc tính nguyên tố

Chẳng hạn với sinh viên Nguyễn Văn Thành thì các thuộc tính họ và tên, mã số sinh viên thuộc kiểu chuỗi, thuộc tính ngày sinh thuộc kiểu ngày tháng, hộ khẩu thường trú thuộc kiểu chuỗi, thuộc tính hình ảnh thuộc kiểu hình ảnh,...

##### 1.2. Miền giá trị (Domain of values)

Thông thường mỗi thuộc tính chỉ chọn lấy giá trị trong một tập con của kiểu dữ liệu và tập hợp con đó gọi là miền giá trị của thuộc tính đó. Chẳng hạn thuộc tính NỮ có miền giá trị là {nam,nữ}, thuộc tính màu da có miền giá trị là {da trắng, da vàng, da đen, da đỏ}, thuộc tính điểm thi là các số thuộc tập {0; 1; 2;...,10}.

Lưu ý rằng nếu không lưu ý đến ngữ nghĩa thì tên của các thuộc tính thường được ký hiệu bằng các chữ cái in hoa đầu tiên trong bảng chữ cái la tinh: A,B,C,D,... Những chữ cái in hoa X,Y,Z,W,... thường dùng thay cho một nhóm nhiều thuộc tính. Đôi khi còn dùng các ký hiệu chữ cái với các chỉ số  $A_1, A_2, \dots, A_n$  để chỉ các thuộc tính trong trường hợp tổng quát hay muốn đề cập đến số lượng các thuộc tính. Tên thuộc tính phải được đặt một cách gọi nhớ, không nên đặt tên thuộc tính quá dài (vì như thế sẽ làm cho việc viết các câu lệnh truy vấn trở nên vất vả hơn), nhưng cũng không nên đặt tên thuộc tính quá ngắn (vì nó sẽ không cho thấy ngữ nghĩa của thuộc tính), đặc biệt không đặt trùng tên hai thuộc tính mang ngữ nghĩa khác nhau thuộc hai đối tượng khác nhau.

Trong nhiều hệ quản trị cơ sở dữ liệu, người ta thường đưa thêm vào miền giá trị của các thuộc tính một giá trị đặc biệt gọi là giá trị rỗng (NULL). Tùy theo ngữ cảnh mà giá trị này có thể đặc trưng cho một giá trị không xác định hoặc chưa xác định ở vào thời điểm nhập tin nhưng có thể xác định vào một thời điểm khác.

## 2. Lược đồ quan hệ (Relation schema)

Tập tất cả các thuộc tính cần quản lý của một đối tượng cùng với các mối liên hệ giữa chúng được gọi là lược đồ quan hệ. Lược đồ quan hệ Q với tập thuộc tính  $\{A_1, A_2, \dots, A_n\}$  được viết là  $Q(A_1, A_2, \dots, A_n)$ , ký hiệu  $Q^+ = \{A_1, A_2, \dots, A_n\}$ .

Chẳng hạn lược đồ quan hệ Sinhviên với các thuộc tính như đã được liệt kê trong ví dụ 1.2 được viết như sau:

Sinhviên(MASV, HOTENSV, NU, NGAYSINH, NOISINH, TINH, MALOP)

Thường thì khi thành lập một lược đồ quan hệ, người thiết kế gán cho nó một ý nghĩa nhất định, gọi là **tân từ của lược đồ quan hệ**. Chẳng hạn tân từ của lược đồ quan hệ Sinhviên là: “Mỗi sinh viên có mỗi MASV duy nhất. Mỗi MASV xác định các thuộc tính còn lại của sinh viên đó như HOTENSV, NU, NGAYSINH, NOISINH, TINH, MALOP”.

Khi phát biểu tân từ cho một lược đồ quan hệ, người thiết kế cần phải mô tả đầy đủ ý nghĩa để người khác tránh hiểu nhầm. Dựa vào tân từ này, người ta xác định được tập khoá, siêu khoá của lược đồ quan hệ (sẽ được trình bày trong những mục kế tiếp).

Nhiều lược đồ quan hệ cùng nằm trong một hệ thống thông tin được gọi là một lược đồ cơ sở dữ liệu.

Khái niệm lược đồ quan hệ ứng với khái niệm loại thực thể ở mô hình thực thể kết hợp.

### 3. Quan hệ (Relation)

Sự thể hiện của lược đồ quan hệ ở một thời điểm nào đó được gọi là quan hệ, rõ ràng là trên một lược đồ quan hệ có thể xác định nhiều quan hệ. Ta dùng các ký hiệu như  $R, S, Q$  để chỉ các lược đồ quan hệ, còn quan hệ được dùng bởi các ký hiệu là  $r, s, q, \dots$

Về trực quan thì quan hệ (hay bảng quan hệ) như là một bảng hai chiều gồm các dòng và các cột.

Một quan hệ có  $n$  thuộc tính được gọi là quan hệ  $n$  ngôi.

Để chỉ quan hệ  $r$  xác định trên lược đồ quan hệ  $Q$  ta viết  $r(Q)$ .

### 4. Bộ (Tuple)

Mỗi bộ là những thông tin về một đối tượng thuộc một quan hệ, bộ cũng còn được gọi là mẫu tin.

Ta dùng các chữ cái thường (như  $t, \mu, \dots$ ) để biểu diễn bộ trong quan hệ, chẳng hạn để nói  $t$  là một bộ của quan hệ  $r$  thì ta viết  $t \in r$ .

### 5. Siêu khoá – Khoá (Super key- Key)

$S$  là siêu khoá (super key) của  $Q$  nếu với  $r$  là quan hệ bất kỳ trên  $Q$ ,  $t_1, t_2$  là hai bộ bất kỳ thuộc  $r$  thì  $t_1.S \neq t_2.S$ .

Một lược đồ quan hệ có thể có một hoặc nhiều siêu khoá.

Chẳng hạn lược đồ quan hệ Sinhvien ở trên có các siêu khoá là:  $\{MASV, HOTENSV\}, \{MASV, HOTENSV, NU\}, \{MASV, HOTENSV, NU, TINH\}, \dots$

Siêu khoá không chứa một siêu khoá nào khác được gọi là khoá chỉ định, trong trường hợp lược đồ quan hệ có nhiều khoá chỉ định (hay **khóá nội**), thì khoá được chọn để cài đặt gọi là **khóá chính** (trong các phần sau khoá chính được gọi tắt là khoá). Chẳng hạn với lược đồ quan hệ Sinhvien trên có khoá là  $\{MASV\}$ .

*Các thuộc tính khoá được gạch dưới theo kiểu liền nét.*

Một thuộc tính được gọi là thuộc tính **khoá ngoại** nếu nó không là thuộc tính khoá của một lược đồ quan hệ này nhưng lại là thuộc tính khoá của một lược đồ quan hệ khác, chẳng hạn như MALOP là khoá ngoại của lược đồ quan hệ Sinhviên.

*Các thuộc tính khoá ngoại được gạch dưới theo kiểu không liền nét.*

Sinhviên(MASV, HOTENSV, NU, NGAYSINH, TINH, MALOP)

Lớp(MALOP, TENLOP, MAKHOA)

Ý nghĩa thực tế của khoá là dùng để nhận diện một bộ trong một quan hệ, nghĩa là, khi cần tìm một bộ  $t$  nào đó trong quan hệ, ta chỉ dò giá trị khoá của  $t$  là đủ.

Trong thực tế đối với các loại thực thể tồn tại khách quan (ví dụ: Sinh viên, Giảng viên, Nhân viên, Hàng hoá,...) người thiết kế cơ sở dữ liệu thường gán thêm cho các lược đồ quan hệ này một thuộc tính giả gọi là mã số để làm khoá (ví dụ: mã số sinh viên, mã số giảng viên, mã số nhân viên, mã số hàng hoá,...). Trong khi đó các lược đồ quan hệ biểu diễn cho sự trừu tượng hoá thường có khoá là một tổ hợp của hai hay nhiều thuộc tính của nó.

Một số HQTCSDL hiện nay có tự động kiểm tra tính duy nhất trên khoá chính. Tức là nếu thêm một bộ mới  $q_2$  có giá trị khoá chính trùng với giá trị khoá chính của một bộ  $q_1$  nào đó đã có trong quan hệ thì hệ thống sẽ báo lỗi và yêu cầu nhập lại một giá trị khác.

#### **Lưu ý:**

- Trong một bộ của quan hệ, các thuộc tính khoá không chứa giá trị rỗng.
- Không được phép sửa đổi giá trị thuộc tính khoá của một bộ  $q$ . Nếu muốn sửa đổi giá trị thuộc tính khoá của một bộ  $q$ , người sử dụng phải huỷ bỏ bộ  $q$  và sau đó thêm một bộ  $q'$  với giá trị khoá đã được sửa đổi.

## **II. CHUYỂN MÔ HÌNH THỰC THỂ KẾT HỢP SANG MÔ HÌNH DỮ LIỆU QUAN HỆ**

Sau đây là một số quy tắc được sử dụng trong việc chuyển đổi mô hình thực thể kết hợp sang mô hình dữ liệu quan hệ.

### **1. Quy tắc 1**

Chuyển đổi mỗi loại thực thể thành một lược đồ quan hệ, các thuộc tính của loại thực thể thành các thuộc tính của lược đồ quan hệ, thuộc tính khoá của loại thực thể là thuộc tính khoá của lược đồ quan hệ.

Chẳng hạn loại thực thể Sinhvien ở ví dụ 1.2 khi áp dụng quy tắc 1 thì sẽ được chuyển thành lược đồ quan hệ Sinhvien như sau:

Sinhvien(MASV, HOTENSV, NU, NGAYSINH, TINH,...)

## 2. Quy tắc 2

Nếu quan hệ mà cả hai nhánh của nó đều có bản số max là n thì quan hệ này sẽ được chuyển thành một lược đồ quan hệ K' gồm các thuộc tính của quan hệ K, cộng thêm các thuộc tính khoá của hai lược đồ quan hệ A, B tương ứng với hai thực thể tham gia vào quan hệ. Khoá của lược đồ quan hệ K' gồm cả hai khoá của hai lược đồ quan hệ A và B.

Chẳng hạn quan hệ “Phancong” giữa ba loại thực thể Giảngviên, Môn học, Lớp được chuyển thành lược đồ quan hệ Phancong và có tập khoá là {MAGV, MAMH, MALOP} như sau: Phancong(MAGV, MAMH, MALOP).

## 3. Quy tắc 3

Quan hệ mà một nhánh có bản số là n (nhánh B) và nhánh còn lại có bản số max là 1 (nhánh A) thì loại bỏ quan hệ này khỏi mô hình thực thể kết hợp và thêm các thuộc tính khoá của lược đồ tương ứng với loại thực thể ở nhánh B vào lược đồ tương ứng với loại thực thể ở nhánh A (khoá của B sẽ thành khoá ngoại của A). Nếu quan hệ có các thuộc tính thì những thuộc tính này cũng được thêm vào lược đồ quan hệ tương ứng với loại thực thể ở nhánh A.

Chẳng hạn quan hệ “thuộc” giữa hai loại thực thể Sinhviên và Lớp nên lược đồ quan hệ Sinhviên được sửa thành như sau:

Sinhvien(MASV, HOTENSV, NU, NGAYSINH, TINH, MALOP)

## 4. Quy tắc 4

Nếu mỗi kết hợp mà cả hai nhánh đều có bản số max là 1 thì áp dụng quy tắc 3 cho một trong hai nhánh tùy chọn.

**Ví dụ 2.1:**



Sau đây là mô hình dữ liệu quan hệ được chuyển từ mô hình thực thể kết hợp ở ví dụ 1.2.

Sinhviên(MASV, HOTENSV, NU, NGAYSINH, NOISINH, TINH, MALOP)

Lớp(MALOP, TENLOP, MAKHOA)

Khoa(MAKHOA, TENKHOA)

Môn học(MAMH, TENMH, DONVIHT)

Giảngviên(MAGV, HOTENGV, HOCVI, CHUYENNGANH, MAKHOA)

Kếtquả(MASV, MAMH, LANTHI, DIEMTHI)

Phân công(MALOP, MAMH, MAGV)

### III. CÁC PHÉP TOÁN ĐẠI SỐ TRÊN QUAN HỆ

#### 1. Đại số quan hệ

Khi nói tới mô hình quan hệ người ta quan tâm tới 3 vấn đề chính:

- Cấu trúc để lưu trữ dữ liệu
- Các ràng buộc trên dữ liệu
- Các phép toán để thao tác dữ liệu

Như đã trình bày trong phần đầu của chương này về mô hình quan hệ, cấu trúc lưu trữ dữ liệu trong mô hình quan hệ chính là cấu trúc Bảng (Table). Các ràng buộc trên dữ liệu chúng ta cũng đã tìm hiểu như ràng buộc khóa chính, ràng buộc khóa ngoại và còn các loại ràng buộc khác sẽ được trình bày chi tiết trong **chương 4**.

*Tập hợp các phép toán để thao tác với dữ liệu trong mô hình quan hệ được gọi là đại số quan hệ.* Các phép toán này giúp cho người sử dụng xác định rõ các yêu cầu lấy tin cơ bản, kết quả của một phép lấy tin là một quan hệ mới, có thể được tạo ra từ một hoặc nhiều quan hệ. Một dãy các phép toán trên quan hệ tạo nên một biểu thức đại số quan hệ mà kết quả của nó cũng là một quan hệ.

Các phép toán đại số quan hệ được chia thành hai nhóm. Một nhóm bao gồm các *phép toán tập hợp* lấy từ lý thuyết tập hợp trong toán học. Nhóm kia bao gồm các *phép toán quan hệ* được xây dựng đặc biệt cho cơ sở dữ liệu quan hệ.

## 2. Các phép toán tập hợp

Cho hai lược đồ quan hệ  $Q_1$  và  $Q_2$  có cùng tập thuộc tính  $\{A_1, A_2, \dots, A_n\}$ .  $r_1, r_2$  lần lượt là hai quan hệ trên  $Q_1$  và  $Q_2$ , ta nói hai quan hệ tương thích nếu chúng được định nghĩa trên cùng một lược đồ quan hệ hoặc trên hai lược đồ quan hệ có cùng tập thuộc tính. Các phép toán sau đây chỉ được xét trên quan hệ tương thích, đó là: phép hợp, phép giao, phép trừ.

### 2.1. Phép Hợp 2 quan hệ (Union)

Cho hai quan hệ tương thích  $r_1$  và  $r_2$ . Hợp của hai quan hệ  $r_1$  và  $r_2$  ký hiệu là  $r_1 + r_2$  là một quan hệ trên lược đồ quan hệ  $Q$  gồm các phần tử thuộc  $r_1$  hoặc thuộc  $r_2$ , tức là:  $r_1 + r_2 = \{t / t \in r_1 \text{ hoặc } t \in r_2\}$

#### Ví dụ 2.1

$r_1$

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

$r_2$

A	B	C	D
x1	y1	z1	v1
a2	b2	c2	d2
x3	y3	z3	v3

Khi đó, nội dung của quan hệ  $r_1 + r_2$  là:

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4
x1	y1	z1	v1
x3	y3	z3	v3

Do thứ tự trước/sau của các bộ trong các quan hệ là không quan trọng nên ta có:

$$r_1 + r_2 = r_2 + r_1$$

$$\text{Và } r + r = r$$

Một cách tổng quát có thể lấy hợp của  $n$  quan hệ tương thích: cho  $n$  quan hệ tương thích  $r_1, r_2, \dots, r_n$

Hợp của  $n$  quan hệ  $r_1, r_2, \dots, r_n$  là một quan hệ  $r_1 + r_2 + \dots + r_n$  gồm các phần tử thuộc  $r_1$  hoặc thuộc  $r_2 \dots$  hoặc thuộc  $r_n$

### 2.2. Phép Giao 2 quan hệ (Intersection)

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ .  $r_1$  và  $r_2$  là hai quan hệ tương thích trên  $Q$ .

Giao của hai quan hệ tương thích  $r_1$  và  $r_2$  ký hiệu là  $r_1 * r_2$  là một quan hệ trên  $Q$  gồm các phần tử vừa thuộc  $r_1$  vừa thuộc  $r_2$ .

Vậy:  $r_1 * r_2 = \{ t / t \in r_1 \text{ và } t \in r_2 \}$

Chẳng hạn với ví dụ 2.1 ở trên thì  $r_1 * r_2$  là:

A	B	C	D
a2	b2	c2	d2

### 2.3. Phép Trừ 2 quan hệ (Minus)

Cho hai quan hệ tương thích  $r_1$  và  $r_2$  có tập thuộc tính  $Q(A_1, A_2, \dots, A_n)$ . Hiệu của hai quan hệ tương thích  $r_1$  cho  $r_2$  ký hiệu là  $r_1 - r_2$  là một quan hệ trên  $Q$  gồm các phần tử chỉ thuộc  $r_1$  mà không thuộc  $r_2$ , nghĩa là  $r_1 - r_2 = \{ t \in r_1 \text{ và } t \notin r_2 \}$ .

Chẳng hạn với ví dụ 2.1. thì  $r_1 - r_2$  là:

A	B	C	D
a1	b1	c1	d1
a3	b3	c3	d3
a4	b4	c4	d4

### 2.4. Tích Decac của 2 quan hệ (Cartesian Product)

Cho hai lược đồ quan hệ  $Q_1(A_1, A_2, \dots, A_n)$ ,

$Q_2(B_1, B_2, \dots, B_m)$ .

Giả sử  $r_1, r_2$  là hai quan hệ trên  $Q_1, Q_2$  tương ứng. Tích Descartes (decac) của  $r_1$  và  $r_2$  ký hiệu là  $r_1 \times r_2$  là quan hệ trên lược đồ quan hệ có tập thuộc tính  $Q = Q_1 \cup Q_2$ .

Vậy quan hệ  $r_1 \times r_2$  là quan hệ trên lược đồ:

$Q = Q_1 \cup Q_2 = \{ A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m \}$  với

$$r_1 \times r_2 = \{(t_1, t_2) : t_1 \in r_1, t_2 \in r_2\}$$

**Ví dụ 2.2:** cho  $r_1$

$r_1$

$r_2$

và  $r_2$  là

Thì kết quả  $r_1 \times r_2$

như sau:

A	B	C
6	5	4
7	5	5

E	F	H
1	5	9
4	6	8
7	5	3

A	B	C	E	F	H
6	5	4	1	5	9
6	5	4	4	6	8
6	5	4	7	5	3
7	5	5	1	5	9
7	5	5	4	6	8
7	5	5	7	5	3

## 2.5. Phép chia 2 quan hệ

Cho 2 lược đồ quan  $Q_1(A_1, A_2, \dots, A_n)$ ,  $Q_2(B_1, B_2, \dots, B_m)$ .

$r$  là quan hệ xác định trên  $Q_1$ ;  $s$  là quan hệ xác định trên  $Q_2$  ( $n > m$  và  $s$  khác rỗng), có  $m$  thuộc tính chung (giống nhau về mặt ngữ nghĩa, hoặc các thuộc tính có thể so sánh được) giữa  $r$  và  $s$ . Phép chia 2 quan hệ  $r$  và  $s$  ký hiệu  $r \div s$ , là một quan hệ  $q$  có  $n - m$  thuộc tính được định nghĩa như sau:

$$q = r \div s = \{t / \forall u \in s, (t, u) \in r\}$$

**Ví dụ 2.3:**

$r$

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
c	d	c	d
c	d	e	f
a	b	d	e

$s$

C	D
c	d
e	f

thì

$r \div s$

A	B
a	b
c	d

### 3. Các phép toán quan hệ

#### 3.1. Phép Chiếu (Projection)

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ ,  $r$  là quan hệ trên  $Q$  và  $X \subseteq Q^+$ .

Phép chiếu của  $r$  lên tập thuộc tính  $X$ , ký hiệu là  $r[X]$  (hoặc  $r.X$ ) sẽ tạo thành lược đồ quan hệ  $r'$ , trong đó tập thuộc tính của  $r'$  chính là  $X$  và quan hệ  $r'$  được trích từ  $r$  bằng cách chỉ lấy các thuộc tính có trong  $X$ .

Phép chiếu chính là phép rút trích dữ liệu theo cột. Chẳng hạn với  $r_1$  ở ví dụ 2.1 thì khi đó ta có quan hệ con của  $r_1$  chiếu lên  $X = \{A, C\}$  là:

$r_1[X]$

A	C
a1	c1
a2	c2
a3	c3
a4	c4

#### 3.2. Phép Chọn (Selection)

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ ,  $r$  là một quan hệ trên lược đồ quan hệ  $Q$ .  $X$  là một tập con của  $Q^+$  và  $E$  là một mệnh đề logic được phát biểu trên tập  $X$ . Phần tử  $t \in r$  thoả mãn điều kiện  $E$  ký hiệu là  $t(E)$ . Phép chọn từ quan hệ  $r$  theo điều kiện  $E$  (ký hiệu là  $r : E$ ) sẽ tạo thành một quan hệ mới ký hiệu là  $r(E)$ , trong đó  $r(E) = \{t: t \in r \text{ và } t(E)\}$

Phép chọn chính là phép rút trích dữ liệu theo dòng. Chẳng hạn với  $r_2$  ở ví dụ 2.2 và điều kiện  $E$  là: " $F \geq 6$ " thì kết quả  $r_2(E)$  hay  $r_2: "F \geq 6"$  có nội dung là

E	F	H
4	6	8

#### 3.3. Phép $\theta$ - kết

Cho hai lược đồ quan hệ  $Q_1$  và  $Q_2$  như sau:

$Q_1(A_1, A_2, \dots, A_n)$

$Q_2(B_1, B_2, \dots, B_m)$

$r$  và  $s$  lần lượt là hai quan hệ trên  $Q_1$  và  $Q_2$ .

$A_i$  và  $B_j$  lần lượt là thuộc tính của  $Q_1, Q_2$  sao cho  $MGT(A_i) = MGT(B_j)$ .  $\theta$  là một trong các phép so sánh ( $=, <, >, \leq, \geq, \neq$ ) trên  $MGT(A_i)$ .

Phép  $\theta$  kết giữa  $r$  và  $s$  theo điều kiện  $A_i \theta B_j$  ký hiệu là  $A_i \theta B_j$  là một quan hệ trên lược đồ quan hệ có tập thuộc tính là  $Q_1 \cup Q_2$ . gồm những bộ thuộc tích Descartes của  $r$  và  $s$  sao  $A_i \theta B_j$ .

$$r \bowtie_{A_i \theta B_j} s = \{t_{12} / \exists t_1 \in r_1, \exists t_2 \in r_2 \text{ sao cho } t_{12}.Q_1^+ = t_1; t_{12}.Q_2^+ = t_2; t_{12}.A_i \theta t_{12}.B_j\}$$

**Ví dụ 2.4** Cho hai quan hệ  $r_1$  và  $r_2$  như sau:

$r_1$			$r_2$		
A	B	C	E	F	H
6	5	4	1	5	9
7	5	5	4	6	8
4	2	6	7	5	3

$A_i$  là thuộc tính B,  $B_j$  là thuộc tính F và  $\theta$  là phép so sánh “ $\geq$ ”, điều kiện  $B \geq F$ , ta được kết quả là quan hệ sau:

A	B	C	E	F	H
6	5	4	1	5	9
6	5	4	7	5	3
7	5	5	1	5	9
7	5	5	7	5	3

### 3.4. Phép Kết Tự Nhiên (Natural join)

Nếu  $\theta$  được sử dụng trong phép kết trên là phép so sánh bằng (=) thì gọi là phép kết bằng. Hơn nữa nếu  $A_i \equiv B_j$  thì phép kết bằng này được gọi là phép kết tự nhiên. Phép kết tự nhiên là phép kết thường dùng nhất trong thực tế.

## 4. Ví dụ về ngôn ngữ đại số quan hệ.

**Ví dụ 2.5:** Cho lược đồ CSDL dùng để quản lý điểm sinh viên được mô tả như ở ví dụ 1.3. Hãy thực hiện các yêu cầu sau bằng ngôn ngữ đại số quan hệ:

1. Lập danh sách các sinh viên lớp có mã lớp là CDTH2A, danh sách cần MASV, HOTENSV
2. Lập danh sách sinh viên nữ và có mã khoa là “CNTT”, danh sách cần MASV, HOTENSV.
3. Lập bảng điểm thi lần 1 của tất cả các môn cho sinh viên lớp CDTH2A, danh sách cần MASV, HOTENSV, TENMH, DIEMTHI.

Ví dụ về ngôn ngữ đại số quan hệ.

4. Lập phiếu điểm thi lần 1 các môn cho sinh viên có MASV="00CDTH189". danh sách cần MAMH, TENMH, DONVIHT, DIEMTHI.

Giải:

1. `Sinhvien: MALOP="CDTH2A" [MASV, HOTENSV]`  
`MALOP`
2. `(Sinhvien |><| Lop: NU and MAKHOA="CNTT")`  
`[MASV, HOTENSV]`  
`MASV`      `MAMH`
3. `((Sinhvien |><| Ketqua ) |><| Monhoc): MALOP = "CDTH2A" and`  
`LANTHI=1) [MASV, HOTENSV, TENMH, DIEMTHI]`  
`MAMH`
4. `(Ketqua |><| Monhoc : MASV='00CDTH189' and LANTHI=1)`  
`[MAMH, TENMH, DONVIHT, DIEMTHI]`

#### IV. BÀI TẬP

**Bài tập 2.1.** Hãy lập mô hình dữ liệu quan hệ

Hãy lập mô hình dữ liệu quan hệ cho các bài toán quản lý 1.1, 1.2, 1.3, 1.4, 1.5 trong phần bài tập chương 1. Hãy xác định khoá cho từng lược đồ cho mỗi bài toán trên.

**Bài tập 2.2.** Cho lược đồ cơ sở dữ liệu

Sinhvien(MASV, HTENSV, 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)

Thực hiện các yêu cầu sau bằng ngôn ngữ đại số quan hệ:

a. Lập danh sách những sinh viên có hộ khẩu thường trú ở tỉnh "LONG AN", danh sách cần các thông tin: MASV, HOTENSV, NGAYSINH, TENLOP

b. Lập danh sách các sinh viên của lớp có MALOP là CDTH2A, danh sách cần các thông tin: MASV, HOTENSV, NGAYSINH, TINH.

c. Lập danh sách các giảng viên có cấp học vị là THAC SY của khoa có MAKHOA là “CNTT”, danh sách cần: MAGV, HOTENGV, CHUYENNGANH.

d. Lập bảng điểm thi lần 1 môn học “869” cho tất cả sinh viên thuộc hai lớp có MALOP là “CDTH2A” và “CDTH2B”, danh sách cần: MASV, HOTENSV, DIEMTHI.

e. Lập danh sách các giảng viên đã dạy lớp CDTH2A, danh sách cần các thông tin: MAGV, HOTENGV, TENKHOA, HOCVI, TENMH.

f. Lập danh sách các môn mà lớp CDTH2A đã học, danh sách cần các thông tin: MAMH, TENMH, DONVIHT, HOTENGV.

g. Lập danh sách những giảng viên đã dạy sinh viên có MASV là “00CDTH189”, danh sách cần MAGV, HOTENGV, HOCVI, CHUYENNGANH, TENKHOA, TENMH.

h. Lập danh sách các sinh viên có mã khoa “CNTT” có điểm thi lần 1 môn học “869” lớn hơn hoặc bằng 8, danh sách cần MASV, HOTENSV, DIEMTHI, TENLOP.



## CHƯƠNG 3. NGÔN NGỮ SQL

### I. TỔNG QUAN

#### 1. Lịch sử phát triển

Vào những năm 1970, SQL (Structure Query Language) lần đầu được hãng IBM phát triển như một bộ phận của hệ quản trị CSDL mô hình quan hệ có tên là SYSTEM R. Sau đó vào các năm 1980 IBM tiếp tục phát triển SQL cho các hệ quản trị cơ sở dữ liệu nổi tiếng là SQL/DS trên nền hệ điều hành VM, DB2 trên nền hệ điều hành MVS, Hệ quản trị cơ sở dữ liệu mở rộng trên nền hệ điều hành IBM OS/2, Hệ quản trị cơ sở dữ liệu cho hệ thống IBM AS/400. Năm 1986, Viện tiêu chuẩn quốc gia Hoa kỳ (ANSI – American National Standards Institute) và Tổ chức Tiêu chuẩn Quốc tế (ISO – International Standards Organization) đã thừa nhận SQL như là *ngôn ngữ chuẩn xử lý dữ liệu*. Ngôn ngữ chuẩn ANSI SQL tiếp tục được cập nhật vào những năm 1989 và 1992 cho đến hôm nay.

SQL được cài đặt cho hệ thống máy tính lớn (mainframe) cũng như máy tính cá nhân. Bên cạnh các sản phẩm của hãng IBM, cũng cần phải kể đến các hệ quản trị cơ sở dữ liệu nổi tiếng khác như ORACLE của Công ty Oracle, SQL Server của hãng Microsoft, SQLBase của hãng Sybase, Ingres của hãng Relational Technologies, ...

#### 2. SQL là ngôn ngữ chuẩn cho các hệ quản trị cơ sở dữ liệu quan hệ

– Mục đích của chuẩn SQL

- + Xác định cú pháp và ngữ nghĩa của ngôn ngữ SQL, định nghĩa và thao tác dữ liệu.
- + Định nghĩa các cấu trúc dữ liệu và phép toán cơ bản để thiết kế, truy cập, lưu trữ, kiểm soát và bảo vệ cơ sở dữ liệu SQL.
- + Cung cấp công cụ đảm bảo tính tương thích của cấu trúc dữ liệu và các modul ứng dụng giữa các hệ quản trị cơ sở dữ liệu.
- + Xác định chuẩn tối thiểu (mức 1) và chuẩn hoàn chỉnh (mức 2), cho phép các cấp độ sử dụng SQL khác nhau trong các sản phẩm.

- + Cung cấp chuẩn ban đầu, có thể chưa hoàn chỉnh, cho phép mở rộng các chức năng xử lý những vấn đề như sự toàn vẹn tham chiếu, giao thức chuyển đổi, các hàm người dùng, các toán tử nổi ngoài phép đẳng nổi, và các hệ thống ký tự quốc gia,...
- Một ngôn ngữ quan hệ chuẩn như SQL sẽ mang lại các lợi ích sau đây:
  - + Giảm thiểu chi phí đào tạo. Các hệ quản trị cơ sở dữ liệu có chung ngôn ngữ chuẩn SQL sẽ làm giảm chi phí chuyển đổi từ hệ này sang hệ khác.
  - + Nâng cao hiệu năng công việc. Các chuyên gia hệ thống thông tin với kiến thức sâu sắc về SQL sẽ nhanh chóng nắm bắt các chương trình ứng dụng của các hệ quản trị cơ sở dữ liệu, vì họ đã quen thuộc với ngôn ngữ của các chương trình này.
  - + Tính khả chuyển của các ứng dụng. Các ứng dụng có thể dễ dàng sử dụng trên các hệ thống khác nhau, nhưng cùng sử dụng SQL.
  - + Tăng tuổi thọ của các ứng dụng. Một ngôn ngữ chuẩn có xu hướng tồn tại thời gian dài, điều đó làm giảm áp lực viết lại chương trình.
  - + Làm giảm sự phụ thuộc vào nhà cung cấp. Vì SQL là ngôn ngữ chung nên người dùng dễ dàng sử dụng nhiều sản phẩm của các nhà cung ứng khác nhau, với giá cả cạnh tranh.
  - + Khả năng giao tiếp giữa các hệ thống chéo. Các hệ quản trị cơ sở dữ liệu quan hệ và các chương trình ứng dụng khác nhau có thể dễ dàng giao tiếp và hợp tác để xử lý dữ liệu và thực hiện chương trình người dùng.
- SQL là ngôn ngữ có cấu trúc. Trong câu lệnh của SQL có một số mệnh đề tuân theo những cú pháp riêng của nó. Có 4 loại lệnh trong SQL:
  - + Các lệnh truy vấn dữ liệu (**DQL – Data Query Language**). Là những câu lệnh truy vấn dữ liệu nhưng không làm thay đổi dữ liệu hay cấu trúc các bảng trong CSDL. Chỉ có duy nhất lệnh SELECT.
  - + Các lệnh định nghĩa dữ liệu (**DDL – Data definition language**). Tạo bảng và chỉnh sửa cấu trúc bảng trong CSDL. Có các lệnh CREAT, ALTER, DROP.

- + Các lệnh xử lý cập nhật dữ liệu (**DML – Data manipulation language**). Thực hiện thay đổi dữ liệu lưu trong các bảng của CSDL. Có các lệnh INSERT, UPDATE, DELETE.
- + Các lệnh kiểm soát dữ liệu (**DCL – Data Control Language**). Quản lý *quyền* sử dụng CSDL và các bảng lưu trong CSDL của người dùng. Có các lệnh GRANT và REVOKE.

### 3. Các thành phần cơ bản của SQL

#### 3.1. Kiểu dữ liệu

SQL có các kiểu dữ liệu sau:

INTEGER : Kiểu số nguyên 2 byte -2 147 483 648 → 2 147 483 647 .

SMALLINT : Kiểu số nguyên 1 byte - 32 768 → 32 767.

DECIMAL( $n,d$ ) : Kiểu số thực độ dài  $n$  (kể cả dấu), số chữ số thập phân  $d$

FLOAT( $n,d$ ) : Kiểu số thực khoa học độ dài  $n$ , số chữ số thập phân  $d$

CHAR( $n$ ) : Kiểu chuỗi ký tự độ dài  $n$

DATE : Kiểu ngày tháng (ngày/tháng/năm)

LOGICAL : Kiểu logic, nhận giá trị *true* hoặc *false*

VARCHAR( $n$ ) : Kiểu chuỗi ký tự độ dài thay đổi, tối đa  $n$  ký tự

LONGVARCHAR: Kiểu chuỗi ký tự độ dài thay đổi (chứa dữ liệu như *ghi chú*,...)

Và các kiểu khác như: Kiểu tiền tệ, Kiểu hình ảnh, Kiểu âm thanh...

Với các kiểu dữ liệu, ngoài miền giá trị quy định, có thể nhận giá trị **NULL**: nghĩa là *không xác định, chưa sử dụng được*. Trường khóa không được có giá trị NULL.

#### 3.2. Các toán tử

Toán tử số học		Toán tử logic		Toán tử tập hợp	
+	Cộng	OR	Hoặc	UNION	Hợp
-	Trừ	AND	Và	INTERSECT	Giao
*	Nhân	NOT	Phủ định	MINUS	Hiệu

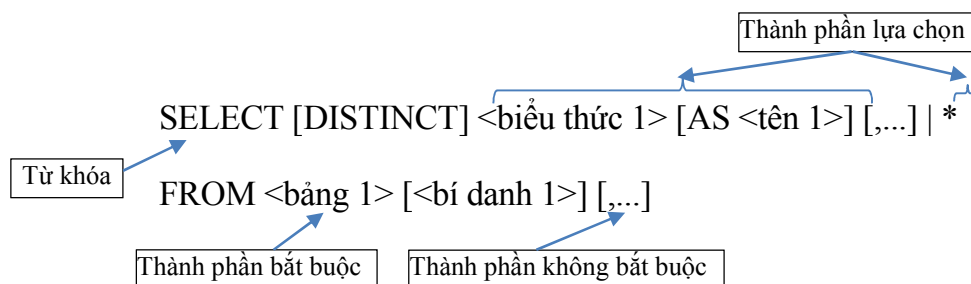
/	Chia		
Toán tử so sánh		Hàm tính toán	
>	Lớn hơn	COUNT()	Đếm số bản ghi
>=	Lớn hơn hoặc bằng	SUM()	Tính tổng
<	Nhỏ hơn	AVG()	Tính trị trung bình
<=	Nhỏ hơn hoặc bằng	MAX()	Tìm trị lớn nhất
=	Bằng	MIN()	Tìm trị nhỏ nhất
<>	Khác	Hàm tập hợp	
		SET()	Tập hợp các phần tử

#### 4. Các quy ước biểu diễn câu lệnh SQL

Câu lệnh gồm: từ khóa, thành phần bắt buộc, thành phần không bắt buộc, thành phần lựa chọn.

- **Từ khóa** là từ phải có trong câu lệnh, được quy định sẵn, không dùng đặt tên cho trường hay bảng. Trong giáo trình này TỪ KHÓA được viết in hoa.
- **Thành phần bắt buộc** là biểu thức phải có trong câu lệnh. Trong giáo trình này <thành phần bắt buộc> được đặt trong dấu ngoặc nhọn <...>.
- **Thành phần không bắt buộc** là biểu thức có thể có/ không trong câu lệnh. Trong giáo trình này [thành phần không bắt buộc] được đặt trong dấu ngoặc vuông [...].
- **Thành phần lựa chọn** được phân cách bởi dấu |. Nếu được chọn thì chỉ một trong các thành phần trên được xuất hiện trong câu lệnh.

#### Ví dụ



## II. CÁC LỆNH ĐỊNH NGHĨA DỮ LIỆU (DDL)

### 1. Tạo một cơ sở dữ liệu

– **Cú pháp:**

```
Create Database <Tên CSDL>
```

**Ví dụ:** Tạo một cơ sở dữ liệu có tên là QLTV (Quản lý thư viện)

```
Create Database QLTV
```

## 2. Tạo một bảng - CREATE

### 2.1. Cú pháp

```
CREATE TABLE <Tên bảng>  
  
(<thuộc tính 1> < Kiểu dữ liệu thuộc tính 1> ,  
  
<thuộc tính 2> < Kiểu dữ liệu thuộc tính 2> ,  
  
.....  
  
<thuộc tính n> < Kiểu dữ liệu thuộc tính n> )
```

*Ngoài ra khi tạo bảng có thể khai báo các ràng buộc dữ liệu sẽ được trình bày ở phần sau.*

**Ví dụ:**

Tạo bảng DOCGIA, có các thuộc tính:

```
CREATE TABLE DOCGIA  
  
(MaDG CHAR(10) NOT NULL,  
  
TenDG CHAR(30) DEFAULT 'Nguyễn Văn An',  
  
DiaChi CHAR(50) DEFAULT 'Đà Nẵng',  
  
Tuoi INTEGER(4))
```

*Khi mới được tạo, bảng DOCGIA chưa có dữ liệu, chỉ là một cấu trúc logic có thể tiếp nhận các dữ liệu.*

### 2.2. Các loại ràng buộc trong bảng

– **NOT NULL** (Không rỗng): Khi có mệnh đề NOT NULL có trong định nghĩa của một cột thì bắt buộc thuộc tính này phải có giá trị. Nếu ta không chỉ thị gì trong định nghĩa của thuộc tính thì nó có thể có hoặc không có giá trị.

**Ví dụ:**

```
CREATE TABLE NHANVIEN  
  
( MaNV INTEGER (10) NOT NULL,  
  
TenNV CHAR(30) )
```

- **UNIQUE** (Duy nhất): chỉ ra rằng buộc duy nhất, dữ liệu của cột phải có giá trị khác biệt. Giá trị NULL được cho phép.

**Ví dụ:**

```
CREATE TABLE NHANVIEN (  
  
MaNV INTEGER (10) NOT NULL,  
  
TenNV CHAR(30) ,  
  
DiachiNV CHAR(50)  
  
CONSTRAINT UNQ_Ten_Diachi UNIQUE(Ten,Diachi))
```

- **PRIMARY KEY** (Khoá chính). Chỉ ra rằng buộc duy nhất (giống UNIQUE), tuy nhiên khoá là dạng khoá UNIQUE cấp cao nhất. Một table chỉ có thể có một PRIMARY KEY. Các giá trị trong PRIMARY KEY phải NOT NULL.

**Ví dụ:** Tạo khóa chính cho thuộc tính MaNV

```
CREATE TABLE NHANVIEN  
  
( MaNV char(10) NOT NULL primary key,  
  
TenNV char(30) ,  
  
DiachiNV char(50) )
```

Hoặc ta có thể viết câu lệnh sau:

```
CREATE TABLE NHANVIEN  
  
( MaNV char(10) NOT NULL,  
  
TenNV char(30) ,  
  
DiachiNV char(50) ,  
  
CONSTRAINT NV_P_K PRIMARY KEY (MaNV) )
```

- **FOREIGN KEY** (Khoá ngoại). Chỉ ra mối liên hệ ràng buộc tham chiếu giữa bảng này với bảng khác. Từ khoá ON DELETE CASCADE được chỉ định trong dạng khoá này để chỉ khi dữ liệu cha bị xoá thì dữ liệu con cũng tự động bị **xoá theo**.

**Ví dụ:** Hai bảng DONVI và bảng NHANVIEN có mối quan hệ cha – con (1\_N). Thuộc tính MaDV trong bảng NHANVIEN(bảng con) là khoá ngoại, được tham chiếu từ thuộc tính MaDV của bảng DONVI(bảng cha)

Ta tạo 2 bảng như sau:

```
CREATE TABLE DONVI
```

```
( MaDV char(2) primary key,
```

```
TenDV char(20)not null )
```

```
CREATE TABLE NHANVIEN
```

```
(MaNV char(10) primary key,
```

```
TenNV char(30)not null,
```

```
Diachi char(50),
```

```
madv char(2) CONSTRAINT k_n_madv FOREIGN KEY (madv)  
REFERENCES DONVI (MaDV) )
```

- |   |
|---|
| <ul style="list-style-type: none"><li>– Trong bảng DONVI: MaDV là khóa chính</li><li>– Trong bảng NHANVIEN:<ul style="list-style-type: none"><li>+ MaNV là khóa chính</li><li>+ madv là khóa ngoại liên kết tới MaDV của bảng DONVI</li></ul></li></ul> |
|---|

- **CHECK** (Ràng buộc kiểm tra giá trị). Ràng buộc CHECK được sử dụng để yêu cầu các giá trị trong cột, hoặc khuôn dạng dữ liệu trong cột phải theo một quy tắc nào đó. Trên một cột có thể có nhiều ràng buộc này.

**Ví dụ:** Yêu cầu giá trị Lương nhập vào phải trên 500

```
CREATE TABLE NHANVIEN
```

```
(MaNV CHAR(10) NOT NULL PRIMARY KEY,
```

```
TenNV CHAR(30),
```

```
Luong NUMBER(10,2) CONSTRAINT CK_SAL CHECK (SAL>500) )
```

- **DEFAULT** (Mặc định). Ràng buộc DEFAULT được sử dụng để quy định giá trị mặc định cho một cột. Giá trị này sẽ tự động gán cho cột nếu người sử dụng không nhập vào khi bổ sung bản ghi.

**Ví dụ:**

```
CREATE TABLE NHANVIEN  
  
( MaNV char(10) primary key,  
  
TenNV char(30)not null,  
  
Gioitinh char(3) DEFAULT 'Nam' )
```

### 3. Sửa đổi cấu trúc – ALTER

Có thể sửa đổi cấu trúc của bảng hiện đang tồn tại bằng lệnh ALTER. Có thể thêm một thuộc tính (cột) mới, thay đổi cấu trúc của một thuộc tính đang có, bổ sung khoá, bổ sung ràng buộc. Chú ý các ví dụ sau.

- Thêm một ràng buộc CHECK

```
ALTER TABLE DONVI  
  
ADD CONSTRAINTcheck_madv  
  
CHECK (MaDV LIKE '[0-9][0-9]')
```

- Thêm một thuộc tính.

```
ALTER TABLE NHANVIEN  
  
ADD COLUMN GhiChu VARCHAR (50)
```

- Thay đổi kiểu của một thuộc tính.

```
ALTER TABLE NHANVIEN  
  
ALTER COLUMN GhiChu VARCHAR(20)
```

- Xóa một thuộc tính

```
ALTER TABLE NHANVIEN  
  
DROP COLUMN GhiChu
```

### 4. Xoá đối tượng – DROP

- Cú pháp:

DROP <Object_name>
--------------------

**Ví dụ:** Xóa bảng SINHVIENT

```
DROP TABLE SINHVIENT
```



### III. CÁC LỆNH XỬ LÝ CẬP NHẬT DỮ LIỆU (DML)

#### 1. Thêm hàng – INSERT

– Cú pháp:

```
INSERT [INTO]<TableName> (Column1, Column2, ..., Columnn)
VALUES (Values1,Values2,....,Valuesn)
```

Lệnh này được dùng để chèn thêm một hoặc nhiều dòng (bản ghi) mới vào một bảng. Dạng đơn giản nhất của lệnh này là thêm mỗi lần 1 dòng. Nó đòi hỏi phải nêu tên của bảng, tên các thuộc tính và giá trị cần gán cho chúng. Nếu không nêu tên các thuộc tính thì điều đó có nghĩa là tất cả các thuộc tính trong bảng đều cần được thêm giá trị theo thứ tự từ trái sang phải.

**Ví dụ:** Giả sử ta đã có cấu trúc bảng NHANVIEN(MaNV, TenNV, Diachi, Tuoi)

– Thêm bản ghi mới vào tất cả các thuộc tính trong bảng.

```
INSERT INTO NHANVIEN
VALUES ( 'DHTL05', 'Nguyễn Công Thành', 'KhoaCNTT', 22 )
```

– Thêm bản ghi mới vào bảng, mục Tuoi không cần:

```
INSERT INTO DOCGIA (MaDG, TenDG, DiaChi)
VALUES ( 'DHTL06', 'Nguyễn Phương Lan', 'Khoa May' )
```

Ngoài ra chúng ta còn có thêm dữ liệu cho bảng từ giá trị của bảng khác:

– Cú pháp:

```
INSERT [INTO]<TableName> (Column1, Column2, ..., Columnn)
SELECT Select_list FROM <Tables>
```

**Ví dụ:**

```
INSERT INTO NHANVIEN_tam (TenNV, Tuoi)
SELECT TenNV, Tuoi FROM NHANVIEN WHERE Tuoi > 20
```

#### 2. Xóa hàng – DELETE

– Cú pháp:

```
DELETE FROM <Table_name> WHERE <Conditions>
```

Lệnh này gồm 1 mệnh đề DELETE FROM để chỉ ra tên gọi của bảng được xét, và một mệnh đề WHERE để chỉ ra các dòng cần phải xóa. Như vậy, ta có thể cùng lúc xóa được nhiều dòng nếu dòng đó thỏa mãn điều kiện. Muốn xóa mọi dòng của một bảng thì không cần đưa vào mệnh đề WHERE.

**Ví dụ:**

- Xóa một bản ghi (dòng) có MaDG='DHTL01' trong bảng DOCGIA.

```
DELETE FROM DOCGIA WHERE MaDG='DHTL01'
```

- Xóa những độc giả có địa chỉ là: 41NC có trong bảng DOCGIA.

```
DELETE FROM DOCGIA WHERE Diachi='41NC'
```

### 3. Sửa đổi giá trị của một hàng – UPDATE

- **Cú pháp:**

```
UPDATE <Table_name>  
  
SET (Column_name= <new value>)  
  
WHERE <Condition>
```

**Ví dụ:**

```
UPDATE DOCGIA  
  
SET (Diachi= 'Khoa Cong trinh')  
  
WHERE MaDG= 'TD001'
```

## IV. CÁC LỆNH KIỂM SOÁT DỮ LIỆU (DCL)

Ngôn ngữ kiểm soát dữ liệu được sử dụng trong việc cấp phát hay huỷ bỏ quyền của người sử dụng.

### 1. Lệnh cấp quyền – GRANT

Câu lệnh này dùng để cấp phát quyền cho người sử dụng trên đối tượng Cơ sở dữ liệu hoặc quyền thực thi các câu lệnh SQL SERVER. Cú pháp có 2 dạng như sau:

Dạng 1: Cấp quyền đối với câu lệnh SQL

GRANT ALL | statement[,...,statementN ]

TO account [,...,accountN]

Dạng 2: Cấp quyền đối với các đối tượng trong cơ sở dữ liệu

GRANT ALL | permission [,...,permissionN]

ON table\_name |view\_name [(column1 [,...,columnN])]

|ON stored\_procedure

TO account [,...,accountN]

Trong đó:

- ALL: là từ khoá được sử dụng khi muốn cấp phát tất cả các quyền cho người sử dụng.
- Account: là tên tài khoản đăng nhập hệ thống
- Permission: là quyền cấp phát cho người sử dụng trên đối tượng cơ sở dữ liệu:
  - + Các quyền có thể cấp trên một bảng: Select, Insert, Delete, Update.
  - + Các quyền có thể cấp trên cột của bảng: Select, Update
  - + Quyền có thể cấp phát với các thủ tục: EXCUTE(thực thi)
- Statement: Là câu lệnh được cấp phát cho người sử dụng Các câu lệnh có

thể cấp phát là:

- + CREATE DATABASE
- + CREATE TABLE
- + CREATE VIEW
- + CREATE PROCEDURE
- + CREATE RULE
- + CREATE DEFAULT
- + BACKUP DATABASE
- + BACKUP LOG

**Ví dụ 1:** cấp 3 quyền SELECT, UPDATE, INSERT trên các thuộc tính (TenNV, DiaChi, Tuoi) của bảng NHANVIEN cho 2 người sử dụng phnhung, htvan.

```
GRANT SELECT, UPDATE, INSERT  
ON NHANVIEN (TenNV, DiaChi, Tuoi)  
TO phnhung, htvan
```

**Ví dụ 2:** cấp quyền tạo bảng, tạo View và tạo thủ tục cho người dùng phnhung.

```
GRANT CREATE TABLE, CREATE VIEW, CREATE PROCEDURE  
TO phnhung
```

## 2. Lệnh thu hồi quyền – REVOKE

Lệnh REVOKE được dùng để huỷ bỏ quyền đã được cấp phát cho người sử dụng. Câu lệnh này cũng có 2 dạng tương tự như câu lệnh GRANT.

Dạng 1: Huỷ quyền thực hiện câu lệnh:

```
REVOKE ALL | statement [,...,statementN]  
FROM account [,...,accountN]
```

Dạng 2: Huỷ quyền thực hiện các đối tượng:

```
REVOKE ALL | permission [,...,permissionN]}  
ON table_name | view_name [(column [,...,columnN])] | stored_procedure  
FROM account [,...,accountN ]
```

## V. CÁC LỆNH TRUY VẤN DỮ LIỆU (DQL)

Câu lệnh SELECT dùng để lọc, tổng hợp dữ liệu theo các tiêu chí cụ thể từ CSDL. Kết quả của câu lệnh SELECT là một bảng. Câu lệnh SELECT có thể thực hiện trên 1 bảng hoặc trên nhiều bảng.

### 1. Cú pháp tổng quát

Câu lệnh SELECT có nhiều mệnh đề, mỗi mệnh đề đảm bảo một chức năng:

- SELECT      Danh sách các cột được trả về trong kết quả truy vấn.
- FROM        Danh sách các bảng hay view chứa dữ liệu được chọn

- WHERE            Cung cấp điều kiện lọc
- ORDER BY      xác định thứ tự sắp xếp các dòng dữ liệu trả về
- GROUP BY      Gom nhóm dữ liệu

```
SELECT [DISTINCT] <biểu thức 1> [AS <tên 1>] [...] | *  
FROM <bảng 1> [<bí danh 1>] [...]  
[INTO <dbf đích>]  
[WHERE <điều kiện nối > [AND | OR <điều kiện lọc>]]  
[GROUP BY <cột nhóm 1> [...]  
          [HAVING <điều kiện nhóm>]]  
[ORDER BY <biểu thức sắp xếp 1> [ASC | DESC] [...]]  
[UNION | INTERSECT | MINUS <câu truy vấn khác>]
```

## **2. Các thành phần cơ bản của câu lệnh**

- Biểu thức sau mệnh đề **SELECT** có thể bao gồm:
  - + Danh sách các cột, kể cả các biểu thức chứa các cột, của các bảng hoặc khung nhìn khai báo sau FROM. Các biểu thức ngăn cách nhau bằng dấu phẩy (,) và có thể đổi tên bằng tùy chọn AS <tên >.
  - + \* là ký tự đại diện tất cả các thuộc tính của bảng sau FROM
  - + Các hàm tính toán : COUNT, SUM, AVG, MIN, MAX
  - + Từ khoá DISTINCT dùng liệt kê những giá trị duy nhất tránh trùng lặp.
- Biểu thức sau mệnh đề **FROM** gồm một hoặc danh sách các bảng quan hệ (có thể đặt bí danh).
- **INTO** <dbf đích>: Lưu bảng kết quả vào đĩa.
- Biểu thức sau mệnh đề **WHERE** bao gồm :
  - + Các thuộc tính của các bảng quan hệ sau FROM
  - + Các toán tử số học : +, -, \*, /

- + Các toán tử so sánh, có thể so sánh với ANY (giá trị nào đó trong 1 tập hợp) hoặc ALL (tất cả giá trị trong 1 tập hợp)
- + Các toán tử Boolean : AND, OR, NOT
- + Các toán tử tập hợp : UNION (hợp), INTERSECT (giao), MINUS (hiệu).
- + Các quan hệ bao hàm : IN ( $\in$ ), NOT IN ( $\notin$ ), CONTAINS (chứa), DOES NOT CONTAIN (không chứa)
- + Điều kiện tồn tại : EXISTS (tồn tại), NOT EXIST (không tồn tại)
- + Câu văn tin con kiểu SELECT ... FROM ... WHERE ...

### 3. Các loại biểu thức

#### 3.1. Truy vấn đơn giản – Mệnh đề *SELECT*

##### – Cú pháp:

```
SELECT [DISTINCT] <biểu thức I> [AS <tên I>] [,...] | *  
FROM <bảng >
```

##### – Công dụng:

Xác định nội dung kết quả gồm các cột liệt kê sau SELECT hoặc tất cả các cột <\*> lấy từ <Bảng>.

- + Từ khoá DISTINCT để các bản ghi giống nhau chỉ lấy 1 bản ghi.
- + Từ khoá AS dùng đặt bí danh (là tên mới của bảng kết quả)

#### Ví dụ 3.1

- Cho lược đồ cơ sở dữ liệu khách hàng

KHACH(TenKH, DiaChi, SoDu)

HOPDONG(SoHD, TenKH, Hang, SoLuong)

CUNGUNG(Ten, DiaChi, Hang, DonGia)

- Trả về các bản ghi gồm tất cả các cột của bảng CUNGUNG.

```
SELECT *
```

```
FROM CUNGUNG
```

- Trả về các bản ghi gồm 2 cột Hang, DonGia của bảng CUNGUNG.

```
SELECT Hang, DonGia  
  
FROM CUNGUNG
```

hoặc

```
SELECT DISTINCT Hang, DonGia  
  
FROM CUNGUNG
```

- Trả về các bản ghi gồm 1 cột Hang với giá trị duy nhất của bảng CUNGUNG.

```
SELECT DISTINCT Hang  
  
FROM CUNGUNG
```

- Có thể cho hiển thị giá trị hằng, chẳng hạn

```
SELECT "Mặt hàng", Hang, "có đơn giá là", DonGia  
  
FROM CUNGUNG
```

### **3.2. Truy vấn có điều kiện – Mệnh đề *WHERE***

- **Cú pháp:**

<pre>SELECT [DISTINCT] &lt;biểu thức 1&gt; [AS &lt;tên 1&gt;] [...]  *  FROM &lt;bảng&gt;  WHERE &lt;điều kiện lọc&gt;</pre>
--

- **Công dụng:**

Trả về các bản ghi gồm các cột liệt kê sau SELECT hoặc tất cả các cột <\*> lấy từ <Bảng> gồm các bộ thỏa mãn <điều kiện lọc>.

- **Ghi chú:** Sau WHERE, trong điều kiện lọc có thể dùng các mệnh đề :

+ <biểu thức> BETWEEN <giá trị 1> AND <giá trị 2> : nghĩa là  
    <giá trị 1> ≤ <biểu thức> ≤ <giá trị 2>

+ <biểu thức> NOT BETWEEN <giá trị 1> AND <giá trị 2>: nghĩa là  
    <biểu thức> <<giá trị 1> hoặc <biểu thức> > <giá trị 2>

+ <biểu thức> IN (<giá trị 1>, <giá trị 2>, ...) : nghĩa là <biểu thức> bằng một trong các giá trị <giá trị 1>, <giá trị 2>, ...

+ <biểu thức> NOT IN (<giá trị 1>, <giá trị 2>, ...) : nghĩa là <biểu thức> khác các giá trị <giá trị 1>, <giá trị 2>, ...

+ <biểu thức ký tự> [NOT] LIKE <chuỗi ký tự> : nghĩa là <biểu thức ký tự> [không] giống <chuỗi ký tự>.

Trong <chuỗi ký tự> có thể dùng dấu:

- % (dấu phần trăm) đại diện cho chuỗi ký tự bất kỳ
- \_ (dấu nối) đại diện cho một ký tự bất kỳ

+ <trường> IS [NOT] NULL: nghĩa là giá trị <trường> là [khác] NULL. NULL là giá trị rỗng.

### Ví dụ 3.2

– Trả về các bản ghi gồm các cột Ten, DiaChi có giá trị duy nhất từ bảng CUNGUNG của các nhà cung ứng hoặc Đường hoặc Sữa. Có thể dùng một trong 2 lệnh sau.

<pre>SELECT DISTINCT Ten, DiaChi FROM CUNGUNG WHERE (Hang = 'Đường') OR (Hang = 'Sữa')</pre>	<pre>SELECT DISTINCT Ten, DiaChi FROM CUNGUNG WHERE Hang IN ('Đường', 'Sữa')</pre>
--	--

– Trả về các bản ghi gồm các Ten, DiaChi của các khách hàng có SoDu không âm từ bảng KHACH

```
SELECT Ten, DiaChi
FROM KHACH
WHERE SoDu >= 0
```

– Trả về các bản ghi gồm các hợp đồng có SoLuong từ 100 đến 200 từ bảng HOPDONG

```
SELECT *
FROM HOPDONG
```



```
WHERE SoLuong BETWEEN 100 AND 200
```

- Trả về các bản ghi gồm các nhà cung ứng có tên bắt đầu bằng ‘Cty’ từ bảng CUNGUNG

```
SELECT *
```

```
FROM CUNGUNG
```

```
WHERE Ten LIKE 'Cty %'
```

- Trả về các bản ghi gồm các nhà cung ứng có mặt hàng ‘Bia’ (tên bắt đầu bằng ‘Bia’) từ bảng CUNGUNG.

```
SELECT *
```

```
FROM CUNGUNG
```

```
WHERE Hang LIKE 'Bia%'
```

- Chọn những khách hàng chưa có địa chỉ

```
SELECT *
```

```
FROM KHACH
```

```
WHERE DiaChi IS NULL
```

- Chọn những khách hàng có số dư xác định

```
SELECT *
```

```
FROM KHACH
```

```
WHERE SoDu IS NOT NULL
```

### **3.3. Truy vấn có sắp xếp – Mệnh đề *ORDER BY***

- **Cú pháp:**

```
SELECT [DISTINCT] <biểu thức l> [AS <tên l>] [,...] | *
```

```
FROM <bảng>
```

```
ORDER BY <biểu thức sắp xếp l> [ASC | DESC] [,...]
```

- **Công dụng**

Trả về các bản ghi gồm các cột liệt kê sau SELECT hoặc tất cả các cột <\*> lấy từ <Bảng> gồm các bộ được sắp xếp theo các biểu thức sau ORDER BY.

Ý nghĩa các tham số:

+ ASC : tăng dần (mặc định)

+ DESC : giảm dần.

**Ví dụ 3.3:**

- Cho biết danh sách khách hàng sắp xếp theo TEN tăng dần :

```
SELECT *  
  
FROM KHACH  
  
ORDER BY Ten
```

- Cho biết danh sách các nhà cung ứng ‘Sữa’ sắp xếp theo thứ tự DonGia giảm dần:

```
SELECT *  
  
FROM CUNGUNG  
  
WHERE Hang = 'Sữa'  
  
ORDER BY DonGia DESC
```

- Cho biết danh sách các nhà cung ứng sắp xếp theo thứ tự Ten và Hang:

```
SELECT *  
  
FROM CUNGUNG  
  
ORDER BY Ten, Hang
```

**3.4. Câu lệnh truy vấn lồng nhau**

- **Cú pháp:**

```
SELECT [DISTINCT] <biểu thức 1> [AS <tên 1>] [...] / *  
  
FROM <bảng 1> [<bí danh 1>] [...]  
  
[WHERE <điều kiện nối , lọc có chứa câu truy vấn con >]  
  
[ORDER BY <biểu thức sắp xếp 1> [ASC / DESC] [...]]
```

- **Công dụng:**

+ Cho phép tạo ra các điều kiện chứa quan hệ tập hợp và số liệu tổng hợp.

+ Trong quan hệ tập hợp, câu truy vấn con đặt trong ngoặc đơn.

**Ví dụ 3.4:**

– Sử dụng các toán tử so sánh : <, <=, =, >, >=, <>

+ Liệt kê các khách hàng có số dư lớn nhất:

```
SELECT Ten
FROM KHACH
WHERE Sodu =
      (SELECT MAX(Sodu)
       FROM KHACH)
```

+ Liệt kê hãng cung ứng gạo có DonGia <= DonGia trung bình :

```
SELECT Ten
FROM CUNGUNG a
WHERE (a.Hang = 'Gạo') AND DonGia <=
      (SELECT AVG(b.DonGia)
       FROM CUNGUNG b
       WHERE b.Hang = 'Gạo')
```

– Sử dụng các toán tử so sánh : <, <=, =, >, >=, <> với ANY / ALL

+ Liệt kê nhân viên từ bảng NHANVIEN có NgaySinh trùng với người khác

```
SELECT a.Manv, a.HoLot, a.Ten
FROM NHANVIEN a
WHERE a.NgaySinh = ANY
      (SELECT b.NgaySinh
       FROM NHANVIEN b
       WHERE b.Manv <> a.Manv)
```

+ Liệt kê nhân viên từ bảng NHANVIEN có NgaySinh không trùng với người khác

```
SELECT a.Manv, a.HoLot, a.Ten
```

```
FROM      NHANVIEN a

WHERE      a.NgaySinh <> ALL

           (SELECT   b.NgaySinh

FROM      NHANVIEN b

WHERE      b.Manv <> a.Manv)
```

– Sử dụng toán tử **IN (NOT IN)** biểu diễn quan hệ thuộc  $\in$

+ Liệt kê Ten, Hang và DonGia của các hãng cung ứng có ít nhất 1 mặt hàng trong HOPDONG:

```
SELECT DISTINCT Ten, Hang, DonGia

FROM CUNGUNG

WHERE Hang IN

      (SELECT Hang

FROM HOPDONG )
```

+ Liệt kê Ten, Hang và DonGia của các hãng cung ứng có ít nhất 1 mặt hàng mà khách hàng ‘KS. Bạch Đằng’ đặt hàng:

```
SELECT DISTINCT Ten, Hang, DonGia

FROM CUNGUNG

WHERE Hang IN

      (SELECT Hang

FROM HOPDONG

WHERE TenKh = ‘KS. Bạch Đằng’ )
```

+ Liệt kê Ten, DiaChi các hãng cung ứng không bán Gạo :

```
SELECT DISTINCT Ten, DiaChi

FROM CUNGUNG

WHERE Ten NOT IN

      SELECT Ten
```

```
FROM CUNGUNG
```

```
WHERE Hang = 'Gạo')
```

+ Liệt kê hãng có ít nhất 1 mặt hàng giống Cty Lương thực :

```
SELECT DISTINCT a.Ten, a.DiaChi
```

```
FROM CUNGUNG a
```

```
WHERE a.Hang IN
```

```
SELECT b.Hang
```

```
FROM CUNGUNG b
```

```
WHERE b.Ten = 'Cty lương thực')
```

+ Liệt kê hợp đồng có TenKh và Hang giống hợp đồng số 3 :

```
SELECT *
```

```
FROM HOPDONG a
```

```
WHERE (a.TenKh, a.Hang) IN
```

```
SELECT b.TenKh, b.Hang
```

```
FROM HOPDONG
```

```
WHERE b.SoHd = 3)
```

+ Liệt kê Hãng không có mặt hàng trong HOPDONG :

```
SELECT DISTINCT Ten
```

```
FROM CUNGUNG
```

```
MINUS
```

```
SELECT DISTINCT Ten
```

```
FROM CUNGUNG
```

```
WHERE Hang IN
```

```
SELECT Hang
```

```
FROM HOPDONG )
```

+ Liệt kê Hãng có mặt hàng không có trong HOPDONG :

```
SELECT DISTINCT Ten
```

```
FROM CUNGUNG
```

```
WHERE Hang NOT IN
```

```
    SELECT Hang
```

```
    FROM HOPDONG )
```

+ Liệt kê Hãng, mà tất cả mặt hàng của nó đều có trong HOPDONG :

```
SELECT DISTINCT Ten
```

```
FROM CUNGUNG
```

```
GROUP BY Ten
```

```
    HAVING SET(Hang) IN
```

```
        SELECT Hang
```

```
        FROM HOPDONG )
```

– Sử dụng **[NOT] EXISTS <bảng> biểu diễn sự tồn tại**

+ Liệt kê Ten, DiaChi khách hàng đã đặt hàng :

```
SELECT      Ten, DiaChi
```

```
FROM KHACH
```

```
WHERE      EXISTS
```

```
    SELECT      Hang
```

```
    FROM      HOPDONG
```

```
    WHERE      TenKh = Ten)
```

+ Liệt kê Ten, DiaChi khách hàng chưa có hợp đồng:

```
SELECT      Ten, DiaChi
```

```
FROM KHACH
```

```
WHERE      NOT EXISTS
```

```
    SELECT      Hang
```

```
    FROM      HOPDONG
```

WHERE TenKh = Ten)

### 3.5. Các hàm tính toán

Các hàm tính toán có thể sử dụng sau SELECT hoặc trong điều kiện nhóm, đối số của nó là tên của thuộc tính mà hàm phải tính toán .

Các hàm tính toán thường dùng:

- COUNT(...) : Đếm số bản ghi
- SUM(...) : Tính tổng
- AVG(...) : Tính giá trị trung bình
- MAX(...) : Tìm giá trị lớn nhất.
- MIN(...) : Tìm trị nhỏ nhất

Ví dụ:

- Tính tổng số hợp đồng

```
SELECT COUNT (*)
```

```
FROM HOPDONG
```

### 3.6. Gom nhóm dữ liệu – mệnh đề GROUP BY

- **Cú pháp:**

```
SELECT [DISTINCT] <biểu thức 1> [AS <tên 1>] [...] | *  
FROM <bảng 1> [<bí danh 1>] [...]  
[WHERE <điều kiện nói > [AND / OR <điều kiện lọc>]]  
GROUP BY <thuộc tính 1> [...]  
[HAVING <điều kiện nhóm>]
```

- **Công dụng:**

- + Nhóm các bản ghi (thỏa mãn <điều kiện nói> hoặc <điều kiện lọc>, nếu có) theo danh sách thuộc tính sau GROUP BY , sau đó kết xuất kết quả tổng hợp (thỏa <điều kiện nhóm>).
- + Các thuộc tính sau GROUP BY phải có trong danh sách thuộc tính sau SELECT.

– **Ghi chú:**

- + Hàm tập hợp SET(...) (tập hợp các phần tử) chỉ sử dụng trong điều kiện nhóm.
- + Các hàm tính toán có thể sử dụng sau SELECT hoặc trong điều kiện nhóm. Xem lại mục 3.5

**Ví dụ 3.5**

- Liệt kê các hãng cùng tổng số mặt hàng mà nó cung ứng :

```
SELECT Ten, COUNT(Hang) AS SoHang  
  
FROM CUNGUNG  
  
GROUP BY Ten
```

- Liệt kê các mặt hàng cùng tổng SoLuong trong HOPDONG

```
SELECT Hang, SUM(SoLuong) AS TongCong  
  
FROM HOPDONG  
  
GROUP BY Hang
```

- Liệt kê các mặt hàng cùng giá bình quân trong CUNGUNG :

```
SELECT Hang, AVG(DonGia) AS DonGiaBQ  
  
FROM CUNGUNG  
  
GROUP BY Hang
```

- Liệt kê các mặt hàng cùng tổng số hãng cung ứng mặt hàng đó :

```
SELECT Hang, COUNT(Ten) AS SoCty  
  
FROM CUNGUNG  
  
GROUP BY Hang
```

- Liệt kê các mặt hàng có nhiều hãng cung ứng mặt hàng đó :

```
SELECT Hang, COUNT(Ten) AS SoCty  
  
FROM CUNGUNG  
  
GROUP BY Hang
```

```
HAVING COUNT(Ten) > 1
```



- Liệt kê các hãng cung ứng cả Đường và Sữa :

```
SELECT DISTINCT Ten
```

```
FROM CUNGUNG
```

```
GROUP BY Ten
```

```
HAVING SET (Hang) CONTAINS ( 'Đường' , 'Sữa' )
```

## **VI. BÀI TẬP**

**Bài tập 3.1.** Cho Lược đồ cơ sở dữ liệu quản lý nhân viên của một công ty như sau:

Nhanvien(MANV,HOTEN, NU,NGAYSINH,LUONG,MAPB, MACV)

Mỗi nhân viên có một mã nhân viên (MANV) duy nhất, mỗi mã nhân viên xác định họ và tên nhân viên (HOTEN), giới tính (NU), lương (LUONG), mã phòng ban (MAPB), mã chức vụ (MACV).

Phongban(MAPB,TENPB,TRUSO,MANVPHUTRACH,KINHPHI,DOANHTHU)

Mỗi phòng ban có tên gọi phòng ban(TENPB), địa điểm đặt trụ sở (TRUSO), mã nhân viên phụ trách(MANVPHUTRACH), kinh phí hoạt động (KINHPHI), và doanhthu(DOANHTHU)

Chucvu(MACV,TENCV,LUONGTHAPNHAT,LUONGCAONHAT)

Mỗi chức vụ có tên gọi chức vụ (TENCV), mức lương tối thiểu(LUONGTHAPNHAT), mức lương tối đa (LUONGCAONHAT).

Hãy biểu diễn các câu hỏi sau bằng SQL:

1. Lập danh sách gồm các thông tin về các phòng ban trong công ty như: mã số phòng ban, tên phòng ban, địa điểm trụ sở, mã số người phụ trách, kinh phí hoạt động, doanh thu.
2. Lập danh sách những nhân viên sinh nhật trong tháng 10
3. Lập danh sách gồm các thông tin mã số nhân viên, họ và tên và lương cả năm của các nhân viên (giả sử rằng lương cả năm =12\*lương)
4. Lập những phòng ban có kinh phí hoạt động cao nhất.
5. Lập danh sách nhân viên của phòng ban có mã số phòng ban là 40.
6. Lập danh sách nhân viên của phòng có mã số phòng ban 10,30,50.

7. Lập danh sách các nhân viên có lương tháng từ 2.500.000 đến 4.000.000
8. Tìm những nhân viên có tuổi cao nhất thuộc phòng ban có MAPB là 10
9. Lập danh sách các nhân viên của phòng 10,30,50. kết quả in ra theo thứ tự tăng dần của mã phòng nếu trùng mã phòng thì sắp xếp giảm dần theo mức lương.
10. Lập danh sách các nhân viên phòng 10,30,50, chỉ in ra những người là lãnh đạo của mỗi phòng ban này.
11. Lập danh sách gồm mã phòng mà người có mức lương cao nhất của phòng lớn hơn hoặc bằng 4.000.000
12. Lập mã phòng ban, tên phòng ban, họ và tên của lãnh đạo phòng tương ứng.
13. Lập danh sách những người làm việc cùng phòng với ông Nguyen Van Thanh
14. Lập biết mã số nhân viên, họ và tên, mức lương của người lãnh đạo ông Nguyen Van Thanh.
15. Lập danh sách nhân viên có mức lương lớn hơn hay bằng mức lương cao nhất của phòng ông Nguyen Van Thanh.
16. Cho biết mã số nhân viên, họ và tên , tổng số nhân viên, mức lương cao nhất, mức lương thấp nhất, mức lương trung bình của từng phòng ban.

**Bài tập 3.2.** Cho một lược đồ cơ sở dữ liệu của 1 thư viện như sau:

TheDocGia(MATHEDG, HOTENDG, DIACHI)

Sach(MASACH, TENSACH, NAMXB, NHAXB, TENTACGIA)

Phieumuon(MAPHIEU, NGAYMUON, MASACH, MATHEDG, NGAYTRA, TRA)

Dùng SQL trả lời các câu hỏi sau đây:

1. Liệt kê các phiếu mượn trong ngày 20/11/2009
2. Liệt kê họ tên độc giả mượn sách trong tháng 3 năm 2009
3. Liệt kê các tên sách thuộc nhà xuất bản “Giáo Dục” hiện có trong thư viện
4. Liệt kê tên các độc giả hiện đang còn giữ sách có tên “Đồ Hoạ Máy Tính” chưa trả cho thư viện
5. Liệt kê tên và địa chỉ liên lạc của các độc giả hiện đang còn nợ sách của thư viện

6. Liệt kê tên các độc giả và tên sách mà các độc giả này đang giữ đã đến kỳ hạn trả cho thư viện. Biết rằng qui định là sau khi mượn 10 ngày thì phải trả sách lại cho thư viện
7. Thống kê xem mỗi thẻ độc giả đã mượn được bao nhiêu cuốn sách trong tháng 11/2009
8. Liệt kê các cuốn sách chưa được ai mượn lần nào
9. Liệt kê tất cả các cuốn sách được mượn nhiều nhất trong tháng 11/2009
10. Liệt kê các độc giả quen thuộc của thư viện. Biết rằng các độc giả quen thuộc là các độc giả đến mượn sách ít nhất là 3 lần

**Bài tập 3.3.** Dựa vào lược đồ CSDL sau:

Congtrinh(MACT, TENCT, DIADIEM, NGAYCAPGP, NGAYKC, NGAYT)

Nhanvien(MANV, HOTEN, NGAYSINH, PHAI, DIACHI, MAPB)

Phongban(MAPB, TENPB)

Phancong(MACT, MANV, SLNGAYCONG)

Hãy thực hiện các câu hỏi sau bằng SQL

1. Danh sách những nhân viên có tham gia vào công trình có mã công trình(MACT) là X. Yêu cầu các thông tin: MANV, HOTEN, SLNGAYCONG, trong đó MANV được sắp tăng dần
2. Đếm số lượng ngày công của mỗi công trình. Yêu cầu các thông tin: MACT, TENCT, TONGNGAYCONG (là thuộc tính tự đặt)
3. Danh sách những nhân viên có sinh nhật trong tháng 08. Yêu cầu các thông tin: MANV, TENNV, NGAYSINH, DIACHI, TENPB, sắp xếp quan hệ kết quả theo thứ tự tuổi giảm dần.
4. Đếm số lượng nhân viên của mỗi phòng ban. Yêu cầu các thông tin: MAPB, TENPB, SOLUONG. (SOLUONG là thuộc tính tự đặt.)

**Bài tập 3.4.** Dựa vào lược đồ cơ sở dữ liệu Giaovien(MAGV, HOTEN, MAKHOA)

Monhoc(MAMH, TENMH)

Phonghoc(PHONG, CHUCNANG)

Khoa(MAKHOA, TENKHOA)

Lop(MALOP, TENLOP, MAKHOA)

Lichday(MAGV, MAMH, PHONG, MALOP, NGAYDAY, TUTIET, ĐENTIET, BAIDAY, LYTHUYET, GHICHU)

Hãy thực hiện các câu hỏi sau bằng SQL

1. Xem lịch báo giảng tuần từ ngày 08/09/2003 đến ngày 14/09/2003 của giáo viên có MAGV (mã giáo viên) là TH3A040. Yêu cầu: MAGV, HOTEN, TENLOP, TENMH, PHONG, NGAYDAY, TUTIET, ĐENTIET, BAIDAY, GHICHU)
2. Xem lịch báo giảng ngày 08/09/2003 của các giáo viên có mã khoa là CNTT. Yêu cầu: MAGV, HOTEN, TENLOP, TENMH, PHONG, NGAYDAY, TUTIET, ĐENTIET, BAIDAY, GHICHU)
3. Cho biết số lượng giáo viên (SOLUONGGV) của mỗi khoa, kết quả cần sắp xếp tăng dần theo cột tên khoa. yêu cầu: TENKHOA, SOLUONGGV (SOLUONGGV là thuộc tính tự đặt).

**Bài tập 3.5.** Hàng năm, Trường X tổ chức kỳ thi giỏi nghề cho các sinh viên của trường, mỗi thí sinh sẽ thi hai môn (chẳng hạn các thí sinh thi giỏi nghề công nghệ thông tin thi hai môn là Visual Basic và Cơ Sở Dữ Liệu).

Giả sử lược đồ cơ sở dữ liệu của bài toán quản lý các kỳ thi trên được cho như sau:

Thisinh(MASV, HOTEN, NGAYSINH, MALOP)

Lop(MALOP, TENLOP, MAKHOA)

Khoa(MAKHOA, TENKHOA, ĐIENTHOAI)

Monthi(MAMT, TENMONTHI)

Ketqua(MASV, MAMT, ĐIEMTHI)

(Phần giải thích các thuộc tính: HOTEN (họ tên thí sinh), NGAYSINH (ngày sinh), MALOP (mã lớp), MASV (mã sinh viên), TENLOP (tên lớp), MAKHOA (mã khoa), TENKHOA (tên khoa), ĐIENTHOAI (số điện thoại khoa), MAMT (mã môn thi), TENMONTHI (tên môn thi), ĐIEMTHI (điểm thi)).

Dựa vào lược đồ cơ sở dữ liệu trên, hãy thực hiện các yêu cầu sau bằng ngôn ngữ SQL:

1. Hãy cho biết số lượng thí sinh của mỗi khoa đăng ký thi giỏi nghề, cần sắp xếp kết quả theo chiều tăng dần của cột TENKHOA.
2. Lập danh sách những thí sinh đạt danh hiệu giỏi nghề (Thí sinh đạt danh hiệu giỏi nghề nếu thí sinh không có môn thi nào điểm dưới 8).
3. Lập danh sách những thí sinh nhỏ tuổi nhất có mã khoa là “CNTT” dự thi giỏi nghề.

## CHƯƠNG 4. RÀNG BUỘC TOÀN VỆ

Ràng buộc toàn vẹn là các quy định, điều kiện từ ứng dụng thực tế. Các điều kiện này là bất biến. Vì vậy, luôn phải đảm bảo cơ sở dữ liệu thỏa ràng buộc toàn vẹn sau mỗi thao tác làm thay đổi tình trạng của cơ sở dữ liệu.

### I. RÀNG BUỘC TOÀN VỆ

#### 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 thỏa 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,...

Ràng buộc toàn vẹn là một điều kiện bất biến không được vi phạm trong một CSDL.

#### 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.

##### 2.1. Điều kiện

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.

Ví dụ sau đây chỉ ra một số ràng buộc toàn vẹn trên lược đồ CSDL quản lý sinh viên:

- 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.

## **2.2. Bối cảnh**

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

## **2.3. Bảng tầm ảnh hưởng**

Trong quá trình phân tích thiết kế một CSDL, người phân tích cần lập bảng tầm ảnh hưởng cho một ràng buộc toàn vẹn nhằm xác định thời điểm cần phải tiến hành kiểm tra khi tiến hành cập nhật dữ liệu.

Thời điểm cần phải kiểm tra ràng buộc toàn vẹn chính là thời điểm cập nhật dữ liệu.

Một bảng tầm ảnh hưởng của một ràng buộc toàn vẹn có dạng sau:

Tên RBTV	Thêm(T)	Sửa(S)	Xoá(X)
$r_1$	+		
$r_2$		-	
$r_3$			-(*)
$r_n$			

Bảng này chứa toàn các ký hiệu + , – hoặc -(\*) .

- Dấu + tại (dòng  $r_1$ , cột Thêm) thì có nghĩa là khi thêm một bộ vào quan hệ  $r_1$  thì RBTV bị vi phạm.
- Dấu - tại ô (dòng  $r_2$ , cột Sửa) thì có nghĩa là khi sửa một bộ trên quan hệ  $r_2$  thì RBTV không bị vi phạm.

**Quy ước:**

- Không được sửa thuộc tính khoá.
- Nếu không bị vi phạm do không được phép sửa đổi thì ký hiệu là -(\*)

#### **2.4. Hành động cần phải có khi phát hiện có RBTV bị vi phạm:**

Khi một ràng buộc toàn vẹn bị vi phạm, cần có những hành động thích hợp. Thông thường có 2 giải pháp:

- Thứ nhất: Đưa ra thông báo và yêu cầu sửa chữa dữ liệu của các thuộc tính cho phù hợp với quy tắc đảm bảo tính nhất quán dữ liệu. Thông báo phải đầy đủ và phải thân thiện với người sử dụng. Giải pháp này là phù hợp cho việc xử lý thời gian thực.
- Thứ hai: Từ chối thao tác cập nhật. Giải pháp này là phù hợp đối với việc xử lý theo lô. Việc từ chối cũng phải được lưu lại bằng những thông báo đầy đủ, rõ ràng vì sao thao tác bị từ chối và cần phải sửa lại những dữ liệu nào?

Khóa nội, khóa ngoại, giá trị NOT NULL là những ràng buộc toàn vẹn miền giá trị của các thuộc tính. Những ràng buộc toàn vẹn này là những ràng buộc toàn vẹn đơn giản trong CSDL.

Các hệ quản trị cơ sở dữ liệu thường có các cơ chế tự động kiểm tra các ràng buộc toàn vẹn về miền giá trị của khóa nội, khóa ngoại, giá trị NOT NULL.

Việc kiểm tra ràng buộc toàn vẹn có thể tiến hành vào những thời điểm sau đây:

- Thứ nhất: Kiểm tra ngay sau khi thực hiện một thao tác cập nhật CSDL. Thao tác cập nhật chỉ được xem là hợp lệ nếu như nó không vi phạm bất cứ một ràng buộc toàn vẹn nào, nghĩa là nó không làm mất tính toàn vẹn của CSDL. Nếu vi phạm ràng buộc toàn vẹn, thao tác cập nhật bị coi là không hợp lệ và sẽ bị hệ thống huỷ bỏ (hoặc có một xử lý thích hợp nào đó).
- Thứ hai: Kiểm tra định kỳ hay đột xuất, nghĩa là việc kiểm tra ràng buộc toàn vẹn được tiến hành độc lập với thao tác cập nhật dữ liệu. Đối với những trường hợp vi phạm ràng buộc toàn vẹn, hệ thống có những xử lý ngầm định hoặc yêu cầu người sử dụng xử lý những sai sót một cách tường minh.

## **II. Phân loại các ràng buộc toàn vẹn**

Phân loại các ràng buộc toàn vẹn dựa trên bối cảnh của một quan hệ hay bối cảnh nhiều quan hệ,



**Ví dụ:** 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:

**Q<sub>1</sub>: Khách (MAKH, TENKH, DIACHIKH, DIENTHOAI)**

Tân từ<sup>1</sup>:

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).

**Q<sub>2</sub>: Hàng (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).

**Q<sub>3</sub>: 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). Mỗi khách hàng trong một ngày có thể có nhiều lần đặt hàng.

**Q<sub>4</sub>: 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 SODH 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 những các mặt hàng trong hoá đơn (TRIGIAHD) và một ngày xuất kho giao hàng cho khách (NGAYXUAT)

**Q<sub>5</sub>: Chitiethd (SOHD, MAHANG, GIABAN, SLBAN)**

---

<sup>1</sup> Khái niệm Tân từ xem lại trang 21, mục Lược đồ quan hệ.

*Ràng buộc toàn vẹn có bối cảnh là một quan hệ*

Tân từ:

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

**Q<sub>6</sub>: 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.

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

### 1.1. Ràng buộc toàn vẹn liên bộ

*Ràng buộc toàn vẹn về khoá chính:*

Đây là một trường hợp đặc 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:**

Với r là một quan hệ trên lược đồ quan hệ Khách ta có RBTV sau:

$R_1: \forall t_1, t_2 \in r$

$t_1.MAKH \neq t_2.MAKH$

Cuối  $\forall$

$R_1$	Thêm	Sửa	Xoá
Khách	+	-	-

*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 các RBTV trong quan hệ sau đây.

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

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

Ràng buộc toàn vẹn có bối cảnh là nhiều quan hệ

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

$R_3: \forall t \in r$

$t.TRIGIAHD > 0$

Cuối  $\forall$

$R_3$	Thêm	Sửa	Xoá
Hoadon	+	+	-

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

$R_4: \forall t \in r$

$t.NGAYLAP \leq t.NGAYXUAT$

Cuối  $\forall$

$R_4$	Thêm	Sửa	Xoá
Hoadon	+	+	-

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

### 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:

R <sub>2</sub> : dathang[MAKH] $\subseteq$ khach[MAKH]			
R <sub>2</sub>	Thêm	Sửa	Xoá
dathang	+	+	-
Khach	-	-	+

## 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:** Với r, s lần lượt là quan hệ của Dathang và Hoadon. Ta có RBTV R<sub>5</sub> như sau:

R<sub>5</sub>:  $\forall t_1 \in r, t_2 \in s$   
 Nếu  $t_1.SODH = t_2.SODH$  thì  
 $t_1.NGAYDH \leq t_2.NGAYXUAT$   
 Cuối  $\forall$

R <sub>5</sub>	Thêm	Sửa	Xoá
Dathang	+	-	-
Hoandon	+	+	-

## 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.

## III. BÀI TẬP

**Bài tập 4.1:** 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, 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 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)

**Yêu cầu:**

- Tìm khoá cho mỗi lược đồ quan hệ trên.
- Hãy phát biểu các ràng buộc toàn vẹn có trong cơ sở dữ liệu trên.
- Thực hiện các yêu cầu sau bằng ngôn ngữ SQL:

- + Lập bảng điểm môn thi có mã môn thi là “CSDL02” cho tất các thí sinh có mã lớp là “CDTH2A”. danh sách cần MASV, HOTEN, NGAYSINH, DIEMTHI và được sắp xếp tăng dần theo MASV.
- + Hãy thống kê xem mỗi môn thi có bao nhiêu thí sinh có điểm thi lớn hơn hay bằng 5?
- + Danh sách cần: MAMT,TENMT,GHICHU,SOLUONG trong đó số lượng (SOLUONG) là thuộc tính tự đặt.
- + Lập danh sách những thí sinh đậu tốt nghiệp (theo tiêu chuẩn đã phân tích ở trên), danh sách cần: MASV, HOTEN, NGAYSINH, DIEMTONG, trong đó DIEMTONG bằng tổng điểm thi của 3 môn thi, DIEMTONG là thuộc tính tự đặt.
- + Nếu cần mở rộng bài toán theo hai hướng; Thứ nhất là quản lý kỳ thi tốt nghiệp cho tất các các khoa trong toàn trường, Thứ hai là quản lý thông tin về phòng thi (PHONGTHI) của mỗi thí sinh, thì lược đồ cơ sở dữ liệu trên cần phải được điều chỉnh như thế nào ?
- + Hãy phát biểu các ràng buộc toàn vẹn có trong lược đồ cơ sở dữ liệu trên.

**Bài tập 4.2:** Cho lược đồ cơ sở dữ liệu đã được phân tích ở **ví dụ 2.1 (mục II, Chương 2)**. Hãy phát biểu các ràng buộc toàn vẹn có trong lược đồ cơ sở dữ liệu trên.

**Bài tập 4.3:** Quản lý đăng ký chuyên đề.

Phòng giáo vụ tại một trường đại học muốn tin học hóa việc quản lý học các chuyên đề của sinh viên. Sau đây là kết quả của việc phân tích thiết kế ứng dụng trên:

Mỗi sinh viên có một mã số duy nhất, một họ tên, thuộc một phái, có một ngày sinh, một địa chỉ và học một ngành duy nhất.

Mỗi ngành có một mã ngành duy nhất, có một tên ngành duy nhất. Ngoài ra cũng cần lưu lại một con số cho biết số chuyên đề mà một sinh viên theo học một ngành cụ thể phải học, và cũng cần lưu lại tổng số sinh viên đã từng theo học ngành này.

Sinh viên phải học các chuyên đề khác nhau. Mỗi chuyên đề có một mã duy nhất và có một tên duy nhất. Cần lưu lại tên về số sinh viên tối đa có thể chấp nhận được mỗi khi có một lớp mở cho chuyên đề cụ thể.

Mỗi chuyên đề có thể được học bởi sinh viên thuộc nhiều ngành và sinh viên thuộc mỗi ngành phải học nhiều chuyên đề. Mỗi ngành học tối đa là 8 chuyên đề.

Vào mỗi học kỳ của mỗi năm học, ta cần lưu lại các chuyên đề nào được mở ra cho học kỳ của năm đó để sinh viên có thể đăng ký. Sinh viên chỉ được đăng ký những chuyên đề có mở.

Khi sinh viên đăng ký học, lưu lại việc đăng ký học một chuyên đề của một sinh viên vào một năm của một học kỳ nào đó.

Một sinh viên chỉ được đăng ký vào các chuyên đề thuộc ngành học của sinh viên đó mà thôi. Mỗi năm có 2 học kỳ. Sinh viên chỉ được đăng ký tối đa là 3 chuyên đề trong một học kỳ mà thôi.

- Hãy thiết kế mô hình ER cho ứng dụng trên.
- Chuyển mô hình ER sang mô hình quan hệ. Xác định khóa chính, khóa ngoại và liệt kê có phân loại tất cả ràng buộc toàn vẹn nhận diện được.

## CHƯƠNG 5. LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU

### I. CÁC VẤN ĐỀ GẶP PHẢI KHI TỔ CHỨC DỮ LIỆU

Khi thiết kế, tổ chức cơ sở dữ liệu quan hệ ta thường đứng trước vấn đề lựa chọn 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 cơ sở dữ liệu 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 sự dư thừa dữ liệu và các dị thường dữ liệu.

- Dư thừa dữ liệu là sự trùng lặp thông tin trong cơ sở dữ liệu.
- Dị thường dữ liệu là các sự cố xảy ra khi cập nhật dữ liệu (lập, dị thường chèn bộ, dị thường xóa bộ, dị thường sửa bộ) làm cho dữ liệu không tương thích, bất định hoặc mất mát.
  - + Dị thường do dữ liệu lập: một số thông tin có thể bị lặp lại một cách vô ích.
  - + Dị thường chèn bộ: không thể chèn bộ mới vào quan hệ, nếu không có đầy đủ dữ liệu.
  - + Dị thường xóa bộ: ngược lại với dị thường chèn bộ, việc xóa bộ có thể dẫn đến mất thông tin.
  - + Dị thường sửa bộ: việc sửa đổi dữ liệu dư thừa có thể dẫn đến sự không tương thích dữ liệu.

Cơ sở lý thuyết của việc thiết kế lược đồ cơ sở dữ liệu quan hệ tốt là khái niệm phụ thuộc dữ liệu. Phụ thuộc dữ liệu biểu diễn các quan hệ nhân quả giữa các thuộc tính trong quan hệ. Cũng dựa trên khái niệm phụ thuộc dữ liệu người ta định nghĩa các dạng chuẩn của lược đồ quan hệ. Còn quá trình biến đổi lược đồ thành lược đồ tương đương thỏa mãn dạng chuẩn gọi là quá trình chuẩn hóa lược đồ quan hệ.

### II. PHỤ THUỘC HÀM

#### 1. Định nghĩa phụ thuộc hàm



Cho lược đồ quan hệ  $R=(A_1, A_2, \dots, A_n)$  và  $X, Y$  là các tập con của  $R^+ = \{A_1, A_2, \dots, A_n\}$ . Ta nói rằng  $X$  xác định hàm  $Y$  hay  $Y$  phụ thuộc hàm  $X$ , ký hiệu  $X \rightarrow Y$ , nếu mọi quan hệ bất kỳ  $r$  của lược đồ  $R$  thoả mãn:

$$\forall u, v \in r : u(X) = v(X) \quad \Rightarrow u(Y) = v(Y)$$

Phụ thuộc hàm  $X \rightarrow Y$  gọi là phụ thuộc hàm *tầm thường* nếu  $Y \subset X$  (hiển nhiên là nếu  $Y \subset X$  thì theo định nghĩa ta có  $X \rightarrow Y$ ).

Phụ thuộc hàm  $X \rightarrow Y$  gọi là phụ thuộc hàm *nguyên tố* nếu không có tập con thực sự  $Z \subset X$  thoả  $Z \rightarrow Y$ .

Tập thuộc tính  $K \subset R$  gọi là *khoá* nếu nó xác định hàm tất cả các thuộc tính và  $K \rightarrow R$  là phụ thuộc hàm nguyên tố.

## 2. Cách xác định 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.

Ví dụ một số phụ thuộc hàm ứng với từng lược đồ quan hệ được xác định như sau:

$MASV \rightarrow HOTENSV, NGAYSINH, MALOP, GIOITINH$

$MALOP \rightarrow TENLOP, MAKHOA$

## 3. Một số tính chất của phụ thuộc hàm – hệ luật dẫn Armstrong

Để có thể xác định được các phụ thuộc hàm khác từ tập phụ thuộc hàm đã có, ta sử dụng các quy tắc suy diễn đơn giản để kiểm tra xem một phụ thuộc hàm có được suy diễn logic từ  $F$  hay không.

Một trong các quy tắc suy diễn đó gọi là hệ tiên đề Armstrong(1974), gồm các luật sau:

1. Luật phản xạ (reflexivity)  $X \rightarrow X$

2. Luật tăng trưởng (augmentation)  $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$

3. Luật bắc cầu (transitivity)  $X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$

Các quy tắc suy rộng:

4. Luật hợp (the union rule) Cho  $X \rightarrow Y, X \rightarrow Z \Rightarrow X \rightarrow YZ$

5. Luật bắc cầu giả (the pseudotransitivity rule)

$$\text{Cho } X \rightarrow Y, WY \rightarrow Z \Rightarrow XW \rightarrow Z$$

6. Luật phân rã (the decomposition rule)

$$\text{Cho } X \rightarrow YZ \Rightarrow X \rightarrow Z$$

Với  $X, Y, Z, W \in R^+$

**Ví dụ:**

Cho lược đồ  $R(ABC)$  và  $F=\{AB \rightarrow C, C \rightarrow A\}$ . Dùng các quy tắc Armstrong ta chứng minh rằng  $(B,C) \rightarrow (A,B,C)$ .

Thật vậy, ta có

$$C \rightarrow A \quad (\text{theo giả thiết})$$

$$BC \rightarrow AB \quad (\text{theo luật tăng trưởng})$$

$$C \rightarrow C \quad (\text{theo luật phản xạ})$$

$$\Rightarrow BC \rightarrow ABC \quad (\text{đccm}) \quad (\text{theo luật hợp})$$

### III. BAO ĐÓNG CỦA TẬP PHỤ THUỘC HÀM VÀ BAO ĐÓNG CỦA TẬP THUỘC TÍNH

#### 1. Bao đóng của tập phụ thuộc hàm $F$

Bao đóng của tập phụ thuộc hàm  $F$ , ký hiệu là  $F^+$ , là tập hợp tất cả các phụ thuộc hàm suy diễn logic từ  $F$ :

$$F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$$

Hay nói cách khác: Bao đóng (closure) của tập phụ thuộc hàm  $F$  (ký hiệu là  $F^+$ ) là tập hợp tất cả các phụ thuộc hàm có thể suy ra từ  $F$  dựa vào các tiên đề Armstrong. Rõ ràng  $F \subseteq F^+$

**Ví dụ:** Cho  $R=(A,B,C)$  và  $F = \{A \rightarrow B, B \rightarrow C\}$ . Khi đó bao đóng  $F^+$  gồm các phụ thuộc hàm  $X \rightarrow Y$  thỏa

(i)  $X$  chứa  $A$ ,  $Y$  bất kỳ:

$$A, B, C \rightarrow A, B, C; \quad A, B, C \rightarrow A, B; \quad A, B, C \rightarrow A, C; \quad A, B, C \rightarrow B, C;$$

$$A, B, C \rightarrow A; \quad A, B, C \rightarrow B; \quad A, B, C \rightarrow C;$$

$A, B \rightarrow A, B, C;$	$A, B \rightarrow A, B;$	$A, B \rightarrow A, C;$	$A, B \rightarrow B, C;$
$A, B \rightarrow A;$	$A, B \rightarrow B;$	$A, B \rightarrow B;$	$A, B \rightarrow C;$
$A, C \rightarrow A, B, C;$	$A, C \rightarrow A, B;$	$A, C \rightarrow A, C;$	$A, C \rightarrow B, C;$
$A, C \rightarrow A;$	$A, C \rightarrow B;$	$A, C \rightarrow B;$	$A, C \rightarrow C;$
$A \rightarrow A, B, C;$	$A \rightarrow A, B;$	$A \rightarrow A, C;$	$A \rightarrow B, C;$
$A \rightarrow A;$	$A \rightarrow B;$	$A \rightarrow B;$	$A \rightarrow C;$

(ii)  $X$  chứa  $B$  nhưng không chứa  $A$ ,  $Y$  không chứa  $A$ :

$BC \rightarrow BC;$	$BC \rightarrow B;$	$BC \rightarrow C$
$B \rightarrow BC;$	$B \rightarrow B;$	$B \rightarrow C$

(iii)  $C \rightarrow C$

Về mặt lý thuyết ta hoàn toàn có thể xây dựng thủ tục tính bao đóng  $\mathbf{F}^+$  của tập phụ thuộc hàm  $\mathbf{F}$ , nhưng trên thực tế bài toán xác định  $\mathbf{F}^+$  là không khả thi vì với số thuộc tính và phụ thuộc hàm lớn sẽ dẫn đến bùng nổ tổ hợp.

Thay vào đó chúng ta sẽ xem xét một bài toán khác: "Kiểm tra xem một phụ thuộc hàm có thuộc bao đóng  $\mathbf{F}^+$  hay không?". Bài toán này gọi là **bài toán thành viên**. Bài toán thành viên thiết thực hơn bài toán tính bao đóng vì trong thực tế rất hiếm khi phải tìm tất cả các phụ thuộc hàm suy diễn lô-gic từ  $\mathbf{F}$ .

Bài toán thành viên liên quan mật thiết với khái niệm bao đóng của tập thuộc tính.

## 2. Bao đóng của tập thuộc tính $X$

Bao đóng của tập thuộc tính  $X \subset R$  (đối với tập phụ thuộc hàm  $\mathbf{F}$ ), ký hiệu là  $X_{\mathbf{F}}^+$  ( $X^+$ ), là tập hợp tất cả các thuộc tính phụ thuộc hàm vào  $X$ :  $X^+ = \{A \mid X \rightarrow A \in \mathbf{F}^+\}$

Từ định nghĩa dễ dàng suy ra:  $X \subset X^+$  và  $X \rightarrow Y \Leftrightarrow Y \subset X^+$ . Nghĩa là  $X^+$  là tập thuộc tính lớn nhất phụ thuộc hàm vào  $X$ .

**Ví dụ:** Cho  $R(ABC)$  và  $F = \{A \rightarrow B, B \rightarrow C\}$ . Khi đó ta dễ dàng thấy bao đóng của thuộc tính B là  $B^+ = \{B, C\}$  vì  $B \rightarrow \{B, C\}$  và B không xác định A.

### 3. Bài toán thành viên

Qua phần trên ta nhận thấy  $X^+$  được định nghĩa thông qua  $F^+$ . Vấn đề nảy sinh khi nghiên cứu lý thuyết CSDL là: Cho trước tập các phụ thuộc hàm F và một phụ thuộc hàm f, bài toán kiểm tra có hay không  $f \in F^+$  gọi là *bài toán thành viên*.

Để giải quyết bài toán bài toán thành viên thật sự không đơn giản; vì mặc dù F là rất nhỏ nhưng  $F^+$  thì có thể rất lớn. Tuy nhiên ta có thể giải bằng cách tính  $X^+$  và so sánh  $X^+$  với tập Y. Dựa vào tính chất  $X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X^+$ , ta có ngay câu trả lời  $X \rightarrow Y \in F^+$  hay không? Như vậy thay vì giải bài toán thành viên ta đưa về giải bài toán tìm bao đóng của tập thuộc tính.

### 4. Thuật toán tìm bao đóng của một tập thuộc tính

Thuật toán tìm bao đóng với độ phức tạp  $O(N^2)$ , với N là số lượng thuộc tính của lược đồ quan hệ Q.

Dữ Liệu Vào    Q, F, X  $\subseteq$  Q<sup>+</sup>

Dữ Liệu Ra    X<sup>+</sup>

```

Bước 1:     Đặt X* = X
Bước 2:     Temp = X*
               $\forall f \quad U \rightarrow V \in F$ 
                 if (U  $\subseteq$  X*)
                       $X^* = X^* \cup V.$ 
                       $F = F - f;$ 
Bước 3:     if (X* = Temp)
                  "X* chính là kết quả cần tìm "
                  Dừng thuật toán
              else
                  trở lại Bước 2:
    
```

**Ví dụ:**

Cho lược đồ quan hệ  $Q(A,B,C,D,E,G,H)$  và tập phụ thuộc hàm

$$F = \{B \rightarrow A, DA \rightarrow CE, D \rightarrow H, GH \rightarrow C, AC \rightarrow D\}.$$

Tìm bao đóng của các tập  $X = \{AC\}$  dựa trên  $F$ .

Giải:

$$- X^+ = AC$$

$$- \text{Đặt } Temp = X^+$$

$$+ \text{ Xét } AC \rightarrow D, \text{ có } AC \subseteq X^+: X^+ = X^+ \cup D = ACD.$$

Loại  $AC \rightarrow D$  khỏi  $F$ . Lặp bước 2

$$+ \text{ Xét } DA \rightarrow CE, \text{ có } DA \subseteq X^+: X^+ = X^+ \cup CE = ACDE.$$

Loại  $DA \rightarrow CE$  khỏi  $F$ . Lặp bước 2

$$+ \text{ Xét } D \rightarrow H, \text{ có } D \subseteq X^+: X^+ = X^+ \cup H = ACDEH.$$

Loại  $D \rightarrow H$  khỏi  $F$  Lặp bước 2

Vì các phụ thuộc hàm  $U \rightarrow V$  còn lại không thỏa điều kiện  $U \subseteq X^+$  nên  $X^+ = Temp$ .  
Thuật toán dừng.

$$\text{Vậy } X^+ = \{ACDEH\}$$

#### IV. KHÓA CỦA LƯỢC ĐỒ QUAN HỆ - MỘT SỐ THUẬT TOÁN TÌM KHÓA

##### 1. Định nghĩa khóa của quan hệ

Cho quan hệ  $R(A_1, A_2, \dots, A_n)$  được xác định bởi tập thuộc tính  $R^+$  và tập phụ thuộc hàm  $F$  định nghĩa trên  $R$ , cho  $K \subseteq R^+$ .

$K$  là một *khoá* của  $R$  nếu thỏa đồng thời cả hai điều kiện sau:

$$1. K \subseteq R^+ \in F^+ \text{ (hay } K^+ = R^+)$$

( $K$  chỉ thỏa điều kiện 1 thì được gọi là siêu khoá)

$$2. \text{ Không tồn tại } K' \subset K \text{ sao cho } K'^+ = R^+$$

Tập  $S \subseteq \{A_1, \dots, A_n\}$  là *siêu khoá* của  $R$  nếu  $S$  chứa khoá. Một lược đồ quan hệ có thể có nhiều siêu khoá, nhiều khoá.

##### 2. Thuật toán tìm một khóa của một lược đồ quan hệ

$$K = Q+;$$

While  $A \in K$  do

if  $(K - A)^+ = Q^+$  then  $K = K - A$

K còn lại chính là một khóa cần tìm.

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:**

Cho lược đồ quan hệ  $R(ABC)$  và tập phụ thuộc hàm

$$F = \{ A \rightarrow B; A \rightarrow C; B \rightarrow A \}$$

Hãy tìm một khóa của R.

Giải:

$$K = \{A, B, C\}$$

Loại thuộc tính A, do  $(K - A)^+ = R^+$  nên  $K = \{B, C\}$

thuộc tính B không loại được do  $(K - B)^+ \neq R^+$  nên  $K = \{B, C\}$

Loại thuộc tính C, do  $(K - C)^+ = R^+$  nên  $K = \{B\}$ . Vậy một khóa của R là B.

### 3. Thuật toán tìm tất cả các khóa của một lược đồ quan hệ

Một số khái niệm hỗ trợ cho thuật toán tìm tất cả các khóa sau đây:

- Tập nguồn (TN): chứa tất cả thuộc tính chỉ xuất hiện ở vế trái mà không xuất hiện ở vế phải của tập phụ thuộc hàm và tập các thuộc tính không tham gia vào tập phụ thuộc hàm F.
- Tập đích (TD): chứa tất cả các thuộc tính chỉ xuất hiện ở vế phải mà không xuất hiện ở vế trái của tập phụ thuộc hàm.
- Tập trung gian (TG): chứa tất cả các thuộc tính tham gia vào cả 2 vế của tập phụ thuộc hàm.
- Dữ liệu vào: Lược đồ quan hệ R và tập phụ thuộc hàm F.
- Dữ liệu ra: Tất cả các khóa K của quan hệ.

**Thuật toán:**

- Bước 0: Tìm tập thuộc tính nguồn (TN), tập thuộc tính trung gian (TG).

Tìm tất cả các tập con của tập trung gian gọi là  $X_i$  (bằng phương pháp duyệt nhị phân)

if  $TG = \theta$  then

$K = TN$  ; kết thúc.

Ngược lại

Qua bước 1

- Bước 1: Tìm tất cả các tập con của TG:  $X_i$

$S = \emptyset$

$\forall X_i \in TG$

If  $(TN \cup X_i)^+ = R^+$  then

$S = S \cup \{TN \cup X_i\}$

$\{S$  là tập các siêu khoá cần tìm $\}$

- Bước 2: Tính  $TN \cup X_i$
- Bước 3: Tính  $(TN \cup X_i)^+$
- Bước 4: Nếu  $X_i^+ = R^+$  thì  $X_i$  là siêu khoá

Nếu một tập con  $TN \cup X_i$  có bao đóng đúng bằng  $R^+$  thì  $TN \cup X_i$  là một siêu khoá của  $R$ .

Giả sử sau bước này có  $m$  siêu khoá:  $S = \{S_1, S_2, \dots, S_m\}$

- Bước 5 : Xây dựng tập chứa tất cả các khoá của  $R$  từ tập  $S$

Xét mọi  $S_i, S_j$  con của  $S$  ( $i \neq j$ ), nếu  $S_i \subset S_j$  thì ta loại  $S_j$  ( $i, j = 1..m$ ), kết quả còn lại chính là tập tất cả các khoá cần tìm.

**Ví dụ:** Cho lược đồ quan hệ  $R(A,B,C)$  và tập phụ thuộc hàm

$F = \{A \rightarrow B; A \rightarrow C; B \rightarrow A\}$

Hãy tìm tất cả các khóa của  $R$ .

Giải: Áp dụng thuật tìm tất cả các khóa đã cho ở trên ta có:

$$TN = \{\phi\}; TG = \{A, B\}$$

Gọi  $X_i$  là tập con của tập trung gian. Ta lập bảng như sau:

$X_i$	$TN \cup X_i$	$(TN \cup X_i)^+$	Siêu khóa	Khóa
$\phi$	$\phi$	$\phi$	-	-
A	A	ABC	A	A
B	B	ABC	B	B
AB	AB	ABC	AB	-

Vậy lược đồ quan hệ R có hai khóa  $K1 = \{A\}$ ,  $K2 = \{B\}$

## V. PHỦ TỐI THIỂU

### 1. Tập phụ thuộc hàm tương đương

Cho F và G là hai tập phụ thuộc hàm, ta nói F và G tương đương (hay F phủ G hoặc G phủ F) và ký hiệu là  $F^+ = G^+$  nếu và chỉ nếu mỗi phụ thuộc hàm thuộc F đều thuộc  $G^+$  và mỗi phụ thuộc hàm thuộc G đều thuộc  $F^+$ .

Ta nói F phủ G nếu  $G^+ \subseteq F^+$

Chẳng hạn cho lược đồ quan hệ  $Q(ABCDEFGH)$ , thì hai tập phụ thuộc hàm F và G (xác định trên Q) là tương đương.

$$F = \{B \rightarrow A; DA \rightarrow CE; D \rightarrow H; GH \rightarrow C; AC \rightarrow D; DG \rightarrow C\}$$

$$G = \{B \rightarrow A; DA \rightarrow CE; D \rightarrow H; GH \rightarrow C; AC \rightarrow D; BC \rightarrow AC; BC \rightarrow D;$$

$$DA \rightarrow AH; AC \rightarrow DEH\}$$

### 2. Phủ tối thiểu

$F_{tt}$  được gọi là tập phụ thuộc hàm tối thiểu (hay phủ tối thiểu) nếu F thỏa đồng thời ba điều kiện sau:

1. F là tập phụ thuộc hàm có vế trái không dư thừa.
2. F là tập phụ thuộc hàm có vế phải một thuộc tính.
3. F là tập phụ thuộc hàm không dư thừa.

#### 2.1. Phụ thuộc hàm có vế trái dư thừa



$F$  là tập phụ thuộc hàm trên lược đồ quan hệ  $Q$ ,  $Z$  là tập thuộc tính,  $Z \rightarrow Y \in F$ . Nói rằng phụ thuộc hàm  $Z \rightarrow Y$  có vế trái dư thừa (phụ thuộc không đầy đủ) nếu có một  $A \in Z$  sao cho:

$$F \equiv F - \{Z \rightarrow Y\} \cup \{(Z-A) \rightarrow Y\}$$

Ngược lại  $Z \rightarrow Y$  là phụ thuộc hàm có vế trái không dư thừa hay  $Y$  phụ thuộc hàm đầy đủ vào  $Z$  (phụ thuộc hàm đầy đủ).

Ta nói  $F$  là tập phụ thuộc hàm có vế trái không dư thừa nếu  $F$  không chứa phụ thuộc hàm có vế trái dư thừa.

### **Thuật toán loại khỏi $F$ các phụ thuộc hàm có vế trái dư thừa**

- Bước 1: - Xét lần lượt các phụ thuộc hàm  $X \rightarrow Y$  của  $F$ .
- Bước 2: - Với mọi tập con thực sự  $X' \neq \emptyset$  của  $X$ .
  - Nếu  $X' \rightarrow Y \in F^+$  thì thay  $X \rightarrow Y$  trong  $F$  bằng  $X' \rightarrow Y$ .
  - Lặp lại bước 2.

### **2.2. Tập phụ thuộc hàm có vế phải một thuộc tính**

Mỗi tập phụ thuộc hàm  $F$  đều tương đương với tập phụ thuộc hàm  $G$  mà vế phải của các phụ thuộc hàm trong  $G$  chỉ gồm một thuộc tính.

$G$  được gọi là tập phụ thuộc hàm có vế phải một thuộc tính.

**Ví dụ:**

$F = \{A \rightarrow BC, B \rightarrow C, AB \rightarrow D\}$  ta suy ra

$$F \equiv \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow D\} = G$$

### **2.3. Tập phụ thuộc hàm không dư thừa**

Nói rằng  $F$  là tập phụ thuộc hàm không dư thừa nếu không tồn tại  $F' \subset F$  sao cho  $F' \equiv F$ . Ngược lại  $F$  là tập phụ thuộc hàm dư thừa.

### **Thuật toán loại khỏi $F$ các phụ thuộc hàm dư thừa:**

- Bước 1: - Lần lượt xét các phụ thuộc hàm  $X \rightarrow Y$  của  $F$
- Bước 2: - Nếu  $X \rightarrow Y$  là thành viên của  $F - \{X \rightarrow Y\}$  thì loại  $X \rightarrow Y$  khỏi  $F$ .
- Bước 3: - Lặp lại bước 2 cho các phụ thuộc hàm tiếp theo của  $F$ .

### 3. Thuật toán tìm phủ tối thiểu

Từ điều kiện xác định phủ tối thiểu, ta có thuật toán tìm phủ tối thiểu như sau:

**Thuật toán:**

- Bước 1: - Loại khỏi F các phụ thuộc hàm có vế trái dư thừa.
- Bước 2: - Tách các phụ thuộc hàm có vế phải trên một thuộc tính thành các phụ thuộc hàm có vế phải một thuộc tính.
- Bước 3: - Loại khỏi F các phụ thuộc hàm dư thừa.

**Chú ý:** Theo thuật toán trên, có thể tìm được nhiều hơn một phủ tối thiểu  $F_{tt}$  để  $F \equiv F_{tt}$  và nếu thứ tự loại các phụ thuộc hàm khác nhau sẽ thu được các phủ tối thiểu khác nhau.

**Ví dụ:** cho  $R(MSCD, MSSV, CD, HG)$  và tập phụ thuộc hàm F:

$F = \{MSCD \rightarrow CD; CD \rightarrow MSCD; CD, MSSV \rightarrow HG; MSCD, HG \rightarrow MSSV; CD, HG \rightarrow MSSV; MSCD, MSSV \rightarrow HG\}$

Hãy tìm một  $F_{tt}$  của F?

Kết quả ta có được một phủ tối thiểu sau:

$F_{tt} = \{MSCD \rightarrow CD; CD \rightarrow MSCD; CD, HG \rightarrow MSSV; MSCD, MSSV \rightarrow HG\}$

## VI. DẠNG CHUẨN CỦA LƯỢC ĐỒ QUAN HỆ

### 1. Một số khái niệm liên quan đến các dạng chuẩn

*Thuộc tính khóa/thuộc tính không khóa:* A là thuộc tính khóa nếu A có tham gia vào bất kỳ một khóa nào đó của quan hệ. Ngược lại A gọi là thuộc tính không khóa.

*Thuộc tính phụ thuộc đầy đủ/ Phụ thuộc hàm đầy đủ:* A là một thuộc tính phụ thuộc đầy đủ vào tập thuộc tính X nếu  $X \rightarrow A$  là một phụ thuộc hàm đầy đủ (tức là không tồn tại  $X' \subsetneq X$  sao cho  $X' \rightarrow A \in F^+$ )

Chú ý rằng một phụ thuộc hàm mà vế trái chỉ có một thuộc tính là phụ thuộc hàm đầy đủ.

### 2. Dạng chuẩn 1 (First Normal Form)

**Định nghĩa:** Lược đồ quan hệ R đạt dạng chuẩn 1 (1NF) nếu và chỉ nếu toàn bộ các thuộc tính của mọi bộ trên R đều mang giá trị đơn.

**Ví dụ:** Xét quan hệ KETQUA sau:

MASV	HOVATEN	KHOA	TENMONHOC	DIEMTHI
01234	Nguyễn Văn An	CNTT	Cơ sở dữ liệu	6
			Toán rời rạc	8
			Lập trình web	7
02345	Lê Văn Thịnh	CNTT	Cơ sở dữ liệu	7

Quan hệ này không đạt chuẩn 1NF vì các thuộc tính TENMONHOC, DIEMTHI của bộ thứ nhất không mang giá trị đơn.

Ta có thể đưa quan hệ trên về quan hệ KETQUA1 đạt chuẩn 1 như sau:

MASV	HOVATEN	KHOA	TENMONHOC	DIEMTHI
01234	Nguyễn Văn An	CNTT	Cơ sở dữ liệu	6
01234	Nguyễn Văn An	CNTT	Toán rời rạc	8
001234	Nguyễn Văn An	CNTT	Lập trình web	7
02345	Lê Văn Thịnh	CNTT	Cơ sở dữ liệu	7

Chú ý rằng khi xét các dạng chuẩn, nếu không xét gì thêm thì mặc định quan hệ đang xét ít nhất đạt dạng chuẩn 1.

### 3. Dạng chuẩn 2 (Second Normal Form)

**Định nghĩa:** Một lược đồ quan hệ R ở dạng chuẩn 2 (2NF) nếu R đạt dạng chuẩn 1 và mọi thuộc tính không khóa của R đều phụ thuộc đầy đủ vào khóa.

**Hệ quả:**

1. Nếu R đạt dạng chuẩn 1 và tập thuộc tính không khóa của R bằng rỗng thì R đạt chuẩn 2.
2. Nếu tất cả các khóa quan hệ chỉ gồm một thuộc tính thì quan hệ đó ít nhất đạt chuẩn 2.

**Thuật toán kiểm tra dạng chuẩn 2:**

Vào: lược đồ quan hệ R, tập phụ thuộc hàm F

### Dạng chuẩn 3 (Third Normal Form)

Ra: Khẳng định R đạt hoặc không đạt chuẩn 2.

- Bước 1: Tìm tất cả các khóa của R.
- Bước 2: Với mỗi khóa K, tìm bao đóng của tất cả tập con thực sự của K.
- Bước 3: Nếu có bao đóng  $S^+$  chứa thuộc tính không khóa thì R không đạt chuẩn 2. Ngược lại thì đạt chuẩn 2.

Ví dụ:

Cho lược đồ quan hệ R(ABCD) và tập phụ thuộc hàm

$F = \{AB \rightarrow C; B \rightarrow D; BC \rightarrow A\}$ . Hỏi R có đạt chuẩn 2 hay không?

Giải:

Tìm tất cả các khóa của R:

$TN = \{B\}$ ,  $TG = \{AC\}$

$X_i$	$TN \cup X_i$	$(TN \cup X_i)^+$	Siêu khóa	Khóa
$\phi$	B	BD	-	-
A	BA	BACD	BA	BA
C	BC	BCAD	BC	BC
AC	BAC	BACD	BAC	-

Tất cả các khóa của R là  $K_1 = \{BA\}$ ,  $K_2 = \{BC\}$ . Gọi Z là tập thuộc tính khóa, X là tập thuộc tính không khóa, ta có:

$$Z = K_1 \cup K_2 = \{BAC\}$$

$$X = R^+ \setminus Z = \{ABCD\} \setminus \{BAC\} = \{D\}$$

Ta thấy  $B \subset K_1$ ,  $B \rightarrow D$ , mà D là thuộc tính không khóa. Vì thuộc tính không khóa D không phụ thuộc đầy đủ vào khóa nên R không đạt chuẩn 2.

### 4. Dạng chuẩn 3 (Third Normal Form)

**Định nghĩa:** Một lược đồ quan hệ R đạt chuẩn 3 (3NF) nếu mọi phụ thuộc hàm  $X \rightarrow A$   $\in F^+$  với  $A \notin X$  đều có.

- Hoặc X là siêu khóa.

- Hoặc A là thuộc tính khóa

**Hệ quả:**

1. Nếu R đạt chuẩn 3 thì R đạt chuẩn 2.
2. Nếu R không có thuộc tính không khóa thì R đạt chuẩn 3.

**Định lý:**

R là một lược đồ quan hệ.

F là tập các phụ thuộc hàm có vẻ phải một thuộc tính.

R đạt chuẩn 3 nếu và chỉ nếu mọi phụ thuộc hàm  $X \rightarrow A \in F^+$  với  $A \notin X$  đều có.

- Hoặc X là siêu khóa.
- Hoặc A là thuộc tính khóa

**Thuật toán kiểm tra dạng chuẩn 3:**

Vào: lược đồ quan hệ R, tập phụ thuộc hàm F.

Ra: Khẳng định R đạt hoặc không đạt chuẩn 3.

- Bước 1: Tìm tất cả các khóa của R.
- Bước 2: Từ F tạo tập phụ thuộc hàm tương đương Ftt có vẻ phải một thuộc tính.
- Bước 3: Nếu mọi phụ thuộc hàm  $X \rightarrow A \in Ftt$  với  $A \notin X$  đều có X là siêu khóa hoặc A là thuộc tính khóa thì R đạt chuẩn 3. Ngược lại R không đạt chuẩn 3.

**Ví dụ:** Cho lược đồ quan hệ R(ABCD),  $F = \{AB \rightarrow C; D \rightarrow B; C \rightarrow ABD\}$ . Hỏi R có đạt chuẩn 3 hay không?

**Giải:**

Tìm tất cả các khóa của R:  $TN = \{\emptyset\}$   $TG = \{ABCD\}$

$X_i$	$TN \cup X_i$	$(TN \cup X_i)^+$	Siêu khóa	Khóa
$\phi$	$\phi$	$\phi$	-	-
A	A	A	-	-
B	B	B	-	-

C	C	CABD	C	C
D	D	DB	-	-
AB	AB	ABCD	AB	AB
AC	AC	ACBD	AC	-
AD	AD	ADBC	AD	AD
BC	BC	BCAD	BC	-
BD	BD	BD	-	-
CD	CD	CDAB	CD	-
ABC	ABC	ABCD	ABC	-
ABD	ABD	ABDC	ABD	-
ACD	ACD	ACDB	ACD	-
BCD	BCD	BCDA	BCD	-
ABCD	ABCD	ABCD	ABCD	-

Tất cả các khóa của R là  $K_1 = \{C\}$ ,  $K_2 = \{AB\}$ ,  $K_3 = \{AD\}$ . Gọi  $Z$  là tập thuộc tính khóa,  $X$  là tập thuộc tính không khóa, ta có:

$$Z = K_1 \cup K_2 \cup K_3 = \{CABD\}$$

$$X = R^+ \setminus Z = \{ABCD\} \setminus \{CABD\} = \{\emptyset\}$$

Vì tập thuộc tính không khóa  $X = \{\emptyset\}$  nên R đạt chuẩn 3 (theo hệ quả 2).

## 5. Dạng chuẩn BC (Boyce Codd Normal Form)

**Định nghĩa:** Một lược đồ quan hệ R đạt dạng chuẩn BC nếu mọi phụ thuộc hàm  $X \rightarrow A$   $\in F^+$  với  $A \notin X$  đều có X là siêu khóa.

**Hệ quả:**

1. Nếu R đạt chuẩn BC thì R đạt chuẩn 3 (hiển nhiên do định nghĩa).
2. Mỗi lược đồ có hai thuộc tính đều đạt chuẩn BC (xét phụ thuộc hàm có thể có của R).

**Định lý:**

R là lược đồ quan hệ.

F là tập phụ thuộc hàm có vẻ phải một thuộc tính.

R đạt chuẩn BC nếu và chỉ nếu mọi phụ thuộc hàm  $X \rightarrow A \in F^+$  với  $A \notin X$  đều có X là siêu khóa.

**Thuật toán kiểm tra dạng chuẩn BC:**

Vào: lược đồ quan hệ R, tập phụ thuộc hàm F.

Ra: Khẳng định R đạt hoặc không đạt chuẩn BC.

- Bước 1: Tìm tất cả các khóa của R.
- Bước 2: Từ F tạo tập phụ thuộc hàm tương đương  $F_{tt}$  có vẻ phải một thuộc tính.
- Bước 3: Nếu mọi phụ thuộc hàm  $X \rightarrow A \in F_{tt}$  với  $A \notin X$  đều có X là siêu khóa thì R đạt chuẩn BC. Ngược lại R không đạt chuẩn BC.

**Ví dụ:**

Cho lược đồ quan hệ R(ABCDEI),  $F = \{ACD \rightarrow EBI; CE \rightarrow AD\}$ . Hỏi R có đạt chuẩn BC hay không?

**Giải:**

- Tìm tất cả các khóa của R:  $TN = \{C\}$   $TG = \{ADE\}$

$X_i$	$TN \cup X_i$	$(TN \cup X_i)^+$	Siêu khóa	Khóa
$\phi$	C	C	-	-
A	CA	CA	-	-
D	CD	CD	-	-
E	CE	CEADBI	CE	CE
AD	CAD	CADEBI	CAD	CAD
AE	CAE	CAEDBI	CAE	-
DE	CDE	CDEABI	CDE	-
ADE	CADE	CADEBI	CADE	-

Tất cả các khóa của R là  $K_1 = \{CE\}$ ,  $K_2 = \{CAD\}$ . Gọi  $Z$  là tập thuộc tính khóa,  $X$  là tập thuộc tính không khóa, ta có:

$$Z = K_1 \cup K_2 = \{CEAD\}$$

$$X = R^+ \setminus Z = \{ABCDEI\} \setminus \{CEAD\} = \{BI\}$$

- Tìm  $F_{tt}$  có vẻ phải một thuộc tính

$$F_{tt} = \{ACD \rightarrow E; ACD \rightarrow B; ACD \rightarrow I; CE \rightarrow A; CE \rightarrow D\}$$

Ta nhận thấy mọi phụ thuộc hàm trong  $F_{tt}$  đều có vế trái là một siêu khóa nên R đạt chuẩn BC.

### ***Thuật toán kiểm tra dạng chuẩn của một lược đồ quan hệ.***

*Vào : lược đồ quan hệ R, tập phụ thuộc hàm F.*

*Ra : khẳng định R đạt chuẩn gì?*

- Bước 1: Tìm tất cả các khóa của R.
- Bước 2: Kiểm tra chuẩn BC, nếu đúng R đạt chuẩn BC, kết thúc thuật toán. Ngược lại qua bước 3.
- Bước 3: Kiểm tra chuẩn 3, nếu đúng R đạt chuẩn 3, kết thúc thuật toán, ngược lại qua bước 4.
- Bước 4: Kiểm tra dạng chuẩn 2, nếu đúng R đạt chuẩn 2, kết thúc thuật toán, ngược lại R đạt chuẩn 1.

***Định nghĩa:*** Dạng chuẩn của một lược đồ cơ sở dữ liệu là dạng chuẩn thấp nhất trong các dạng chuẩn của các lược đồ quan hệ con.

## **VII. BÀI TẬP**

**Bài tập 5.1.** Cho lược đồ quan hệ  $R(B,C,D,E,F,G,I,J)$  và tập các phụ thuộc hàm

$$P = \{I \rightarrow B; DE \rightarrow GJ; D \rightarrow C; CF \rightarrow J; B \rightarrow I; C \rightarrow G; F \rightarrow J\}$$

- Tìm tất cả các khóa của lược đồ quan hệ R.
- Tìm một phủ tối thiểu của tập phụ thuộc hàm P trên.

**Bài tập 5.2.** Cho lược đồ quan hệ  $S(A,B,C,D,E,F,G)$  và tập các phụ thuộc hàm

$$P = \{B \rightarrow FG; F \rightarrow CE; G \rightarrow BD; B \rightarrow A\}$$



Chứng tỏ rằng phụ thuộc hàm  $AB \rightarrow D$  được suy diễn từ P nhờ hệ luật dẫn Amstrong?  
(Nêu rõ là áp dụng luật gì)

### Bài tập 5.3.

a. Cho lược đồ quan hệ  $Q(ABCD)$ ,  $r$  là một quan hệ trên  $Q$ .

$r$			
A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d1
a1	b3	c2	d1
a2	b2	c2	d2

phụ thuộc hàm nào sau đây không thoả  $r$ :  $D \rightarrow A$ ;  $AC \rightarrow D$ ;  $CD \rightarrow A$ ;  $D \rightarrow B$ ;

b. Cho lược đồ quan hệ  $Q(ABCD)$ ,  $r$  là quan trên  $Q$  được cho như sau:

A	B	C	D
a1	b1	c1	d2
a3	b1	c2	d1
a1	b1	c2	d2

Những phụ thuộc hàm nào sau đây thoả  $r$  ?

$AB \rightarrow D$ ;  $C \rightarrow B$ ;  $B \rightarrow C$ ;  $BC \rightarrow A$ ;  $BD \rightarrow A$ .

c. Cho lược đồ quan hệ  $Q(ABCD)$ ,  $r$  là quan hệ được cho như sau:

A	B	C	D
x	u	x	y
y	x	z	x
z	y	y	y
y	z	w	z

Những phụ thuộc hàm nào sau đây không thoả  $r$  ?

$A \rightarrow B$ ;  $A \rightarrow C$ ;  $B \rightarrow A$ ;  $C \rightarrow D$ ;  $D \rightarrow C$ ;  $D \rightarrow A$

### Bài tập 5.4.

a. Cho lược đồ quan hệ  $Q(ABCD)$  và tập phụ thuộc hàm  $F = \{A \rightarrow B; BC \rightarrow D\}$ .

Những phụ thuộc hàm nào sau đây thuộc  $F^+$  ?

$C \rightarrow D$ ;  $A \rightarrow D$ ;  $AD \rightarrow C$ ;  $AC \rightarrow D$ ;  $BC \rightarrow A$ ;  $B \rightarrow CD$ .

b. Cho lược đồ quan hệ  $Q(ABCDEFGH)$  và tập phụ thuộc hàm

$$F = \{ AB \rightarrow C; B \rightarrow D; CD \rightarrow E; CE \rightarrow GH; G \rightarrow A \}$$

Những phụ thuộc hàm nào sau đây không thuộc vào  $F^+$  ?

$$AB \rightarrow E; AB \rightarrow GH; CGH \rightarrow E; CB \rightarrow E; GB \rightarrow E.$$

c. Cho lược đồ quan hệ  $Q, F$  như sau: với  $Q(ABCD)$   $F = \{ A \rightarrow B; A \rightarrow C \}$ .

Trong các phụ thuộc hàm sau, những phụ thuộc hàm được suy ra từ  $F$  ?

$$A \rightarrow D; C \rightarrow D; AB \rightarrow B; \quad BC \rightarrow A; \quad A \rightarrow BC$$

**Bài tập 5.5.** Cho lược đồ quan hệ  $Q(ABCD)$  và tập phụ thuộc hàm

$$F = \{ A \rightarrow D; D \rightarrow A; AB \rightarrow C \}$$

a. Tính  $AC^+$

b. Chứng minh  $BD \rightarrow C$

**Bài tập 5.6.** Cho lược đồ quan hệ  $Q(A,B,C,D,E)$  và tập các phụ thuộc hàm

$$F = \{ A \rightarrow BC; C \rightarrow DE \}$$

a. Lược đồ quan hệ  $Q$  đạt dạng chuẩn nào?

b. Nếu chưa đạt dạng chuẩn 3 (3NF) hãy phân rã  $Q$  thành lược đồ quan hệ đạt tối thiểu dạng chuẩn 3 bảo toàn thông tin.

**Bài tập 5.7.** Cho biết dạng chuẩn của các lược đồ quan hệ sau:

a.  $Q(ABCDEG); \quad F = [A \rightarrow BC, C \rightarrow DE, E \rightarrow G]$

b.  $Q(ABCDEGH); \quad F = [C \rightarrow AB, D \rightarrow E, B \rightarrow G]$

c.  $Q(ABCDEGH); \quad F = [A \rightarrow BC, D \rightarrow E, H \rightarrow G]$

d.  $Q(ABCDEG); \quad F = [AB \rightarrow C; C \rightarrow B; ABD \rightarrow E; G \rightarrow A]$

e.  $Q(ABCDEGHI); F = [AC \rightarrow B; BI \rightarrow ACD; ABC \rightarrow D; H \rightarrow I; ACE \rightarrow BCG, CG \rightarrow AE]$

## **TÀI LIỆU THAM KHẢO**

### **A. TÀI LIỆU TIẾNG VIỆT**

- [1]. Vũ Đức Thi, 1997. *Cơ sở dữ liệu kiến thức và thực hành*. Nhà xuất bản thống kê.
- [2]. Tô Văn Nam, 2005. *Giáo trình cơ sở dữ liệu*, Nhà xuất bản Giáo dục.
- [3]. Phạm Thế Quế, 2006. *Cơ sở dữ liệu*. Học viện công nghệ bưu chính viễn thông.
- [4]. Nguyễn Kim Anh, 2005. *Nguyên lý của các hệ cơ sở dữ liệu*, Tái bản lần 2, Nhà xuất bản Đại học Quốc gia Hà nội.

### **B. TÀI LIỆU TIẾNG ANH**

- [5]. Neeraj Sharma, 2010. *Database Fundamentals*. IBM Corporation.
- [6]. Hector Garcia-Molina, 2002. *Database Systems, The Complete Book, Second edition*. Pearson Education.
- [7]. Joe Celko, 1999. *Data and Databases: Concepts in Practice*. Morgan Kaufmann Publishers.