

**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**CAO THỊ NHẬN**

**NGUYỄN THỊ THANH BÌNH**

**GIÁO TRÌNH**

**CƠ SỞ DỮ LIỆU**

**Dành cho sinh viên ngành Công nghệ Thông tin**

**Đà Lạt 2010**

## LỜI MỞ ĐẦU

Để đáp ứng nhu cầu học tập của sinh viên chuyên ngành Công nghệ Thông tin, bài giảng Cơ sở dữ liệu được biên soạn theo chương trình hệ thống tín chỉ của Trường Đại học Đà Lạt, cung cấp các kiến thức cơ bản về lý thuyết cơ sở dữ liệu.

Giáo trình gồm 8 chương sau:

**Chương 1:** giới thiệu chung

**Chương 2:** trình bày mô hình thực thể kết hợp E\_R (Entity Relationship) để mô hình hóa các hoạt động trong thế giới thực, nhìn thế giới thực như là một tập các đối tượng căn bản được gọi là các thực thể, và các mối quan hệ ở giữa các đối tượng này. Mô hình được phát triển để làm thuận tiện cho việc thiết kế cơ sở dữ liệu bằng cách đặc tả một tổ chức.

**Chương 3:** trình bày mô hình dữ liệu quan hệ, các quy tắc chuyển đổi cơ sở dữ liệu biểu diễn dạng lược đồ E\_R sang mô hình dữ liệu quan.

**Chương 4, 5 và 6:** trình bày các ngôn ngữ truy vấn trên cơ sở dữ liệu. Trong chương 4 trình bày ngôn ngữ đại số quan hệ, chương 5 trình bày ngôn ngữ tân từ và chương 6 trình bày ngôn ngữ truy vấn SQL.

**Chương 7:** trình bày các khái niệm liên quan đến phụ thuộc hàm, khóa, các thuật toán tìm phủ tối thiểu của tập phụ thuộc hàm, tìm khóa và cuối chương trình bày về ràng buộc toàn vẹn dữ liệu.

**Chương 8:** Chương này giới thiệu các dạng chuẩn, phân rã bảo toàn thông tin, bảo toàn phụ thuộc hàm, qua đó cũng trình bày cách phân rã bảo toàn bảo toàn thông tin và bảo toàn phụ thuộc.

Mặc dù đã rất cố gắng trong quá trình biên soạn bài giảng nhưng chắc chắn bài giảng sẽ còn nhiều thiếu sót và hạn chế. Rất mong nhận được sự đóng góp ý kiến quý báu của sinh viên và các bạn đọc để giáo trình ngày một hoàn thiện hơn.

## MỤC LỤC

<i>Lời mở đầu</i> .....	2
Mục lục .....	3
<b>Chương 1: Giới thiệu chung</b> .....	<b>8</b>
1. Giới thiệu chung.....	8
1.1. Giới thiệu .....	8
1.2. Định nghĩa CSDL .....	10
1.3. Các đối tượng sử dụng CSDL. ....	11
1.4. Hệ quản trị CSDL .....	12
1.5. Các mức biểu diễn một CSDL.....	12
1.6. Sơ đồ tổng quát một hệ quản trị CSDL .....	14
1.7. Tính độc lập giữa dữ liệu và chương trình. ....	15
2. Các cách tiếp cận của một CSDL.....	15
2.1. Cách tiếp cận theo mô hình dữ liệu mạng .....	16
2.2. Mô hình dữ liệu phân cấp .....	17
2.3. Mô hình dữ liệu quan hệ thực thể.....	18
2.4. Mô hình dữ liệu quan hệ.....	18
2.5. Mô hình dữ liệu hướng đối tượng .....	19
3. Bài tập.....	19
<b>Chương 2: Mô hình thực thể kết hợp</b> .....	<b>21</b>
1. Mô hình thực thể kết hợp. ....	21
1.1. Thực thể - tập thực thể.....	21
1.2. Thuộc tính.....	21
1.3. Mối kết hợp.....	22
1.4. Bản số .....	23
1.5. Khoá.....	24

1.6. Số chiều của mỗi kết hợp.....	25
1.7. Tổng quát hóa và chuyên biệt hóa.....	25
1.8. Tập thực thể yếu .....	26
2. Ví dụ .....	27
3. Bài tập.....	29
 <b>Chương 3: Mô hình dữ liệu quan hệ .....</b>	<b>30</b>
1. Các khái niệm cơ bản .....	30
1.1. Thuộc tính.....	30
1.2. Quan hệ n ngôi.....	31
1.3. Bộ.....	31
1.4. Lược đồ quan hệ .....	32
1.5. Khóa của một quan hệ .....	33
1.6. Ràng buộc toàn vẹn .....	35
2. Các thao tác cơ bản trên quan hệ.....	35
2.1. Phép thêm .....	35
2.2. Phép xóa .....	36
2.3. Phép sửa.....	36
3. Các bước chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ .....	37
3.1. Biến các tập thực thể chuyên biệt hóa về dạng bình thường.....	37
3.2. Chuyển tất cả các tập thực thể thành quan hệ .....	38
3.3. Mỗi kết hợp.....	39
3.4. Nhập tất cả các quan hệ có cùng khóa.....	39
 <b>Chương 4: Ngôn ngữ đại số quan hệ .....</b>	<b>40</b>

1. Các phép toán cơ sở .....	40
1.1. Các phép toán tập hợp .....	40
1.2. Các phép toán quan hệ .....	41
2. Các phép toán khác .....	46
2.1. Phép kết hai quan hệ .....	46
2.2. Phép kết nối nội .....	47
2.3. Phép kết nối trái .....	48
2.4. Phép kết nối phải .....	49
2.5. Hàm kết hợp và gom nhóm .....	49
2.6. Các phép toán cập nhật trên quan hệ .....	50
3. Bài tập.....	52

## **Chương 5: Ngôn ngữ tân từ ..... 55**

1. Ngôn ngữ tân từ có biến là bộ .....	55
1.1. Một số khái niệm .....	55
1.2. Định nghĩa hình thức của phép tính bộ .....	55
1.3. Lượng từ tồn tại $\exists$ và với mọi $\forall$ .....	57
2. Ngôn ngữ tân từ có biến là miền giá trị.....	58
3. Bài tập.....	59

## **Chương 6: Ngôn ngữ truy vấn SQL ..... 61**

1. Các lệnh hỏi.....	61
1.1. Cú pháp lệnh truy vấn.....	61
1.2. Phép chiếu.....	62
1.3. Phép chọn .....	62
1.4. Phép kết .....	63

1.5. Một số lưu ý.....	63
2. Truy vấn lồng .....	65
3. Hàm kết hợp và gom nhóm .....	68
4. Các lệnh khai báo cấu trúc CSDL .....	69
5. Các thao tác cập nhật dữ liệu.....	72
6. Bài tập.....	73
<b>Chương 7: Phụ thuộc hàm, khóa, ràng buộc toàn vẹn .....</b>	<b>74</b>
1. Phụ thuộc hàm .....	74
1.1. Khái niệm phụ thuộc hàm.....	74
1.2. Hệ luật dẫn Amstrong.....	75
1.3. Thuật toán tìm bao đóng của tập thuộc tính .....	77
1.4. Bài toán thành viên.....	78
1.5. Phủ tối thiểu của một tập phụ thuộc hàm .....	78
2. Khóa .....	82
2.1. Định nghĩa .....	82
2.2. Thuật toán tìm khóa.....	82
3. Ràng buộc toàn vẹn .....	85
3.1. Định nghĩa – các yếu tố của ràng buộc toàn vẹn.....	85
3.2. Các loại ràng buộc toàn vẹn .....	87
4. Bài tập.....	93
<b>Chương 8: Dạng chuẩn và chuẩn hóa CSDL .....</b>	<b>96</b>
1. Dạng chuẩn của lược đồ quan hệ .....	96
1.1. Dạng chuẩn 1 .....	96
1.2. Dạng chuẩn 2 .....	97
1.3. Dạng chuẩn 3 .....	100

1.4.	Dạng chuẩn BC .....	101
1.5.	Kiểm tra dạng chuẩn .....	101
2.	Phép phân rã .....	102
2.1.	Phân rã bảo toàn thông tin .....	102
2.2.	Phân rã bảo toàn phụ thuộc hàm.....	103
3.	Thiết kế CSDL bằng cách phân rã .....	104
3.1.	Phân rã thành dạng chuẩn BC (hoặc dạng chuẩn 3) bảo toàn thông tin.....	104
3.2.	Phân rã thành dạng chuẩn 3 vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm .....	108
4.	Bài tập.....	108
	<i>Tài liệu tham khảo .....</i>	<i>109</i>

## Chương 1

# Giới Thiệu Chung

---

Chương 1 giới thiệu cho học viên các khái niệm cơ bản về cơ sở dữ liệu (CSDL) và hệ quản trị CSDL.

## 1. Giới thiệu chung

### 1.1. Giới thiệu

Xét một hệ thống các tập tin cổ điển: được tổ chức một cách riêng rẽ, phục vụ cho một mục đích của một đơn vị hay một đơn vị con trực thuộc cụ thể. Ứng dụng là một hệ quản lý nhân sự của một công ty.

Tại một công ty đã được trang bị máy vi tính cho tất cả các phòng ban. Trong đó:

- Phòng Tổ chức lưu giữ những thông tin về lý lịch của nhân viên như họ tên, bí danh, giới tính, ngày sinh, ngày vào đoàn, ngày vào đảng, ngày tuyển dụng, hệ số lương, hệ số phụ cấp, hoàn cảnh gia đình ... Phần mềm được sử dụng để quản lý là Microsoft Access.
- Phòng Tài vụ lưu giữ những thông tin về họ tên, hệ số lương, hệ số phụ cấp, đơn vị ... để tính toán lương và phụ cấp dựa trên danh sách do phòng tổ chức gửi sang. Các bảng tính này được làm bởi Microsoft Excel.
- Bộ phận Văn thư sử dụng Microsoft word để thảo các văn bản báo cáo về tình hình của đơn vị trong đó có nêu tổng số công nhân viên và trình độ chuyên môn được đào tạo.
- Trong khi đó tại tổng công ty các phòng ban cũng được trang bị máy vi tính và cũng quản lý những thông tin của các cán bộ chủ chốt của công ty con là giám đốc, phó giám đốc, trưởng phó phòng.

Việc phát triển của hệ thống quản lý như sau:

- Những chương trình ứng dụng mới phải được viết khi một nhu cầu nảy sinh.
- Các tập tin thường trực mới được tạo ra theo yêu cầu.
- Trong một thời gian dài các tập tin có thể ở các dạng thức khác nhau.
- Những chương trình ứng dụng có thể viết bởi các ngôn ngữ khác nhau.



### **Ưu điểm:**

- Việc xây dựng những hệ thống tập tin riêng lẻ tại từng đơn vị quản lý ít tốn thời gian bởi lượng thông tin cần quản lý và khai thác là nhỏ, không đòi hỏi đầu tư nhiều nên triển khai nhanh.
- Thông tin được khai thác chỉ phục vụ cho mục đích hẹp nên khả năng đáp ứng nhanh chóng, kịp thời.

### **Hạn chế:**

Do thông tin được tổ chức ở mỗi phòng ban mỗi khác (thông tin và công cụ triển khai) nên rõ ràng sự phối hợp về mặt tổ chức và khai thác ở các phòng ban rất khó khăn. Thông tin ở phòng ban này không được sử dụng cho phòng ban khác, tại công ty con với tổng công ty. Hơn nữa cùng một thông tin được cập nhật vào máy tại nhiều nơi khác nhau. Do đó có những hạn chế sau:

- **Dữ liệu dư thừa và không nhất quán**
  - ✓ Những thông tin giống nhau có thể bị trùng lặp ở một số nơi.
  - ✓ Tất cả những bản sao dữ liệu có thể không được cập nhật đúng.
- **Khó khăn trong việc truy xuất dữ liệu**
  - ✓ Có thể phải viết một trình ứng dụng mới thỏa yêu cầu mới đặt ra.
  - ✓ Có thể tạo ra dữ liệu này một cách thủ công nhưng mất thời gian.
- **Khó khăn khi viết chương trình ứng dụng đòi hỏi thông tin liên quan đến nhiều đơn vị**
  - ✓ Dữ liệu ở những tập tin khác nhau.
  - ✓ Dữ liệu ở các dạng thức khác nhau.

Ngoài ra còn một số vấn đề chưa được giải quyết như:

- **Nhiều người sử dụng**
  - ✓ Muốn thời gian trả lời nhanh hơn cho các tác vụ đồng thời.
  - ✓ Cần sự bảo vệ đối với những cập nhật đồng thời.
- **Các vấn đề bảo mật**
  - ✓ Mỗi người sử dụng hệ thống chỉ có thể truy xuất những dữ liệu mà họ được phép thấy.

- ✓ Chẳng hạn như những người tính lương và phụ cấp chỉ giải quyết các mẫu tin nhân viên, và không thể thấy được thông tin về lý lịch nhân viên; những thủ quỹ chỉ truy xuất những dữ liệu về các khoản chi...
- ✓ Khó khăn đề ràng buộc điều này trong những chương trình ứng dụng.
- **Các vấn đề toàn vẹn.**
  - ✓ Dữ liệu có thể được yêu cầu thỏa mãn những ràng buộc.
  - ✓ Chẳng hạn như nhân viên đang hưởng chế độ nào đó ở phòng tổ chức thì sẽ không được hưởng chế độ trợ cấp cao nhất của phòng công đoàn.
  - ✓ Với cách tiếp cận xử lý-tập tin, khó khăn để thực hiện hoặc thay đổi những ràng buộc như vậy.
- **Khó khăn khi nâng cấp hệ thống.**
  - ✓ Do hệ thống được tổ chức thành các tập tin riêng lẻ nên thiếu sự chia sẻ thông tin giữa các phòng ban. Bên cạnh đó việc kết nối các hệ thống này với nhau hay nâng cấp ứng dụng sẽ gặp rất nhiều khó khăn.

Những nhược điểm này đã dẫn tới việc phát triển các hệ thống quản trị cơ sở dữ liệu.

## 1.2. Định nghĩa cơ sở dữ liệu

Cơ sở dữ liệu (database) là một hệ thống các thông tin có cấu trúc được lưu trữ trên các thiết bị lưu trữ thứ cấp (băng từ, đĩa từ...) nhằm thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng với nhiều mục đích khác nhau.

### Ưu điểm của CSDL:

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

Để đạt được những ưu điểm trên CSDL đặt ra những vấn đề cần giải quyết, đó là:

- **Tính chủ quyền của dữ liệu.** Do sự chia sẻ của CSDL nên tính chủ quyền của dữ liệu có thể bị lu mờ và làm mờ nhạt tinh thần trách nhiệm, được thể hiện trên vấn đề an toàn dữ liệu, khả năng biểu diễn các mối liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu. Nghĩa là người khai thác CSDL phải có nghĩa vụ cập nhật các thông tin mới nhất của CSDL.

- **Tính bảo mật và quyền khai thác thông tin của người sử dụng (NSD).** Do có nhiều người cùng khai thác CSDL một cách đồng nên cần phải có một cơ chế bảo mật và phân quyền khai thác CSDL.
- **Tranh chấp dữ liệu.** Nhiều người được phép truy cập vào cùng một tài nguyên dữ liệu (data source) của CSDL với những mục đích khác nhau như xem, thêm, xóa, sửa dữ liệu. Như vậy cần phải có cơ chế ưu tiên truy nhập dữ liệu cũng như cơ chế giải quyết tình trạng khoá chết (DeadLock) trong quá trình khai thác cạnh tranh. Cơ chế ưu tiên có thể được thực hiện dựa trên:
  - ✓ Cấp quyền ưu tiên cho người khai thác CSDL, người nào có quyền ưu tiên cao hơn thì được ưu tiên cấp quyền truy nhập dữ liệu trước.
  - ✓ Thời điểm truy nhập, ai có yêu cầu truy xuất trước thì có quyền truy nhập dữ liệu trước.
  - ✓ Hoặc dựa trên cơ chế lập lịch truy xuất hay cơ chế khoá.
  - ✓ ...
- **Đảm bảo dữ liệu khi có sự cố.** Việc quản lý tập trung có thể làm tăng khả năng mất mát hoặc sai lệch thông tin khi có sự cố như mất điện đột xuất, một phần hay toàn bộ đĩa lưu trữ dữ liệu bị hư... Một số hệ điều hành mạng có cung cấp dịch vụ sao lưu đĩa cứng (cơ chế sử dụng đĩa cứng dự phòng - RAID), tự động kiểm tra và khắc phục lỗi khi có sự cố. Tuy nhiên bên cạnh dịch vụ của hệ điều hành, một CSDL nhất thiết phải có một cơ chế khôi phục dữ liệu khi các sự cố bất ngờ xảy ra để đảm bảo CSDL luôn ổn định.

### 1.3. Các đối tượng sử dụng CSDL

- Những người sử dụng không chuyên về lĩnh vực tin học và CSDL, do đó CSDL cần có các công cụ để giúp cho người sử dụng không chuyên có thể khai thác hiệu quả CSDL
- Các chuyên viên tin học biết khai thác CSD, đây là những người có thể xây dựng các ứng dụng khác nhau phục vụ cho những yêu cầu khác nhau trên CSDL.
- Những người quản trị CSDL, là những người hiểu biết về tin học, các hệ quản trị CSDL và hệ thống máy tính. Họ là người tổ chức CSDL (khai báo cấu trúc CSDL, ghi nhận các yêu cầu bảo mật cho các dữ liệu cần bảo vệ...), do đó họ cần phải nắm rõ các vấn đề về kỹ thuật để có thể phục hồi dữ liệu khi có sự cố. Họ là những người cấp quyền hạn khai thác CSDL nên họ có thể giải quyết các vấn đề tranh chấp dữ liệu.

#### 1.4. Hệ quản trị CSDL

Để giải quyết tốt những vấn đề đặt ra cho một CSDL như đã trình bày trong 1.2 thì cần có một hệ thống các phần mềm chuyên dụng. Hệ thống này được gọi là hệ quản trị CSDL (Database Management System - DBMS), là công cụ hỗ trợ tích cực cho các nhà phân tích, thiết kế và khai thác CSDL. Hiện nay có nhiều hệ quản trị CSDL trên thị trường như: Visual Foxpro, SQL server, DB2, Microsoft Access, Oracle...

Một hệ quản trị CSDL phải có:

- Ngôn ngữ giao tiếp giữa NSD và CSDL:
  - ✓ Ngôn ngữ mô tả dữ liệu (Data Definition Language - DDL) để cho phép khai báo cấu trúc của CSDL, khai báo các mối liên hệ của dữ liệu (Data Relationship) và các quy tắc (Rules, Constraint) quản lý áp đặt trên dữ liệu.
  - ✓ Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML) cho phép NSD có thể thêm, xóa, dữ liệu trong CSDL.
  - ✓ Ngôn ngữ truy vấn dữ liệu (Structured Query Language - SQL) cho phép NSD truy vấn các thông tin cần thiết.
  - ✓ Ngôn ngữ quản lý dữ liệu (Data Control Language - DCL) cho phép những người quản trị hệ thống thay đổi cấu trúc của các bảng dữ liệu, khai báo bảo mật thông tin và cấp quyền khai thác CSDL cho NSD.
- Tự điển dữ liệu (Data dictionary) dùng để mô tả các ánh xạ liên kết, ghi nhận các thành phần cấu trúc của CSDL, các chương trình ứng dụng, mật mã, quyền sử dụng...
- Có biện pháp bảo mật tốt.
- Có cơ chế giải quyết tranh chấp dữ liệu.
- Có cơ chế sao lưu và phục hồi dữ liệu khi có sự cố xảy ra.
- Có giao diện tốt, dễ sử dụng.
- Bảo đảm tính độc lập giữa dữ liệu và chương trình: khi có sự thay đổi dữ liệu (như sửa đổi cấu trúc lưu trữ các bảng dữ liệu, thêm các chỉ mục,...) thì các chương trình ứng dụng đang chạy trên CSDL vẫn không cần phải viết lại hay cũng không ảnh hưởng đến NSD khác.

#### 1.5. Các mức biểu diễn một CSDL

Để hệ thống có thể dùng được thì phải tìm dữ liệu một cách hiệu quả. Điều này dẫn đến việc cần phải thiết kế những cấu trúc dữ liệu phức tạp đối với việc biểu diễn dữ liệu trong

cơ sở dữ liệu. Vì nhiều người sử dụng các hệ thống cơ sở dữ liệu không phải là người hiểu biết nhiều về máy tính nên những người phát triển hệ thống phải che dấu đi sự phức tạp khỏi người sử dụng thông qua một số mức trừu tượng, nhằm làm đơn giản hóa sự tương tác của người sử dụng đối với hệ thống.

Có 3 mức biểu diễn dữ liệu, cụ thể như sau:

#### **Mức trong (mức vật lý - Physical level)**

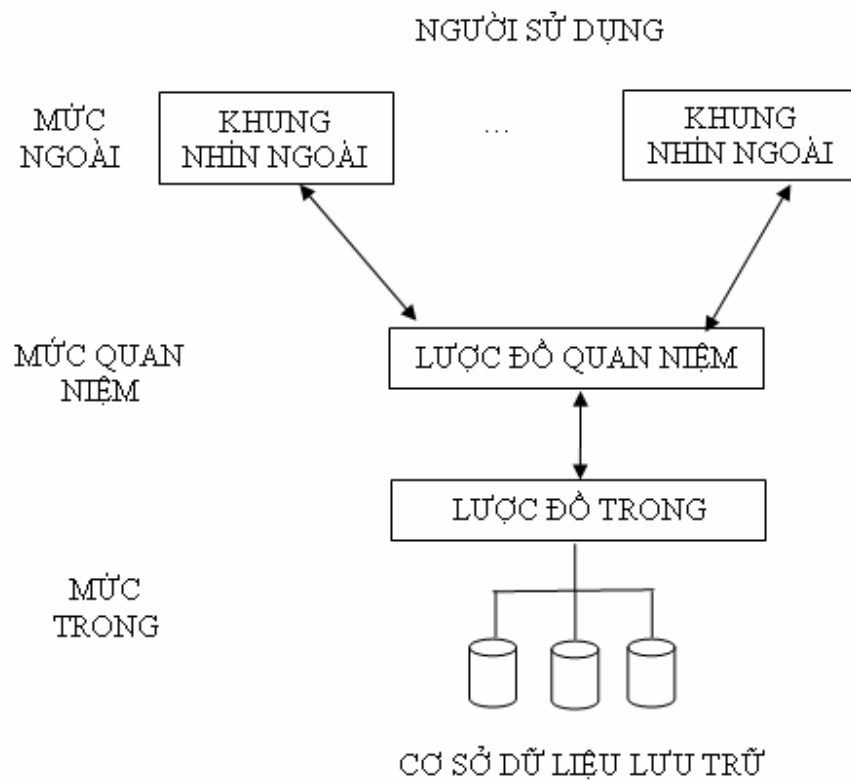
- Nói đến cách thức dữ liệu được lưu trữ thực sự. Trả lời câu hỏi dữ liệu gì và lưu trữ chúng như thế nào? Cần các chỉ mục gì?
- Ví dụ như chỉ mục, B-cây, băm.
- Những người thao tác tại mức này là người quản trị CSDL và những NSD chuyên môn.

#### **Mức quan niệm hay mức logic (conception level, logical level)**

- Trả lời câu hỏi cần phải lưu trữ bao nhiêu loại dữ liệu? Đó là những dữ liệu gì? Mối quan hệ giữa chúng như thế nào?
- Như vậy CSDL mức vật lý là sự cài đặt cụ thể của CSDL mức quan niệm.

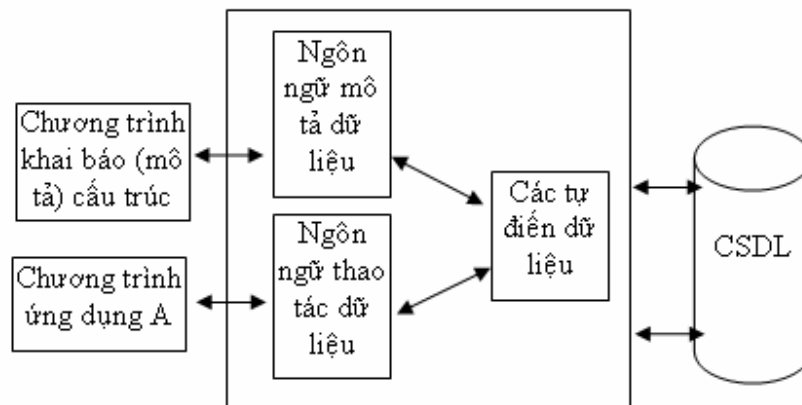
#### **Mức ngoài hay mức nhìn (view level)**

- Là mức của NSD và các chương trình ứng dụng.
- Làm việc tại mức này là các nhà chuyên môn, kỹ sư tin học, NSD không chuyên.
- Mỗi NSD hay mỗi chương trình ứng dụng có thể được nhìn CSDL theo một góc độ khác nhau. Tùy thuộc vào vai trò của NSD mà có thể thấy tất cả hoặc một phần của CSDL. NSD hay chương trình ứng dụng có thể không được biết về cấu trúc tổ chức lưu trữ thông tin trong CSDL. Họ chỉ có thể làm việc trên một phần CSDL theo cách nhìn do người quản trị hay chương trình ứng dụng quy định, gọi là khung nhìn.



**Hình 1.1. Ba mức trừu tượng dữ liệu**

## 1.6. Sơ đồ tổng quát một hệ quản trị CSDL



**Hình 1.2. Sơ đồ tổng quát của một hệ quản trị CSDL**

Hình 1.2 minh họa sơ đồ tổng quát của một hệ quản trị CSDL. Có 3 mức: mức chương trình khai báo cấu trúc và chương trình ứng dụng; mức mô tả CSDL, thao tác CSDL và các tự điển dữ liệu; mức CSDL.

Mỗi hệ quản trị CSDL có một ngôn ngữ khai báo hay mô tả (Data Definition Language - DDL) cấu trúc CSDL riêng. Những người thiết kế và quản trị CSDL thực hiện các công việc khai báo cấu trúc CSDL.

Các chương trình khai báo cấu trúc CSDL được viết bằng ngôn ngữ mà hệ quản trị CSDL cho phép. Hai công việc khai báo cấu trúc logic (là việc khai báo các loại dữ liệu và các mối liên hệ giữa các loại dữ liệu đó, cùng các ràng buộc toàn vẹn dữ liệu) và khai báo vật lý (dữ liệu được lưu trữ theo dạng nào, có bao nhiêu chỉ mục)

Các chương trình ứng dụng được viết bằng ngôn ngữ thao tác CSDL (Data Manipulation Language - DML) với mục đích:

- Truy xuất dữ liệu
- Cập nhật dữ liệu (thêm, xóa, sửa)
- Khai thác dữ liệu
- Ngôn ngữ thao tác CSDL còn được sử dụng cho những người sử dụng thao tác trực tiếp với CSDL.

Tự điển dữ liệu (Data Dictionary - DD) là một CSDL của hệ quản trị CSDL sử dụng để lưu trữ cấu trúc CSDL, các thông tin bảo mật, bảo đảm an toàn dữ liệu và các cấu trúc ngoài. Tự điển dữ liệu còn được gọi là siêu CSDL (Meta-Database)

### **1.7. Tính độc lập giữa dữ liệu và chương trình**

- **Độc lập dữ liệu vật lý**
  - ✓ Khả năng cập nhật lược đồ vật lý không làm cho các chương trình ứng dụng phải bị viết lại.
  - ✓ Việc cập nhật ở mức này thường cần thiết để cải tiến hiệu suất.
- **Độc lập dữ liệu logic**
  - ✓ Khả năng cập nhật lược đồ logic mà không làm cho các chương trình ứng dụng phải bị viết lại.
  - ✓ Những cập nhật thường được thực hiện khi cấu trúc logic của cơ sở dữ liệu thay đổi.

## **2. Các cách tiếp cận của một CSDL**

Nền tảng của cấu trúc cơ sở dữ liệu là mô hình dữ liệu. Mô hình dữ liệu được định nghĩa là một sưu tập các công cụ khái niệm dùng cho việc mô tả dữ liệu, các mối quan hệ dữ liệu, các ngữ nghĩa dữ liệu và các ràng buộc dữ liệu.

## 2.1. Cách tiếp cận theo mô hình dữ liệu mạng

Mô hình dữ liệu mạng (Network Data Model) hay gọi tắt là mô hình mạng (Network Model) là mô hình được biểu diễn bởi một đồ thị có hướng. Mô hình này được xây dựng bởi Honeywell vào năm 1964-1965. Trong mô hình này, có 3 khái niệm được sử dụng: mẫu tin hay bản ghi (record), loại mẫu tin (record type) và loại liên hệ (set type).

**Loại mẫu tin:** đặc trưng cho một loại đối tượng riêng biệt. Chẳng hạn như trong quản lý nhân sự tại một đơn vị thì đối tượng cần phản ánh của thế giới thực có thể là Phòng, nhân viên, công việc, lý lịch... do đó có các loại mẫu tin đặc trưng cho từng đối tượng này. Trong mô hình mạng, mỗi loại mẫu tin được thể hiện bằng một hình chữ nhật, một thể hiện (instance) của một loại mẫu tin được gọi là bản ghi (record). Ở ví dụ trên loại mẫu tin Phòng có các mẫu tin là các phòng ban trong công ty, loại mẫu tin Nhân viên có các mẫu tin là các nhân viên đang làm việc tại các phòng ban của công ty...

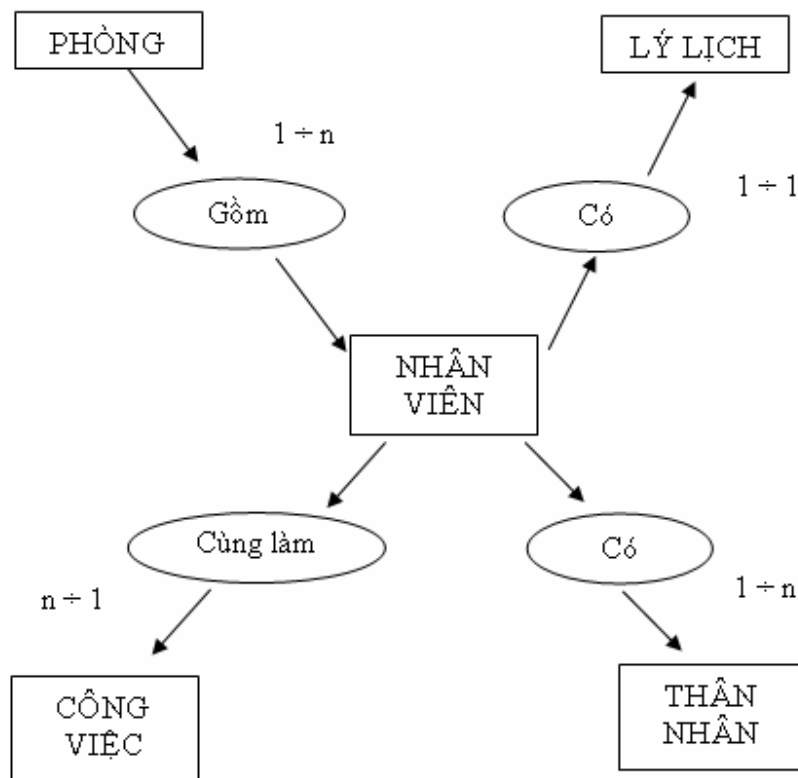
**Loại liên hệ:** là sự liên kết giữa một loại mẫu tin chủ với một mẫu tin thành viên. Trong mô hình mạng, một loại liên hệ được thể hiện bằng hình bầu dục và sự liên kết giữa hai loại mẫu tin được thể hiện bằng các cung có hướng đi từ loại mẫu tin chủ đến loại liên hệ và từ loại liên hệ đến loại mẫu tin thành viên.

Trong loại liên kết người ta còn chỉ ra số lượng các mẫu tin tham gia vào trong mỗi kết hợp. Có một số loại sau:

- $1 \div 1$  (One to One): Mỗi mẫu tin của loại mẫu tin chủ kết hợp với đúng 1 mẫu tin của loại mẫu tin thành viên. Ví dụ, mỗi nhân viên có duy nhất một lý lịch cá nhân.
- $1 \div n$  (One to Many): Mỗi mẫu tin của loại mẫu tin chủ kết hợp với 1 hay nhiều mẫu tin của loại mẫu tin thành viên. Ví dụ, mỗi phòng ban có thể có một hoặc nhiều nhân viên.
- $n \div 1$  (Many to One): Nhiều mẫu tin của loại mẫu tin chủ kết hợp với đúng 1 mẫu tin của loại mẫu tin thành viên. Ví dụ, nhiều nhân viên cùng làm một công việc.
- **Đệ quy (Recursive):** Một loại mẫu tin chủ cũng có thể đồng thời là mẫu tin thành viên với chính nó. Ví dụ, trưởng phòng cũng là một nhân viên thuộc phòng trong công ty.

Hình 1.4 minh họa một mô hình dữ liệu mạng.





**Hình 1.4. Mô hình dữ liệu mạng**

**Ưu điểm:** đơn giản, dễ sử dụng.

**Hạn chế:** không thích hợp trong việc biểu diễn CSDL lớn do hạn chế về khả năng diễn đạt ngữ nghĩa của dữ liệu, đặc biệt là các dữ liệu và mối liên hệ phức tạp của dữ liệu trong thực tế là rất hạn chế.

## 2.2. Mô hình dữ liệu phân cấp

Mô hình dữ liệu phân cấp (Hierarchical Data Model), còn gọi là mô hình phân cấp (Hierarchical Model), được thực hiện thông qua sự kết hợp giữa IBM và North American Rockwell vào khoảng năm 1965. Mô hình là một cây, trong đó mỗi nút của cây biểu diễn một thực thể, giữa nút con với nút cha được liên hệ với nhau theo một mối quan hệ xác định.

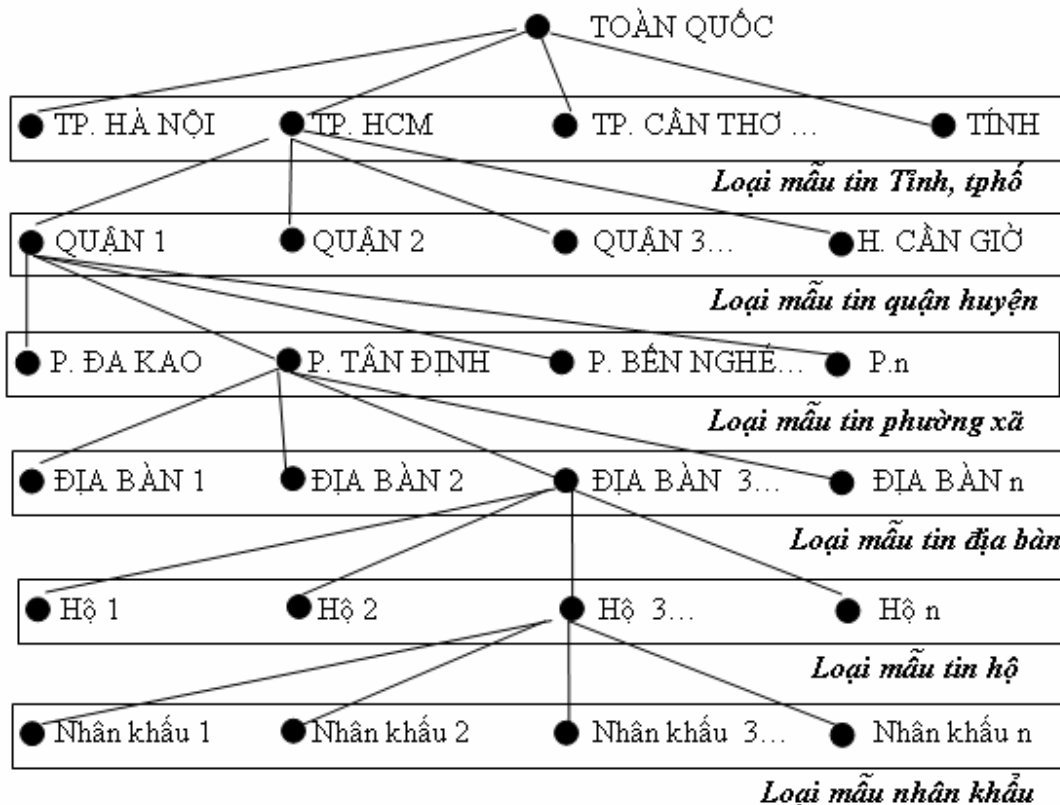
**Loại mẫu tin:** giống trong mô hình dữ liệu mạng

**Loại mối liên hệ:** kiểu liên hệ là phân cấp:

Mẫu tin thành viên chỉ đóng vai trò thành viên của một mối liên hệ duy nhất, tức là nó thuộc một chủ duy nhất. Như vậy, mối liên hệ từ mẫu tin chủ tới các mẫu tin thành viên là  $1 \div n$  và từ mẫu tin thành viên với mẫu tin chủ là  $1 \div 1$

Giữa hai loại mẫu tin chỉ tồn tại một mối quan hệ duy nhất.

Hình 1.5 minh họa mô hình dữ liệu phân cấp một CSDL về điều tra dân số.



**Hình 1.5. Mô hình dữ liệu phân cấp**

### 2.3. Mô hình dữ liệu quan hệ

Mô hình dữ liệu quan hệ (Relational Data Model) còn được gọi là mô hình quan hệ (Relational Model) do E.F.Codd đề xuất năm 1970. Nền tảng cơ bản là khái niệm lý thuyết tập hợp trên các quan hệ, tức là tập của các bộ giá trị (value tuples). Trong mô hình dữ liệu này những khái niệm được sử dụng là thuộc tính (attribute), quan hệ (relation), lược đồ quan hệ (relation schema), bộ (tuple), khóa (key).

Mô hình quan hệ là mô hình được nghiên cứu nhiều nhất, và có cơ sở lý thuyết vững chắc nhất. Mô hình quan hệ cùng với mô hình dữ liệu thực thể kết hợp đang được sử dụng rộng rãi nhất hiện nay trong việc phân tích và thiết kế CSDL. Chúng ta sẽ nghiên cứu chi tiết về mô hình này trong chương sau.

### 2.4. Mô hình dữ liệu thực thể - kết hợp

Mô hình dữ liệu thực thể - kết hợp (Entity – Relationship Model) do Peter Pin-Shan Chen đề xuất năm 1976. Mô hình quan hệ-thực thể được dựa trên sự nhận thức thế giới gồm có

một sưu tập các đối tượng căn bản, được gọi là các tập thực thể, và các mối quan hệ ở giữa các đối tượng này. Trong mô hình này các khái niệm được sử dụng là tập thực thể (entity set), thực thể (entity), thuộc tính của loại thực thể (entity attribute), khóa của loại thực thể (entity key), loại mối kết hợp (entity relationship), số ngôi của mối kết hợp (relationship degree), thuộc tính của mối kết hợp (relationship attribute), bản số của mối kết hợp (relationship cardinal).

Chi tiết của mô hình này sẽ được nghiên cứu trong chương sau.

## **2.5. Mô hình dữ liệu hướng đối tượng**

Mô hình dữ liệu hướng đối tượng (Object Oriented Data Model) ra đời vào cuối những năm 80 và đầu những năm 90. Đây là loại mô hình tiên tiến nhất hiện nay dựa trên cách tiếp cận hướng đối tượng. Mô hình này sử dụng các khái niệm như lớp (class), sự kế thừa (inheritance), kế thừa bội (multi - inheritance). Đặc trưng cơ bản của cách tiếp cận này là tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính tái sử dụng (reusability).

## **3. Bài tập**

### **Bài 1:**

Dựa vào những khái niệm đã học, hãy biểu diễn CSDL có các loại mẫu tin Phòng, Nhân viên, Công việc, lý lịch đã trình bày trong mô hình mạng theo cách tiếp cận phân cấp. Biết:

- Loại liên hệ là phân cấp.
- Phòng có nhiều nhân viên, mỗi nhân viên chỉ thuộc một phòng duy nhất
- Công việc có nhiều nhân viên cùng làm; mỗi nhân viên chỉ làm một công việc duy nhất
- Mỗi nhân viên có một lý lịch, mỗi lý lịch chỉ thuộc duy nhất một nhân viên.

### **Bài 2:**

Dựa vào những khái niệm đã học, hãy biểu diễn CSDL về tổng điều tra dân số toàn quốc có các loại mẫu tin tỉnh – thành phố, quận huyện, phường xã, địa bàn, hộ điều tra và nhân khẩu đã trình bày trong mô hình phân cấp theo cách tiếp cận mạng. Biết:

- Nhân khẩu thuộc một hộ điều tra
- Hộ điều tra thuộc một địa bàn
- Địa bàn điều tra thuộc một phường xã

- Phường xã thuộc một quận huyện
  - Quận huyện thuộc một tỉnh, thành phố
-

## Chương 2

# Mô Hình Thực Thể Kết Hợp

---

Mô hình dữ liệu thực thể kết hợp (E-R - entity-relationship data model) do Peter Pin-Shan Chen đề xuất năm 1976, nhìn thế giới thực như là một tập các đối tượng căn bản được gọi là các *thực thể*, và *các mối quan hệ* ở giữa các đối tượng này. Mô hình đã được phát triển để làm thuận tiện cho việc thiết kế cơ sở dữ liệu bằng cách đặc tả một tổ chức. Một lược đồ như vậy biểu diễn một cấu trúc logic tổng quát của cơ sở dữ liệu.

### 1. Mô hình thực thể kết hợp

Những khái niệm căn bản mà mô hình thực thể kết hợp dùng: thực thể, tập thực thể, khóa, mối quan hệ, và thuộc tính.

#### 1.1. Thực thể - Tập thực thể

Một *thực thể* (an entity) là một “sự vật” hoặc “đối tượng” mà nó tồn tại và có thể phân biệt được với các đối tượng khác. Ví dụ như một nhân viên trong một tổ chức là một thực thể.

Một *tập thực thể* (an entity set) là một tập hợp các thực thể cùng loại mà chúng chia sẻ cùng những tính chất hoặc thuộc tính. Ví dụ như tập hợp tất cả những người mà họ là nhân viên của một tổ chức là một tập thực thể *khách hàng*.

Mỗi tập thực thể được đặt một tên gọi, thông thường là danh từ. Ví dụ như KHACHHANG, HOADON,...

Ký hiệu: hình chữ nhật với tên gọi

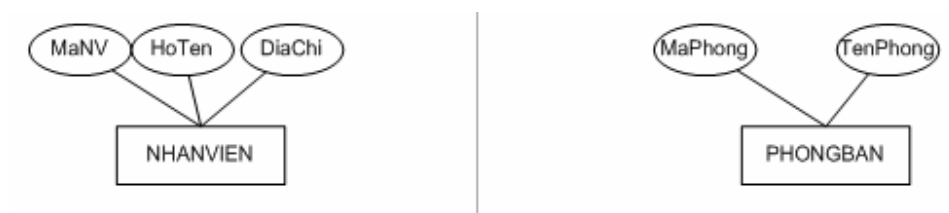


**Hình 2.1. Tập thực thể**

#### 1.2. Thuộc tính

Mỗi một tập thực thể có nhiều đặc trưng riêng được gọi là các thuộc tính. Mỗi một thuộc tính được đặt một tên, chẳng hạn như MaKH (mã khách hàng), HoTenKH (họ tên khách hàng)...

Ký hiệu: hình oval với tên gọi, có đường nối với thực thể



**Hình 2.2. Thuộc tính của tập thực thể**

Khi phân tích một thuộc tính, thông thường cần xét đến kiểu dữ liệu và miền giá trị tương ứng của thuộc tính đó. Chẳng hạn các thuộc tính Mã nhân viên (MANV), Họ tên (HoTen), Địa chỉ (DiaChi) là những chuỗi ký tự phản ánh những thông tin trên về nhân viên; Đơn giá là số nguyên dương,...

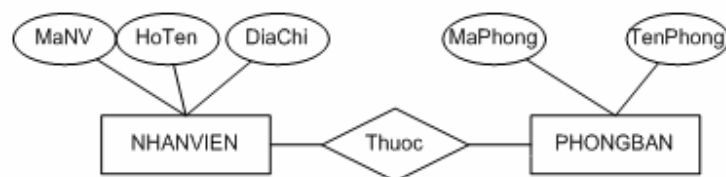
Có một số loại thuộc tính sau:

- **Thuộc tính đơn, kết hợp:** Thuộc tính đơn là thuộc tính không thể chia ra thành những phần con nhỏ hơn. Còn thuộc tính kết hợp thì có thể chia ra thành những phần con (tức là thành những thuộc tính khác). Ví dụ *tên nhân viên* có thể cấu trúc như là thuộc tính kết hợp bao gồm *tên*, *chữ lót*, và *họ*. Thuộc tính kết hợp có thể xuất hiện theo sự phân cấp. Ví dụ thuộc tính *địa chỉ nhân viên* có các thuộc tính thành phần là *số nhà*, *đường*, *thành phố*, *tỉnh*.
- **Thuộc tính rỗng.** Một giá trị rỗng (null value) được dùng đến khi một thực thể không có giá trị đối với một thuộc tính. Ví dụ một nhân viên nào đó không có người trong gia đình thì giá trị của thuộc tính *tên người trong gia đình* đối với nhân viên đó phải là rỗng. Giá trị rỗng cũng có thể được dùng để chỉ ra rằng giá trị của thuộc tính là chưa biết. Một giá trị chưa biết có thể hoặc là giá trị tồn tại nhưng chúng ta không có thông tin đó, hoặc là không biết được (không biết giá trị thực sự có tồn tại hay không).
- **Thuộc tính suy ra.** Giá trị của loại thuộc tính này có thể được suy ra từ các thuộc tính hoặc thực thể liên hệ khác. Ví dụ tập thực thể *khách hàng* có thuộc tính *số lượng tài khoản vay* cho biết bao nhiêu tài khoản vay mà khách hàng có từ ngân hàng. Chúng ta có thể suy ra giá trị của thuộc tính này bằng cách đếm số lượng các thực thể *tài khoản vay* được kết hợp với khách hàng. Một ví dụ khác về tập thực thể *nhân viên* có các thuộc tính *ngày bắt đầu*, cho biết ngày bắt đầu làm việc tại ngân hàng của nhân viên, và thuộc tính *thời gian làm việc*, cho biết tổng số giờ đã làm việc của nhân viên. Giá trị của *thời gian làm việc* có thể được suy ra từ giá trị của *ngày bắt đầu* và ngày hiện thời.

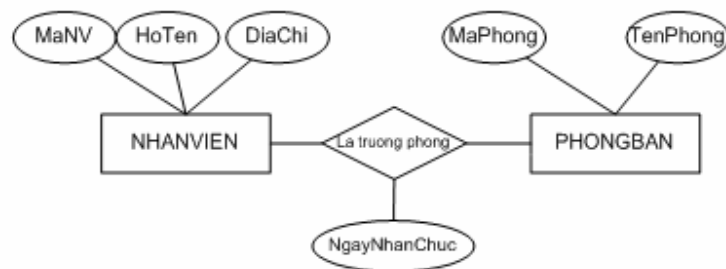
### 1.3. Môi kết hợp

Quan hệ giữa các tập thực thể được gọi là môi kết hợp. Mỗi môi kết hợp cũng được đặt tên thể hiện môi kết hợp, chẳng hạn như mỗi hóa đơn chỉ bán cho một khách hàng, một khách hàng có thể mua hàng nhiều lần (nhiều hóa đơn)

Ký hiệu là hình thoi nối với những thực thể tham gia vào mối kết hợp



**Hình 2.3. a. Mối kết hợp không có thuộc tính**



**Hình 2.3.b. Mối kết hợp có thuộc tính**

## 1.4. Bản số

### 1.4.1. Thể hiện của một thực thể

Trong tập thực thể KHACHHANG có nhiều khách hàng, ví dụ như {KH01, Nguyễn Trọng, Vũ, 11 Nguyễn Công Trứ, 063552540} là một khách hàng, đây được gọi là một thể hiện của thực thể KHACHHANG.

Một thể hiện của một tập thực thể được nhận biết bằng tập hợp tất cả các giá trị của tất cả các thuộc tính của thể hiện thuộc tập thực thể, đó chính là dữ liệu của đối tượng trong thế giới thực.

### 1.4.2. Thể hiện của một mối kết hợp

Xét mối kết hợp *bán* giữa HÓA ĐƠN và MẶT HÀNG, trong đó một thể hiện của quan hệ này sẽ tương ứng với việc một hóa đơn bán mặt hàng với số lượng và đơn giá là bao nhiêu.

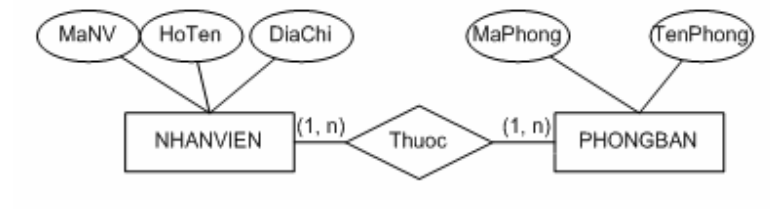
Như vậy, một thể hiện của một mối kết hợp là tập hợp các thể hiện của các tập thực thể tham gia vào mối kết hợp đó.

#### 1.4.2.1. Bản số

Bản số của một tập thực thể đối với một mối kết hợp là cặp (bản số tối thiểu, bản số tối đa). Trong đó chúng được định nghĩa như sau:

- Bản số tối thiểu: bằng 0 hoặc 1, là số lần tối thiểu mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mỗi kết hợp.
- Bản số tối đa: bằng 1 hoặc n, là số lần tối đa mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mỗi kết hợp.

Ký hiệu:



**Hình 2.4. Bản số**

Ví dụ một nhân viên thuộc về một hoặc nhiều phòng ban, một phòng ban có thể có một hoặc nhiều nhân viên trực thuộc.

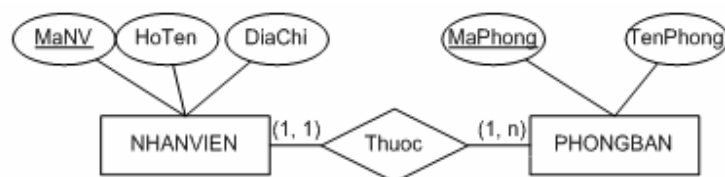
## 1.5. Khóa

### 1.5.1. Khóa của một tập thực thể

Khóa của một tập thực thể là một thuộc tính hoặc một số thuộc tính của thực thể, sao cho với mỗi giá trị của các thuộc tính này, tương ứng một và chỉ một thể hiện của tập thực thể (xác định một thực thể duy nhất). Trong nhiều trường hợp khóa của tập thực thể thường là thuộc tính chỉ định của tập thực thể đó.

Một thực thể có thể có nhiều khóa. Khi đó cần chọn ra một khóa để làm khóa chính.

Ký hiệu: Là thuộc tính được gạch dưới.



**Hình 2.5. Khóa**

### 1.5.2. Khóa của một mối kết hợp

Khóa của một mối kết hợp nhận được bằng cách kết hợp khóa của các tập thực thể tham gia vào mối kết hợp đó. Tập hợp tất cả các giá trị của các thuộc tính khóa của một mối kết hợp xác định duy nhất một thể hiện của mối kết hợp đó.

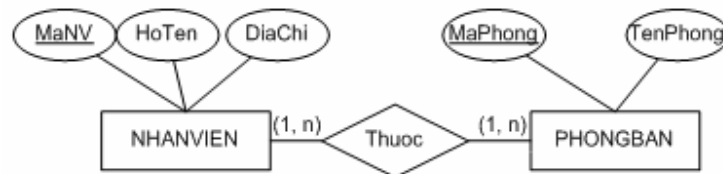


Trong mô hình, khóa của mỗi kết hợp ngầm hiểu mà không được ghi ra nếu không quan tâm.

## 1.6. Số chiều (bậc) của một mối kết hợp, mối kết hợp tự thân (đệ quy)

### 1.6.1. Số chiều của một mối kết hợp

Số chiều của một mối kết hợp là số tập thực thể tham gia vào mối kết hợp đó.

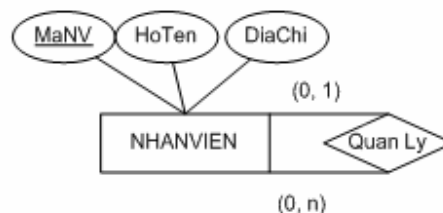


**Hình 2.6. Mối kết hợp nhị phân**

### 1.6.2. Mối kết hợp tự thân (đệ quy, vai trò)

Mối kết hợp tự thân là một mối kết hợp từ một tập thực thể đi đến chính tập thực thể đó.

Ví dụ một nhân viên có thể không chịu sự quản lý của ai hoặc một người, một người có thể không quản lý ai hoặc nhiều người.



**Hình 2.7. Mối kết hợp đệ quy**

## 1.7. Tổng quát hóa và chuyên biệt hóa

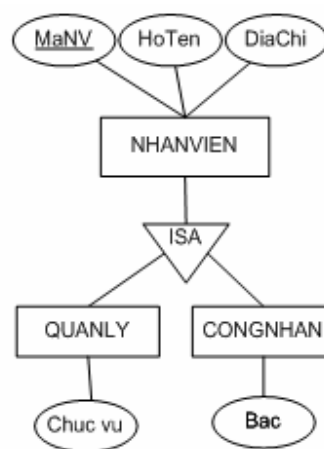
Mặc dù khái niệm bản số của tập thực thể đối với mỗi kết hợp cho chúng ta nhận biết mỗi thể hiện của tập thực thể tham gia tối thiểu là bao nhiêu, và tối đa là bao nhiêu vào mỗi kết hợp. Nhưng trong thực tế, một lớp các đối tượng trong tổ chức có khi tồn tại tình trạng là: một số đối tượng (tập con) của nó tham gia vào một mối kết hợp này, số còn lại có thể tham gia hoặc không vào những mối kết hợp khác, trong khi có thể tất cả các phần tử của chúng lại cùng tham gia vào mỗi kết hợp khác nữa. Hoặc một tập con này có những đặc tính này, còn những phần tử khác thì có thêm những đặc tính khác hoặc không. Chẳng hạn cùng là nhân viên trong nhà máy, thì công nhân trực tiếp tham gia sản xuất, nhân viên quản lý tham gia công tác quản lý. Đối với nhân viên quản lý người ta quan tâm đến chức vụ, còn đối với nhân viên công nhân thì người ta lại quan tâm đến bậc. Để phản ánh tình

trạng đó trong phương pháp mô hình hóa, người ta dùng khái niệm chuyên biệt hóa / tổng quát hóa.

Chuyên biệt hóa nghĩa là phân hoạch một thực thể thành các tập (thực thể) con.

Tổng quát hóa là gộp các tập thực thể thành một tập thực thể bao hàm tất cả các thể hiện của các tập thực thể con. Các chuyên biệt được thừa hưởng tất cả các thuộc tính của các tập thực thể mức trên và chính nó có thể có những thuộc tính khác. Các tập thực thể chuyên biệt có thể có những mối kết hợp khác nhau với những tập thực thể khác và do đó các xử lý sẽ có thể khác nhau tùy theo từng chuyên biệt thành phần.

Ký hiệu:



**Hình 2.8. Chuyên biệt hóa / tổng quát hóa**

### 1.8. Tập thực thể yếu

Một tập thực thể có thể không có đủ các thuộc tính để cấu thành một khóa chính, được gọi là *tập thực thể yếu*. Một tập thực thể mà nó có khóa chính được gọi là *tập thực thể mạnh*. Tập thực thể yếu phải tham gia vào mối quan hệ mà trong đó có một tập thực thể mạnh (tập thực thể mà tập thực thể yếu phụ thuộc)

Coi tập thực thể *thân nhân (THANNHAN)* có các thuộc tính là *tên thân nhân (TenTN)*, *ngày sinh (NTNS)*, *phái (Phai)*, *quan hệ với nhân viên (QuanHe)*. Như vậy tập thực thể này không có khóa chính nên nó là tập thực thể yếu.

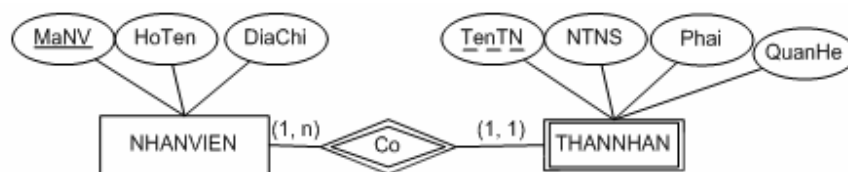
Mặc dù tập thực thể yếu không có khóa chính nhưng chúng ta cần một phương tiện để phân biệt trong số những thực thể thuộc tập thực thể này mà chúng phụ thuộc vào một thực thể mạnh nào đó.

- *Phân phân biệt* của một tập thực thể yếu là tập hợp các thuộc tính cho phép sự phân biệt nói trên được thực hiện. Ví dụ thuộc tính *tên thân nhân (TenTN)* là phân phân biệt của tập thực thể yếu *thân nhân (THANNHAN)*.

- Phân phân biệt của một tập thực thể yếu cũng còn được gọi là *khóa bộ phận* của tập thực thể yếu.
- Khóa chính của tập thực thể yếu được hình thành bởi khóa chính của tập thực thể mạnh mà tập thực thể yếu phụ thuộc vào, cộng với phân phân biệt của tập thực thể yếu. Ví dụ khóa chính của *THANNHAN* là  $\{MaNV, TenTN\}$ .

Kí hiệu:

Một tập thực thể yếu được chỉ ra bởi một hộp nét đôi trong lược đồ, và quan hệ định danh tương ứng được vẽ bởi hình thoi nét đôi. Trong hình, *phân phân biệt* được gạch dưới gián đoạn.



**Hình 2.9. Tập thực thể yếu**

## 2. Ví dụ

### 2.1. Quản lý đề án

Giả sử sau đây là một số yêu cầu dữ liệu đối với một công ty chuyên thực hiện các đề án:

Công ty gồm nhiều nhân viên, mỗi nhân viên được gán mã nhân viên để tiện việc quản lý, có họ tên, ngày sinh, mức lương được hưởng. Công ty gồm nhiều phòng ban, mỗi phòng ban có chức năng riêng của mình, có mã phòng, tên phòng, có một trưởng phòng. Mỗi nhân viên chỉ thuộc vào một phòng ban và một phòng có thể có nhiều nhân viên. Mỗi nhân viên trong phòng còn có thể chịu sự quản lý trực tiếp từ một nhân viên khác.

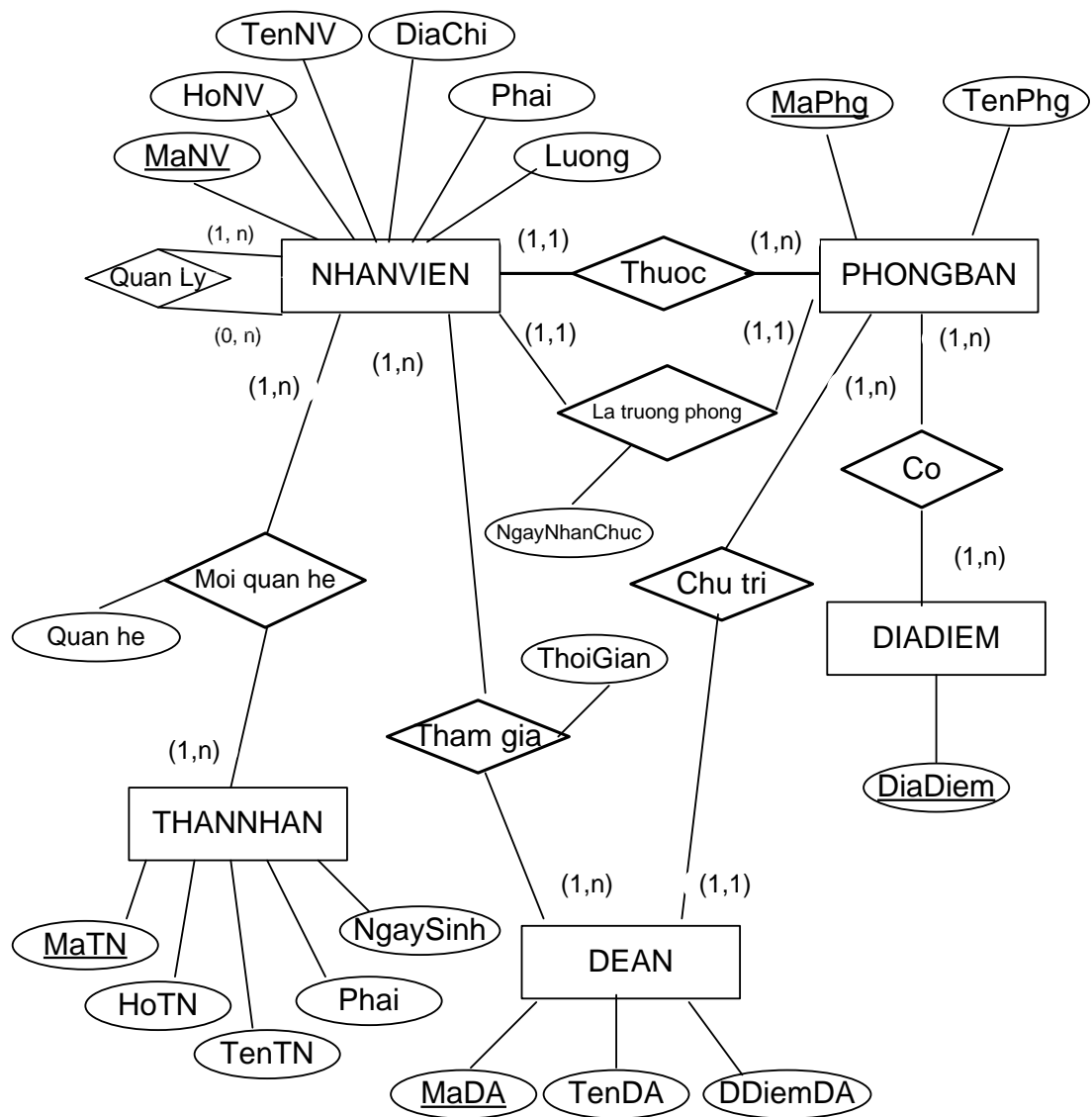
Do công ty thực hiện đề án, nên mỗi phòng có thể có nhiều văn phòng giao dịch hay làm việc khác nhau ở tại những địa điểm khác nhau.

Mỗi một đề án khi được xây dựng, có mã đề án, tên đề án, địa điểm thực hiện đề án đó và do một phòng ban chịu trách nhiệm chủ trì đề án.

Quá trình thực hiện đề án có thể được chia nhỏ thành nhiều công việc và phân công cho các nhân viên thực hiện, khi đó công ty sẽ ghi nhận lại thời gian phân công công việc cho nhân viên (tính bằng số giờ / tuần) để theo dõi tiến độ thực hiện.

Nhằm có thể chăm lo đời sống của nhân viên, công ty có ghi nhận lại những thông tin về những thân nhân của nhân viên, bao gồm những người như cha mẹ, chồng vợ, và con cái.

Từ mô tả tình huống trên, mô hình quan hệ thực thể như sau:



### 3. Bài tập

Hãy xây dựng mô hình thực thể kết hợp cho tình huống sau:

Giả sử sau đây là một số yêu cầu dữ liệu đối với ví dụ hoạt động ngân hàng:

Ngân hàng được tổ chức thành các chi nhánh. Mỗi chi nhánh tọa lạc tại một thành phố và được định danh bởi một tên duy nhất. Ngân hàng theo dõi tài sản của mỗi chi nhánh.

Các khách hàng của ngân hàng được định danh bởi số CMND của họ. Ngân hàng lưu trữ các tên khách hàng, đường phố và thành phố mà khách hàng sinh sống. Khách hàng có thể có tài khoản gửi và tài khoản vay. Một khách hàng có thể được kết hợp với một nhân viên ngân hàng. Nhân viên này có thể là nhân viên cho vay hoặc nhân viên bình thường.

Các nhân viên ngân hàng được định danh bởi mã số nhân viên của họ. Bộ phận quản lý ngân hàng lưu trữ tên và số phone của mỗi nhân viên, tên của các người phụ thuộc nhân viên và mã số nhân viên của người quản lý nhân viên. Ngân hàng cũng theo dõi ngày bắt đầu làm việc của nhân viên, và thời gian thuê nhân viên làm việc.

Ngân hàng đưa ra các loại tài khoản gửi, tài khoản tiết kiệm và tài khoản séc. Các tài khoản gửi có thể được nắm giữ bởi nhiều hơn một khách hàng, và một khách hàng có thể có nhiều hơn một tài khoản. Mỗi tài khoản gửi được gán bởi một số tài khoản duy nhất. Ngân hàng duy trì thông tin về cân đối của mỗi tài khoản gửi và ngày gần nhất mà tài khoản gửi được truy cập bởi mỗi khách hàng nắm giữ tài khoản đó. Ngoài ra, mỗi tài khoản tiết kiệm còn có mức lãi suất, và tài khoản séc có số tiền rút vượt mức.

Một tài khoản vay đầu tiên xuất phát từ một chi nhánh nào đó, và nó có thể được nắm giữ bởi một hoặc nhiều khách hàng. Mỗi tài khoản vay được gán bởi một số tài khoản duy nhất. Với mỗi tài khoản vay ngân hàng theo dõi số tiền vay và số tiền trả. Mặc dù số thứ tự lần trả tiền vay (gọi tắt là số lần trả) không xác định duy nhất lần trả đối với các tài khoản vay của ngân hàng nhưng nó xác định duy nhất lần trả đối với một tài khoản vay cụ thể. Ngày và số tiền trả đối với mỗi lần trả tiền vay cũng được ngân hàng theo dõi ghi nhận.

# Mô Hình Dữ Liệu Quan Hệ

---

Mô hình dữ liệu quan hệ lần đầu tiên được đề nghị bởi Edgar F. Codd vào năm 1970. Hiện nay mô hình quan hệ là mô hình ưu thế đối với các ứng dụng xử lý dữ liệu thương mại. Chương này sẽ trình bày chi tiết về các khái niệm đã nhắc tới trong chương 1 và coi đó như là những cơ sở nền tảng để tiếp tục nghiên cứu các phần tiếp theo.

## 1. Các khái niệm cơ bản

### 1.1. Thuộc tính

Thuộc tính (attribute) là một tính chất riêng biệt của một đối tượng cần được lưu trữ trong CSDL để phục vụ cho việc khai thác dữ liệu về đối tượng.

Ví dụ:

- Đối tượng LOPHOC có các thuộc tính mã lớp, tên lớp, khóa, số học viên.
- Đối tượng SINHVIEN có các thuộc tính mã sinh viên, họ tên, ngày sinh, quê quán.

Các thuộc tính được đặc trưng bởi một tên thuộc tính, kiểu giá trị (data type) và miền giá trị (domain).

Trong các ứng dụng thực tế, người phân tích – thiết kế thường đặt *tên thuộc tính* một cách gợi nhớ, tuy nhiên không nên đặt tên quá dài (vì làm cho việc viết câu lệnh truy vấn vất vả hơn) nhưng cũng không nên quá ngắn (vì không thể hiện được ngữ nghĩa một cách rõ ràng).

Ví dụ: nếu có hai đối tượng HOCVIEN và GIAOVIEN đều có thuộc tính tên thì nên đặt tên một cách rõ ràng là Tên\_học\_viên và Tên\_giáo\_viên vì chúng mang ngữ nghĩa hoàn toàn khác nhau.

Mỗi một thuộc tính đều phải thuộc một *kiểu dữ liệu*. Kiểu dữ liệu có thể là vô hướng - là các kiểu dữ liệu cơ bản như chuỗi, số, logic, ngày tháng... hoặc các kiểu có cấu trúc được định nghĩa dựa trên các kiểu dữ liệu đã có sẵn.

Mỗi hệ quản trị CSDL có thể gọi tên các kiểu dữ liệu nói trên bằng các tên gọi khác nhau, ngoài ra còn bổ sung thêm một số kiểu dữ liệu riêng của mình. Ví dụ, Microsoft Access

có kiểu dữ liệu text, memo là kiểu chuỗi; SQL Server có kiểu dữ liệu text, char, varchar, nvarchar là kiểu chuỗi.

Mỗi một thuộc tính có thể chỉ chọn lấy những giá trị trong một tập hợp con của kiểu dữ liệu. Tập hợp các giá trị mà một thuộc tính A có thể nhận được gọi là *miền giá trị* của thuộc tính A, thường được ký hiệu là  $MGT(A)$  hoặc  $Dom(A)$ .

Ví dụ:

Điểm của sinh viên là một số, nhưng luôn nằm trong đoạn từ 0 đến 10.

Với kiểu dữ liệu cấu trúc thì miền giá trị chính là tích đề các (hoặc tập con của tích đề các) của các miền giá trị thành phần.

Trong nhiều hệ quản trị CSDL, thường đưa thêm vào miền giá trị của các thuộc tính một giá trị đặc biệt gọi là giá trị null. Tùy theo ngữ cảnh mà giá trị này có thể là một giá trị không thể xác định được hay một giá trị chưa được xác định ở thời điểm nhập tin và có thể được xác định vào một thời điểm khác.

Nếu thuộc tính có kiểu dữ liệu vô hướng thì nó được gọi là thuộc tính đơn hoặc nguyên tố, nếu thuộc tính có kiểu dữ liệu có cấu trúc thì ta nói rằng nó là thuộc tính kép.

## 1.2. Quan hệ n ngôi

Một quan hệ R có n ngôi được định nghĩa trên tập các thuộc tính  $U = \{A_1, A_2, \dots, A_n\}$  và kèm theo nó là một tân từ, để xác định mối quan hệ giữa các thuộc tính  $A_i$ , và được ký hiệu là  $R(A_1, A_2, \dots, A_n)$ . Tập thuộc tính của quan hệ R còn được ký hiệu là  $R^+$ .

Với  $A_i$  là một thuộc tính có miền giá trị là  $MGT(A_i)$ , như vậy  $R(A_1, A_2, \dots, A_n)$  là tập con của tích đề các:  $MGT(A_1) \times MGT(A_2) \times \dots \times MGT(A_n)$

Quan hệ còn được gọi là bảng (table).

Ví dụ:

KHOA(Mã\_khoa, Tên\_khoa) là một quan hệ 2 ngôi với tân từ : “*Mỗi khoa có một mã khoa duy nhất để phân biệt với các khoa khác, có một tên gọi*”.

SINHVIEN (Mã\_sinh\_viên, Tên\_sinh\_viên, Ngày\_sinh, Quê\_quán, Khoa) là một quan hệ 5 ngôi với tân từ : “*Mỗi sinh viên có một mã sinh viên duy nhất để phân biệt với các sinh viên khác, có họ tên, ngày tháng năm sinh, quê quán và học tại một khoa trong trường*”.

## 1.3. Bộ

Một bộ (tuple) giá trị là các thông tin của một đối tượng thuộc quan hệ. Bộ giá trị cũng thường được gọi là một mẫu tin hay bản ghi (record), dòng của bảng (row).

Một bộ  $q$  là một vectơ gồm  $n$  thành phần thuộc tập hợp con của tích đề các miền giá trị của các thuộc tính và thỏa mãn tất cả các ràng buộc của quan hệ.

Ví dụ: các bộ giá trị dựa trên các thuộc tính của quan hệ SINHVIEN

$q_1 = (SV001, \text{Trần Văn Mạnh}, 10/10/1980, \text{Lâm Đồng}, \text{CTK27})$

$q_2 = (SV002, \text{Nguyễn Thị Hoa Huệ}, 25/11/1985, \text{Khánh Hòa}, \text{MTK27})$

$q_3 = (SV003, \text{Tăng Thanh Hà}, 11/11/1982, \text{Tp. Hồ Chí Minh}, \text{NVK27})$

Để lấy thành phần  $A_i$  – là giá trị thuộc tính  $A_i$  của một bộ giá trị  $q$ , ký hiệu  $q.A_i$ . Đây được gọi là *phép chiếu một bộ lên thuộc tính  $A_i$*

Ví dụ:

$q_1.\text{Tên\_sinh\_viên} = \text{“Trần Văn Mạnh”}$

$q_2.\text{Khoa} = \text{“MTK27”}$

#### 1.4. Lược đồ quan hệ

Lược đồ quan hệ (Relation schema) là sự trừu tượng hóa của quan hệ, một sự trừu tượng hóa ở mức cấu trúc của một bảng hai chiều. Khi nói đến lược đồ quan hệ tức là đề cập tới cấu trúc tổng quát của một quan hệ; khi nói đến một quan hệ thì hiểu rằng đó là một bảng có cấu trúc cụ thể trên một lược đồ quan hệ với các bộ giá trị của nó.

Lược đồ cơ sở dữ liệu là tập hợp các lược đồ quan hệ con  $\{R_i\}$ , ký hiệu là  $\zeta$ .

Thể hiện (hay tình trạng) của quan hệ  $R$ , ký hiệu là  $T_R$ , là tập hợp các bộ giá trị của quan hệ  $R$  vào một thời điểm. Như vậy, tại những thời điểm khác nhau thì quan hệ có những thể hiện khác nhau.

Thể hiện của các lược đồ quan hệ con  $T_{R_i}$  gọi là tình trạng của lược đồ CSDL  $\zeta$

Ví dụ về thể hiện của quan hệ KHOA và LOPHOC

**Bảng 3.1. Thể hiện của quan hệ KHOA**

Mã khoa	Tên khoa	Ngày thành lập
CNTT	Công nghệ thông tin	10/10/1994
TH	Toán học	20/10/1976
VL	Vật lý	20/10/1976
HH	Hóa học	20/10/1976



**Bảng 3.2. Thể hiện của quan hệ LOPHOC**

Mã lớp	Tên lớp	Số học viên	Mã khoa
CTK27	Công nghệ thông tin K27	75	CNTT
HHK18	Hóa học K18	95	HH
THK20	Toán học K20	120	TH

### 1.5. Khóa của một quan hệ

Quan hệ R định nghĩa trên tập các thuộc tính  $U = \{A_1, A_2, \dots, A_n\}$

Khi đó  $K \subseteq U$  là khóa của quan hệ R nếu thỏa:

- (i) K xác định được giá trị của  $A_j$ , với mọi  $j = 1, 2, \dots, n$
- (ii) Không tồn tại  $K' \subseteq K$  mà  $K'$  có thể xác định được giá trị của  $A_j$ , với mọi  $j = 1, 2, \dots, n$

Theo định nghĩa trên,  $K$  là tập con nhỏ nhất mà giá trị của nó có thể xác định được duy nhất một bộ giá trị của quan hệ.

Khóa theo định nghĩa trên gọi là *khóa đề nghị* (candidate key).

$K$  được gọi là *siêu khóa* (superkey) của quan hệ R nếu  $K' \subseteq K$  là một khóa của quan hệ. Như vậy một quan hệ Q luôn có ít nhất một siêu khóa và có thể có nhiều siêu khóa.

Ví dụ:

Với quan hệ LOPHOC (MaLop, TenLop, SoSV, MaKhoa)

Siêu khoá :

$$K1 = \{\text{MaLop}, \text{TenLop}\}$$

$$K2 = \{\text{MaLop}, \text{SoSV}\}$$

$$K3 = \{\text{MaLop}, \text{TenLop}, \text{MaKhoa}\}$$

$$K4 = \{\text{MaLop}, \text{SoSV}, \text{MaKhoa}\}$$

Ý nghĩa thực tế của khóa là dùng để nhận diện một bộ trong một quan hệ, khi cần thiết tìm thông tin một bộ  $q$  nào đó ta chỉ cần biết giá trị của khóa của  $q$  là đủ để dò tìm và hoàn toàn xác định được nó trong quan hệ.

Trong thực tế, đối với các loại thực thể tồn tại khách quan như NHANVIEN, SINHVIEN, MATHANG,... người thiết kế CSDL thường gán thêm một thuộc tính giả là mã số để làm khóa như MaNV, MSSV, MaHang,... Đối với các lược đồ quan hệ biểu diễn cho sự trừu tượng hóa thường có khóa chỉ định là một tổ hợp của hai hay nhiều thuộc tính.

Trong trường hợp lược đồ quan hệ có nhiều khóa đề nghị, khi cài đặt lên một hệ quản trị CSDL người ta chọn ra một khóa trong số các khóa đề nghị này để sử dụng. Khi đó khóa này được gọi là *khóa chính* (primary key) và các khóa còn lại là *khóa tương đương*. Khóa chính chỉ thật sự có ý nghĩa trong quá trình khai thác CSDL, khóa chính hoàn toàn không có vai trò gì khác so với các khóa chỉ định còn lại.

Trong các hệ quản trị CSDL có cài đặt cơ chế tự động kiểm tra tính duy nhất của khóa chính. Khi người sử dụng thêm một bộ mới  $q_2$  có giá trị khóa chính trùng với giá trị khóa chính của một bộ  $q_1$  đã có trong quan hệ thì hệ thống sẽ báo lỗi và yêu cầu nhập lại giá trị khác.

Các thuộc tính tham gia vào một khóa được gọi là thuộc tính khóa. Về mặt ký hiệu, trong lược đồ quan hệ các thuộc tính khóa được gạch dưới.

Trong một bộ của quan hệ, các thuộc tính khóa không chứa giá trị rỗng.

Không được phép sửa đổi giá trị của thuộc tính khóa, nếu người sử dụng muốn sửa giá trị thuộc tính khóa của bộ  $q$ , cần phải hủy bỏ bộ  $q$  sau đó thêm vào bộ  $q'$  với giá trị khóa đã được sửa đổi.

Ví dụ: Một lược đồ CSDL như sau:

KHOA (MaKhoa, TenKhoa, NgayThanhLap)

LOPHOC (MaLop, TenLop, NienKhoa, SoHocvien, MaKhoa)

MONHOC (MaMon, TenMon, SoTC)

HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, MaLop)

GIAOVIEN (MaGV, HoGV, TenGV, NgaySinh, HocVi, ChuyenNganh)

KQUATHI (MaHV, MaMon, LanThi, NgayThi, DiemThi, GhiChu)

DAY (MaGV, MaLop, MaMon)

Với hai quan hệ  $R$  và  $S$ , một tập thuộc tính  $K$  của quan hệ  $R$  được gọi là *khóa ngoại* (foreign key) của quan hệ  $R$  nếu  $K$  là khóa của quan hệ  $S$ .

Ví dụ:

- Trong quan hệ LOPHOC, MaKhoa là khoá ngoại vì MaKhoa là khóa của quan hệ KHOA.
- Trong quan hệ HOCVIEN, MaLop là khoá ngoại vì MaLop là khóa của quan hệ LOPHOC.

### 1.6. Ràng buộc toàn vẹn

Ràng buộc toàn vẹn (RBTV) là một quy tắc định nghĩa trên một hay nhiều quan hệ do môi trường ứng dụng quy định. Đó chính là quy tắc để bảo đảm tính nhất quán của dữ liệu trong CSDL.

Thông thường mỗi RBTV được định nghĩa bằng một thuật toán trong CSDL.

Ví dụ:

- Trong quan hệ KQUATHI, DiemThi là một số nguyên nằm trong khoảng từ 0 đến 10.
- Trong quan hệ KQUATHI, LanThi là 1 hoặc 2.

## 2. Các thao tác cơ bản trên quan hệ

Các thao tác cơ bản trên một quan hệ là thêm (insert), xóa (delete), sửa (update) các bộ giá trị của quan hệ.

### 2.1. Phép thêm

Việc thêm một bộ mới  $t$  vào quan hệ  $R(A_1, A_2, \dots, A_n)$  làm cho thể hiện  $T_R$  tăng thêm một phần tử mới  $T_R = R \cup t$

Dạng thức của phép thêm bộ mới là:

$$INSERT(R; A_1 = v_1, A_2 = v_2, \dots, A_n = v_n),$$

với:  $A_1, A_2, \dots, A_n$  là các thuộc tính và  $v_1, v_2, \dots, v_n$  là giá trị muốn thêm vào (với điều kiện là các giá trị này thuộc  $MGT(A_1), MGT(A_2), \dots, MGT(A_n)$  tương ứng)

Trong trường hợp này nếu như các thuộc tính không được liệt kê trong danh sách gán giá trị của bộ  $t$  trong câu lệnh INSERT sẽ nhận giá trị null.

Nếu xem thứ tự của các thuộc tính là cố định và các giá trị  $v_1, v_2, \dots, v_n$  là hoàn toàn tương ứng thì phép chèn có thể viết dưới dạng tường minh như sau:

$$INSERT(R; v_1, v_2, \dots, v_n)$$

Chú ý rằng có thể phép chèn không được thực hiện hoặc làm mất tính nhất quán của dữ liệu vì các lý do sau:

- Giá trị khóa của bộ mới là null hoặc trùng với giá trị khóa của một bộ đã có trong CSDL. Hệ quản trị CSDL sẽ không cho thêm mới trong trường hợp này.
- Bộ mới không phù hợp với lược đồ quan hệ. Trường hợp này xảy ra khi người sử dụng thêm mới các giá trị sai thứ tự, sai kiểu hoặc độ lớn của các thuộc tính trong lược đồ. Hệ quản trị CSDL có thể sẽ không cho bổ sung nếu không tương thích kiểu giá trị, hoặc vẫn cho bổ sung bộ mới nhưng tính nhất quán của dữ liệu không được đảm bảo.
- Một số giá trị của bộ mới không thuộc miền giá trị của thuộc tính tương ứng. Trong trường hợp này, nếu quan hệ đã được đảm bảo tính nhất quán bởi các RBTV về miền giá trị thì hệ quản trị CSDL sẽ không cho bổ sung; ngược lại, nếu không có RBTV về miền giá trị thì tính nhất quán của CSDL bị vi phạm mà hệ quản trị CSDL không phát hiện được.

## 2.2. Phép xóa

Phép xóa bộ  $t$  của quan hệ sẽ xóa đi một (hoặc nhiều) bộ  $t$  khỏi thể hiện của quan hệ  $T_R = R \setminus t$

Dạng thức của phép xóa là:

$$DELETE(R; A_1 = v_1, A_2 = v_2, \dots, A_n = v_n),$$

với  $A_i = v_j; j = 1..n$  chính là điều kiện thỏa một số thuộc tính của bộ  $t$  để loại một bộ  $t$  ra khỏi quan hệ.

Ví dụ: Với quan hệ:

HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, Khoa),

và phép loại bỏ :  $DELETE(HOCVIEN; QueQuan = "NhaTrang")$

thì tất cả các bộ trong thể hiện HOCVIEN có quê quán ở Nha trang sẽ bị loại bỏ.

## 2.3. Phép sửa

Dạng thức của phép sửa là:

$$UPDATE(R; A_1 = c_1, A_2 = c_2, \dots, A_n = c_n; A_1 = v_1, A_2 = v_2, \dots, A_n = v_n),$$

với  $A_i = c_j; j = 1..n$  là điều kiện thỏa tìm kiếm bộ muốn sửa và  $A_i = v_j; j = 1..n$  là giá trị mới cần cập nhật

Ví dụ: Với quan hệ:

HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, MaLop),

Trong thể hiện của HOCVIEN có bộ:

$q = \{ \text{"HV001"}, \text{"Nguyễn Văn"}, \text{"Mạnh"}, 20/10/85, \text{"Nghệ An"}, \text{"CNTT"} \}$

và phép cập nhật:  $UPDATE(HOCVIEN; MaHV = HV001; QueQuan = \text{Lâm Đồng})$

khi đó kết quả đạt được sẽ là bộ q được sửa lại với giá trị :

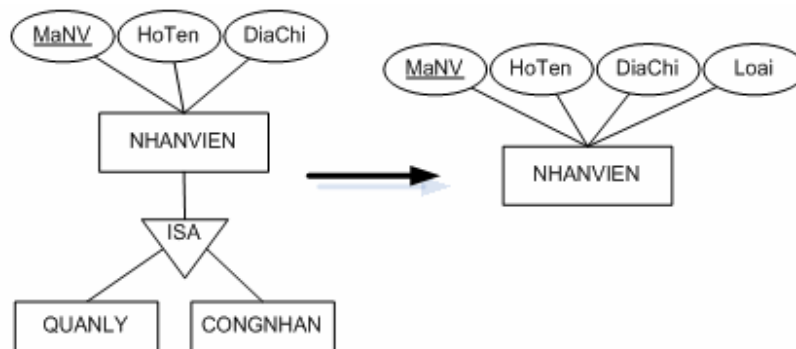
$q = \{ \text{"HV001"}, \text{"Nguyễn Văn"}, \text{"Mạnh"}, 20/10/85, \text{"Lâm Đồng"}, \text{"CNTT"} \}$

### 3. Các bước chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ

#### 3.1. Biến các tập thực thể chuyên biệt hóa về dạng bình thường

##### 3.1.1. Tập thực thể chuyên biệt không có thuộc tính riêng

Trong trường hợp này, đưa tập thực thể chuyên biệt lên làm thuộc tính của tập thực thể mức tổng quát, khi đó có thuộc tính mới cho biết loại của thực thể chuyên biệt.

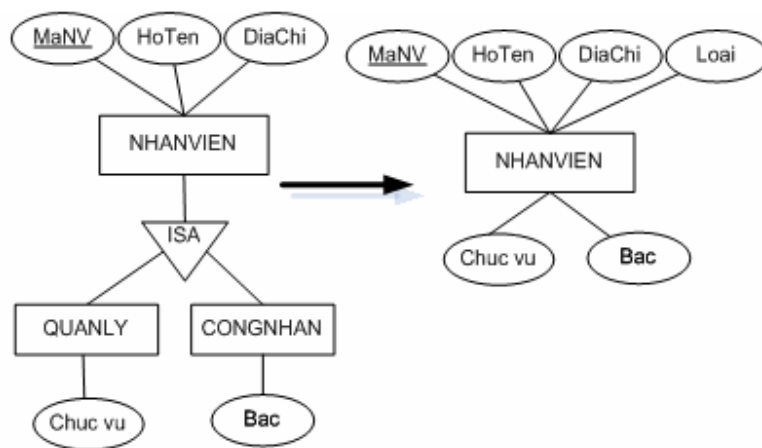


**Hình 3.1. Biến đổi tập thực thể chuyên biệt không có thuộc tính riêng**

Trong ví dụ trên  $Loai = \{ NV\_QL, NV\_CN \}$

##### 3.1.2. Tập thực thể chuyên biệt có thuộc tính riêng

Nếu số lượng thuộc tính riêng ở tập thực thể chuyên biệt ít, gom lên mức tổng quát, và bổ sung thêm ràng buộc.



**Hình 3.2. Biến đổi tập thực thể chuyên biệt có thuộc tính riêng**

Trong ví dụ trên  $Loai = \{NV\_QL, NV\_CN\}$  và các ràng buộc:

- RBTV1: “Nếu  $Loai = NV\_QL$  thì thuộc tính *Chucvu* mới có giá trị”
- RBTV2: “Nếu  $Loai = NV\_CN$  thì thuộc tính *Bac* mới có giá trị”

### 3.2. Chuyển tất cả các tập thực thể thành quan hệ

#### Tập thực thể mạnh

Với mỗi tập thực thể mạnh, chuyển thành quan hệ với khóa là khóa của tập thực thể.

NHANVIEN (MaNV, HoTen, DiaChi)

PHONGBAN (MaPhong, TenPhong)

#### Tập thực thể yếu

Với mỗi tập thực thể yếu, chuyển thành quan hệ với:

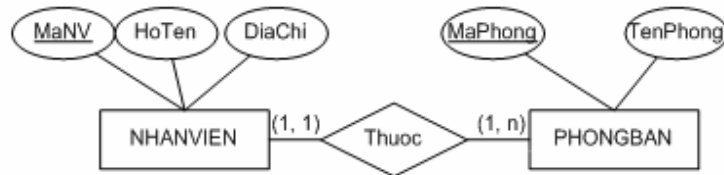
- Các thuộc tính là các thuộc tính của tập thực thể yếu và khóa của tập thực thể mạnh mà nó phụ thuộc
- Khóa của quan hệ là khóa của tập thực thể mạnh và phần phân biệt của tập thực thể yếu.

THANNHAN (MaNV, TenTN, NTNS, Phai, QuanHe)

### 3.3. Mỗi kết hợp

**Mỗi kết hợp có bản số (1,1) và (1, n), chuyển thành các quan hệ với:**

- Tập thực thể có sự tham gia (1, n) chuyển như bình thường
- Tập thực thể có sự tham gia (1, 1) chuyển thành quan hệ gồm *tất cả các thuộc tính của thực thể và thuộc tính khóa của thực thể có sự tham gia (1, n)*.

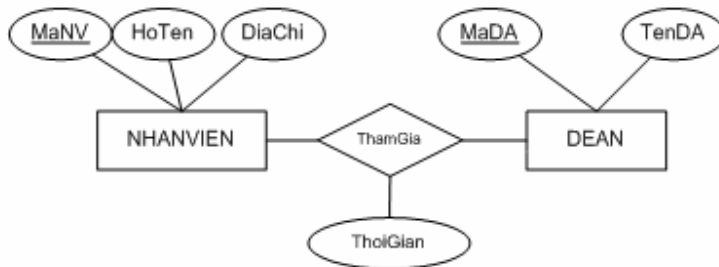


NHANVIEN (MaNV, HoTen, DiaChi, MaPhong)

PHONGBAN (MaPhong, TenPhong)

**Mỗi kết hợp khác, chuyển thành một quan hệ với:**

- Thuộc tính là khác khóa của các tập thực thể tham gia vào mỗi kết hợp và mọi thuộc tính của mỗi kết hợp
- Khóa là khóa của các tập thực thể tham gia vào mỗi kết hợp và có thể có thêm thuộc tính của mỗi kết hợp



THAMGIA (MaNV, MaDA, ThoiGian): *Khóa là khóa của các tập thực thể tham gia vào mỗi kết hợp*

### 3.4. Nhập tất cả các quan hệ có cùng khóa

Sau khi thực hiện chuyển đổi, có thể có một số quan hệ có cùng khóa, khi đó thực hiện việc nhập tất cả các quan hệ có cùng khóa lại thành một quan hệ.

## Chương 4

# Ngôn Ngữ Đại Số Quan Hệ

---

Ngôn ngữ đại số quan hệ là ngôn ngữ phi thủ tục. Nó bao gồm tập hợp các phép toán được áp dụng trên các thể hiện của quan hệ, kết quả của một câu truy vấn là một thể hiện của quan hệ. Ngôn ngữ đại số quan hệ có ưu điểm trong việc thể hiện kế hoạch thực hiện câu truy vấn và các kỹ thuật tối ưu hóa câu truy vấn.

### 1. Các phép toán cơ sở

#### 1.1. Các phép toán tập hợp

Các phép toán cơ bản được áp dụng trên tập các bộ giá trị của các quan hệ, được hình thành từ lý thuyết tập hợp toán học: hợp hay hội (union), hiệu (minus), giao (intersection), tích đề các (cartesian product operation), chia (division), bù (complement). Quan hệ kết quả của các phép toán hợp, hiệu, trừ có cùng tên thuộc tính với quan hệ đầu tiên (quy ước).

Giả thiết:  $R(A_1, A_2, \dots, A_n)$  và  $S(B_1, B_2, \dots, B_n)$

#### Tính khả hợp

Quan hệ  $R(A_1, A_2, \dots, A_n)$  và  $S(B_1, B_2, \dots, B_n)$  được gọi là khả hợp khi:

- Số bậc của  $R$  và  $S$  là bằng nhau, nghĩa là cùng số lượng thuộc tính.
- Miền giá trị của thuộc tính phải tương thích  $MGT(A_i) = MGT(B_i)$ , với  $1 \leq i \leq n$

##### 1.1.1. Phép hợp (union)

Hợp (hay còn gọi là hội) của hai quan hệ khả hợp  $R$  và  $S$ , ký hiệu  $R \cup S$ , là quan hệ  $Q$  được định nghĩa như sau:

$$Q = R \cup S = \{t \mid t \in R \vee t \in S\}$$

Ví dụ:



A	B
A1	b1
a1	b2
a2	b1

R

A	B
a1	b2
a2	b3

S

A	B
a1	b1
a1	b2
a2	b1
a2	b3

 $R \cup S$ 

Nói cách khác, hợp của hai quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S, với các bộ giá trị bằng gộp các bộ của cả R và S, những bộ giá trị trùng nhau chỉ giữ lại 1 bộ. Trong ví dụ trên bộ {a1, b2} xuất hiện trong cả R và S, do đó chỉ xuất hiện 1 lần trong  $R \cup S$

### 1.1.2. Phép trừ (minus)

Phép trừ (hay còn gọi là hiệu) của hai quan hệ khả hợp R và S, ký hiệu  $R - S$ , là quan hệ Q được định nghĩa như sau:

$$Q = R - S = \{t \mid t \in R \wedge t \notin S\}$$

Ví dụ:

A	B
a1	b1
a1	b2
a2	b1

R

A	B
a1	b2
a2	b3

S

A	B
a1	b1
a2	b1

 $R - S$ 

Nói cách khác, hợp của hai quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S, với các bộ giá trị là các bộ của cả R sau khi đã loại bỏ các bộ có mặt trong S.

### 1.1.3. Phép giao (intersect)

Giao của hai quan hệ khả hợp R và S, ký hiệu  $R \cap S$ , là quan hệ Q được định nghĩa như sau:

$$Q = R \cap S = \{t \mid t \in R \wedge t \in S\}$$

Ví dụ:

A	B
a1	b1
a1	b2
a2	b1

R

A	B
a1	b2
a2	b3

S

A	B
a1	b2

$R \cap S$

Nói cách khác, giao của hai quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S, với các bộ giá trị là các bộ giống nhau trong cả R và S.

#### 1.1.4. Phép tích đề các (Cartesian Product Operation)

Giả thiết:  $R(A_1, A_2, \dots, A_n)$  và  $S(B_1, B_2, \dots, B_m)$

Phép tích đề các của hai quan hệ của hai quan hệ R và S, ký hiệu  $R \times S$ , là quan hệ Q được định nghĩa như sau:

$$Q = R \times S = \{tq \mid t \in R \wedge q \in S\}$$

Ví dụ:

A	B
a1	b1
a1	b2
a2	b1

C	D
c1	d2
c2	d3

S

A	B	C	D
a1	b1	c1	d2
a1	b1	c2	d3
a1	b2	c1	d2
a1	b2	c2	d3
a2	b1	c1	d2
a2	b1	c2	d3

$R \times S$

Vậy tích đề các của hai quan hệ R và S là một quan hệ gồm (n+m) ngôi với n thuộc tính đầu là một bộ của R và m thuộc tính sau là một bộ thuộc S.

#### 1.1.5. Phép chia (division)

Giả thiết:  $R(A_1, A_2, \dots, A_n)$  và  $S(B_1, B_2, \dots, B_m)$  ( $n > m, S \neq \emptyset$ ), có  $m$  thuộc tính chung. Khi đó phép chia trên 2 quan hệ  $R$  và  $S$ , ký hiệu  $R \div S$ , là quan hệ  $Q$  có  $(n-m)$  ngôi được định nghĩa như sau:

$$Q = R \div S = \{t \mid \forall u \in S, (t, u) \in R\}$$

Ví dụ 1:

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d3
a2	b1	c2	d3
a2	b1	c1	d1
a3	b1	c1	d3

R

C	D
C1	d1
C2	d3

S

A	B
a1	b1
a2	b1

$R \div S$

Nói cách khác, phép chia của quan hệ  $R$  cho  $S$  đưa ra tất cả các bộ trong quan hệ  $R$ , sao cho khớp với tất cả các bộ trong quan hệ  $S$

## 1.2. Các phép toán quan hệ

Phần này trình bày các phép toán trên quan hệ và ví dụ minh họa dựa trên lược đồ cơ sở dữ liệu Quản lý đề án:

NHANVIEN (MaNV, HoNV, tenNV, NgaySinh, DiaChi, Phai, Luong, MaNQL, Phong)

*Tân từ:* Mỗi nhân viên có Mã nhân viên (MaNV) duy nhất để phân biệt với các nhân viên khác, có họ tên (HoNV, TenNV), ngày sinh (NgaySinh), địa chỉ (DiaChi), phái Nam hoặc Nữ (Phai), mức lương (Luong), người quản lý trực tiếp (MaNQL) và thuộc về một phòng ban (Phong)

PHONGBAN (MaPhong, TenPhong, TruongPhong, NgayNhanChuc)

*Tân từ:* Mỗi một phòng ban có một mã phòng duy nhất (MaPhong) để phân biệt với các phòng ban khác, có tên phòng (TenPhong), người trưởng phòng (TruongPhong), và ngày nhận chức của trưởng phòng (NgayNhanChuc)

DIADIEMPHONG (MaPhong, DiaDiem)

*Tân từ:* Mỗi một phòng ban (MaPhong) có thể có nhiều địa điểm làm việc khác nhau (DiaDiem)

DEAN (MaDA, TenDA, DdiemDA, Phong)

*Tân từ:* Mỗi một đề án có một mã đề án duy nhất (MaDA) để phân biệt với các đề án khác, có tên đề án (TenDA), địa điểm thực hiện (DdiemDA), và do một phòng ban chủ trì đề án đó (Phong)

PHANCONG (MaNV, MaDA, ThoiGian)

*Tân từ:* Mỗi một nhân viên (MaNV) được phân công tham gia đề án (MaDA) dưới dạng tham gia số giờ trên 1 tuần (ThoiGian)

THANNHAN(MaTN, HoTN, TenTN, Phai, NgaySinh)

*Tân từ:* Mỗi thân nhân có Mã thân nhân (MaTN) duy nhất để phân biệt với các thân nhân khác, có họ tên (HoTN, TenTN), phái (Phai) ngày sinh (NgaySinh)

NVIEN\_TNHAN(MaNV, MaTN, QuanHe)

*Tân từ:* Mỗi nhân viên (MaNV) có thể có nhiều thân nhân (MaTN), được diễn giải bởi quan hệ (QuanHe) như vợ, chồng, con, anh em...

### 1.2.1. Phép chọn (selection)

Cho phép chọn ra những bộ trong R thỏa mãn biểu thức điều kiện chọn P cho trước. Ký hiệu là  $\sigma_P(R)$  với định nghĩa:

$$\sigma_P(R) = \{t \mid t \in R \wedge P(t)\}$$

với R là quan hệ được chọn, P là biểu thức logic chứa các phép so sánh ( $\neq, \geq, \leq, =, \dots$ ), các phép toán logic ( $\wedge, \vee, \neg$ ) dạng:

<Thuộc tính> <phép so sánh> <Thuộc tính> hay <Hằng số>

Như vậy kết quả của phép chọn là một quan hệ có cùng danh sách thuộc tính với quan hệ R.

Ví dụ:

- Chọn những nhân viên có lương  $\geq 500000$

$$\sigma_{Lương \geq 500000}(NHANVIEN)$$

- Cho biết những nhân viên thuộc phòng số 5 và có lương  $\geq 500000$

$$\sigma_{Luong \geq 500000 \wedge Phong = 5}(NHANVIEN)$$

### 1.2.2. Phép chiếu (projection)

Cho phép trích chọn ra những cột (thuộc tính) trong R chỉ ra trong danh sách thuộc tính. Ký hiệu là  $\pi_{A_1, A_2, \dots, A_k}(R)$  với  $A_1, A_2, \dots, A_k$  là danh sách các thuộc tính cần chọn và R là quan hệ cần trích chọn.

Nhận thấy rằng số lượng các bộ kết quả luôn nhỏ hơn hoặc bằng số lượng các bộ trong R. Các bộ trùng nhau sẽ loại đi và chỉ giữ lại bộ.

Ví dụ:

- Cho biết mã nhân viên, họ tên của tất cả các nhân viên

$$\pi_{MaNV, HoNV, TenNV}(NHANVIEN)$$

- Cho biết mã nhân viên, họ tên, phòng làm việc và mức lương của tất cả các nhân viên

$$\pi_{MaNV, HoNV, TenNV, Phong, Luong}(NHANVIEN)$$

- Cho biết các đề án cùng với các phòng phụ trách đề án đó

$$\pi_{MaDA, TenDA, Phong}(DEAN)$$

### 1.2.3. Phép gán (assignment)

Khi gặp những truy vấn phức tạp, phép gán cho phép diễn tả một cách rõ ràng hơn câu truy vấn. Khi đó, câu truy vấn chính là một chuỗi các phép gán theo sau đó là một biểu thức có giá trị như là kết quả của câu truy vấn.

Ký hiệu:  $\leftarrow$

Việc gán được thực hiện cho một biến quan hệ tạm và được sử dụng cho các biểu thức theo sau.

### 1.2.4. Phép đổi tên:

Vì cho phép đặt tên nên có thể tham chiếu tới kết quả của biểu thức đại số quan hệ, và cho phép tham chiếu tới một quan hệ bằng nhiều tên

Các phép đổi tên sau:

- Đổi tên quan hệ và tên thuộc tính: Cho biểu thức đại số quan hệ E có n thuộc tính, biểu thức  $\rho_{\lambda(A_1, A_2, \dots, A_n)}(E)$  trả về kết quả biểu thức E dưới tên  $\lambda$  và các tên của thuộc tính đổi thành  $A_1, A_2, \dots, A_n$
- Đổi tên quan hệ: Cho biểu thức đại số quan hệ E, biểu thức  $\rho_{\lambda}(E)$  trả về kết quả biểu thức E dưới tên  $\lambda$
- Đổi tên thuộc tính: Cho biểu thức đại số quan hệ E có n thuộc tính, biểu thức  $\rho_{(A_1, A_2, \dots, A_n)}(E)$  trả về kết quả biểu thức E với các tên của thuộc tính đổi thành  $A_1, A_2, \dots, A_n$

### 1.2.5. Chuỗi các phép toán

Kết hợp các phép toán đại số quan hệ với nhau để tạo ra một quan hệ kết quả theo yêu cầu.

Có nhiều cách để thể hiện một truy vấn cho trước.

Ví dụ: Cho biết mã nhân viên, họ tên cùng lương của nhân viên làm việc trong phòng số 4

- Cách 1:  $\pi_{MaNV, HoNV, TenNV, Luong}(\sigma_{Phong=4}(NHANVIEN))$
- Cách 2:

$$NVIEN\_P4 \leftarrow \sigma_{Phong=4}(NHANVIEN)$$

$$KQ \leftarrow \pi_{MaNV, HoNV, TenNV, Luong}(NVIEN\_P4)$$

- Cách 3:

$$NVIEN\_P4 \leftarrow \sigma_{Phong=4}(NHANVIEN)$$

$$KQ(MaNV, HoNV, TenNV, LuongNV) \leftarrow \pi_{MaNV, HoNV, TenNV, Luong}(NVIEN\_P4)$$

Trong đó có sử dụng phép gán và phép đặt lại tên.

## 2. Các phép toán khác

### 2.1. Phép kết hai quan hệ

Giả thiết:  $R(A_1, A_2, \dots, A_n)$  và  $S(B_1, B_2, \dots, B_m)$ , việc ghép bộ  $t = (a_1, a_2, \dots, a_n) \in R$  vào  $v = (b_1, b_2, \dots, b_m) \in S$  được định nghĩa như sau:

$A \in R^+$  và  $B \in S^+$  là hai thuộc tính có thể so sánh được.

Gọi  $\theta$  là một trong các phép so sánh  $\{<, >, =, \geq, \leq, \neq\}$

Khi đó, phép kết nối hai quan hệ R và S trên các thuộc tính A, B với phép so sánh  $\theta$  được định nghĩa:

$$R \bowtie_{A\theta B} S = \{v = (t, u) \mid t \in R, u \in S, t.A \theta u.B\}$$

Có thể nhận thấy rằng phép kết nối được thực hiện qua 2 bước: (1) tích đề các hai quan hệ R và S, (2) chọn các bộ thỏa điều kiện  $A\theta B$ .

Nếu  $\theta$  là phép toán so sánh bằng nhau được gọi là phép *kết nối bằng* (equi join). Nếu các thuộc tính so sánh là giống tên nhau thì trong kết quả của phép nối sẽ loại đi một cột (thuộc tính), khi đó phép kết được gọi là phép *kết tự nhiên* (natural join) và về mặt ký hiệu bỏ đi  $A\theta B$ . Các trường hợp còn lại được gọi là phép kết nối *theta* ( $\theta$  join)

Ví dụ:

A	B	C			
a1	1	1			
a2	2	1			
a3	2	2			

R

C	D	E			
1	d1	e1			
2	d2	e2			
3	d3	e3			

S

A	B	C	C	D	E
a1	1	1	1	d1	E1
a2	2	1	1	d1	E1
A2	2	1	2	D2	E2
A3	2	2	1	d1	E1
a3	2	2	2	d2	E2

$R \bowtie_{R.B \geq S.C} S$

Và kết quả của phép nối tự nhiên (kết bằng trên thuộc tính C) là

A	B	C	D	E
a1	1	1	d1	e1
a2	2	1	d1	e1
a3	2	2	d2	e2

## 2.2. Phép kết nối nội (inner join)

Thực chất của phép kết nối nội là phép kết nối bằng đã nêu trên. Tuy nhiên, ngay cả khi hai thuộc tính có cùng tên thì kết quả vẫn giữ lại 2 tên thuộc tính đó.

Ví dụ:

A	B	C	A	D	E	
a1	B 1	c1	a1	d1	e1	
a2	B 2	c1	a2	d2	e2	
a3	B 3	c2	a4	d4	e4	
a5	B 5	c5	a6	d6	e6	
a7	B 7	c7	a7	d7	e7	

A	D	E
a1	d1	e1
a2	d2	e2
a4	d4	e4
a6	d6	e6
a7	d7	e7

R

A	B	C	A	D	E	
a1	b1	c1	a1	d1	E 1	
a2	b2	c1	a2	d2	E 2	
a7	b7	c7	a7	d7	E 7	

S

$$R \underset{A}{\overset{R.A=S.}{\bowtie}} S$$

## 2.3. Phép kết nối trái (left join)

Phép kết nối trái hai quan hệ R và S trên các thuộc tính A và B với phép so sánh bằng được định nghĩa là tất cả các bộ v đạt được bằng cách xếp bộ giá trị của R và S cạnh nhau, nếu có giá trị giống nhau trên hai thuộc tính kết nối; và các bộ v đạt được nhờ cách đặt bộ R với các bộ null của S, nếu không tìm được giá trị tương ứng của thuộc tính kết nối trên quan hệ S.

$$R.A=S.B$$

$$R \ltimes S = \{v = (t, u) \mid (t \in R, u \in S : t.A \theta u.B) \vee (t \in R, u = u_{null} : t.A \notin S[B])\}$$

Với  $S[B]$  là tập tất cả các giá trị của thuộc tính B của S

Ví dụ:



A	B	C	A	D	E	A	B	C	A	D	E
a1	b1	c1	a1	d1	e1	a1	b1	c1	a1	d1	e1
a2	b2	c1	a2	d2	e2	a2	b2	c1	a2	d2	e2
a3	b3	c2	a4	d4	e4	a3	b3	c2	null	null	null
a5	b5	c5	a6	d6	e6	a5	b5	c5	null	null	null
a7	b7	c7	a7	d7	e7	a7	b7	c7	a7	d7	e7

R

S

$R \bowtie_{R.A=S.A} S$

Ý nghĩa của phép kết nối này là xác định được các bộ giá trị của quan hệ bên trái nhưng không có bộ giá trị tương ứng trong quan hệ bên phải.

2.4. Phép kết nối phải (right join)

Phép kết nối phải hai quan hệ R và S trên các thuộc tính A và B với phép so sánh bằng được định nghĩa là tất cả các bộ v đạt được bằng cách xếp bộ giá trị của R và S cạnh nhau, nếu có giá trị giống nhau trên hai thuộc tính kết nối; và các bộ v đạt được nhờ cách đặt bộ null của R với các bộ của S, nếu không tìm được giá trị tương ứng của thuộc tính kết nối trên quan hệ R.

$R.A=S.B$

$$R \Join S = \{v = (t,u) \mid (t \in R, u \in S : t.A \theta u.B) \vee (t = t_{null}, u \in S, : u.B \notin R[A])\}$$

Với  $R[A]$  là tập tất cả các giá trị của thuộc tính A của R

Ví dụ:

A	B	C	A	D	E	A	B	C	A	D	E
a1	b1	c1	a1	d1	e1	a1	b1	c1	a1	d1	e1
a2	b2	c1	a2	d2	e2	a2	b2	c1	a2	d2	e2
a3	b3	c2	a4	d4	e4	null	null	Null	a4	d4	e4
a5	b5	c5	a6	d6	e6	null	null	Null	a6	d6	e6
a7	b7	c7	a7	d7	e7	a7	b7	c7	a7	d7	e7

R

S

$R \Join_{R.A=S.A} S$

Ý nghĩa của phép kết nối này là xác định được các bộ giá trị của quan hệ bên phải nhưng không có bộ giá trị tương ứng trong quan hệ bên trái.

Chú ý rằng trong một số ngôn ngữ truy vấn CSDL, người ta gọi hai phép kết nối trái và phải chung lại là phép *kết nối ngoài* (outer join) vì cho phép giữ lại tất cả các bộ của hai quan hệ không tìm được bộ giá trị giống nhau trên các thuộc tính kết nối.

## 2.5. Hàm kết hợp và gom nhóm

Dùng để tính toán các giá trị mang tính chất tổng hợp trong đại số quan hệ. Trong đó:

**Hàm kết hợp:** đầu vào là một tập giá trị và trả về một giá trị đơn

- Avg(): giá trị trung bình
- Min(): giá trị nhỏ nhất
- Max(): giá trị lớn nhất
- Sum(): tính tổng
- Count(): đếm số mẫu tin

**Gom nhóm:** công thức như sau:  $\mathfrak{F}_{G_1, G_2, \dots, G_N, F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(E)$ , với:

- E là biểu thức đại số quan hệ
- $G_i$  là tên thuộc tính gom nhóm (có thể không có)
- $F_i$  là hàm gom nhóm
- $A_i$  là tên thuộc tính tính toán trong hàm gom nhóm

Ví dụ:

- Cho biết số nhân viên trong công ty và mức lương trung bình

$$\mathfrak{F}_{COUNT(MaNV), AVG(Luong)}(NHANVIEN)$$

- Cho biết số lượng nhân viên và lương trung bình của mỗi phòng ban

$$Phong \mathfrak{F}_{COUNT(MaNV), AVG(Luong)}(NHANVIEN)$$

## 2.6. Các phép toán cập nhật trên quan hệ

Các thao tác được viết thông qua phép toán gán. Cụ thể như sau:

### 2.6.1. Thêm

Phép thêm:  $r \leftarrow r \cup E$ , với  $r$  là một quan hệ và  $E$  là một biểu thức đại số quan hệ.

Thông thường, đưa ra bộ cần chèn một cách tường minh hoặc viết một câu truy vấn mà kết quả truy vấn chính là một tập các bộ cần chèn.

Ví dụ: Chèn một bộ tường minh

$$DEAN \leftarrow DEAN \cup \{ 'DA01', 'Phổ cập tin học', 'Đức Trọng', 4 \}$$

### 2.6.2. Xóa

Phép xóa:  $r \leftarrow r - E$ , với  $r$  là một quan hệ và  $E$  là một biểu thức đại số quan hệ. Chú ý rằng phép xóa thực hiện xóa một hoặc nhiều bộ mà không thể xóa đi giá trị của các thuộc tính.

Ví dụ:

- Xóa tất cả các phân công công tác tham gia đề án của nhân viên mang mã số NV01

$$PHANCONG \leftarrow PHANCONG - (\sigma_{MaNV='NV01'}(NHANVIEN))$$

- Xóa tất cả các đề án do phòng mang tên ‘Quản Lý’ chủ trì

$$r1 \leftarrow (\sigma_{TenPhong='Quản Lý'}(PHONGBAN \bowtie_{MaPhong=Phong} DEAN))$$

$$r2 \leftarrow \pi_{MaDA, TenDA, DDiemDA, Phong}(r1)$$

$$DEAN \leftarrow DEAN - r2$$

### 2.6.3. Sửa

Phép sửa:  $r \leftarrow \pi_{F1, F2, \dots, Fn}(r)$ , với:

- $F_i$  là một biểu thức, gồm hằng và thuộc tính của  $r$  để đưa ra giá trị mới cho thuộc tính này.
- Mỗi  $F_i$  có giá trị trả về là giá trị mới cho thuộc tính thứ  $i$  của  $r$ , thuộc tính này có thể được giữ nguyên hoặc cập nhật với giá trị mới.

Phép sửa có thể được viết thông qua phép xóa và thêm. Khi đó, phép xóa sẽ xóa đi các bộ chứa giá trị cũ và phép thêm sẽ thêm những bộ chứa giá trị mới.

Ví dụ:

- Cộng thêm lương mỗi nhân viên với số tiền là 120000

$$NHANVIEN \leftarrow (\pi_{MaNV, HoNV, TenNV, NgaySinh, DiaChi, Phai, Luong+120000, MaNQL, Phong}(NHANVIEN))$$

- Với nhân viên nam, cộng thêm lương với số tiền là 100000; với nhân viên nữ, cộng thêm lương với số tiền là 150000

$$NHANVIEN \leftarrow \left( \pi_{MaNV, HoNV, TenNV, NgaySinh, DiaChi, Phai, Luong+100000, MaNQL, Phong} (\sigma_{Phai='Nam'}(NHANVIEN)) \right) \cup \left( \pi_{MaNV, HoNV, TenNV, NgaySinh, DiaChi, Phai, Luong+150000, MaNQL, Phong} (\sigma_{Phai='Nu'}(NHANVIEN)) \right)$$

### 3. Bài tập

#### Bài 1:

Với lược đồ cơ sở dữ liệu Quản lý đề án trong 2.2. Hãy viết các biểu thức đại số quan hệ theo yêu cầu:

1. Cho biết thông tin cá nhân về những nhân viên có tên ‘Mai’
2. Tìm mã nhân viên, họ tên và địa chỉ của tất cả nhân viên làm việc phòng ‘Hành Chính’
3. Tìm mã nhân viên, họ tên và địa chỉ của tất cả nhân viên làm việc phòng ‘Hành Chính’ và ‘Tài Vụ’
4. Cho biết mã nhân viên, họ tên nhân viên và tên các đề án mà nhân viên tham gia.
5. Tìm mã đề án, tên đề án, tên phòng ban chủ trì đề án cùng mã trưởng phòng, tên trưởng phòng đó.
6. Cho biết mã nhân viên, họ tên của những nhân viên tham gia vào đề án có mã là ‘DA01’ và có thời gian làm việc cho đề án trên 30giờ/tuần
7. Cho biết mã nhân viên, họ tên của những nhân viên có cùng tên với người thân.
8. Cho biết mã nhân viên, họ tên của những nhân viên có người trưởng phòng có họ tên là ‘Nguyễn’ ‘Mai’
9. Cho biết mã nhân viên, họ tên của những nhân viên có người quản lý có họ tên là ‘Nguyễn’ ‘Mai’
10. Cho biết mã nhân viên, họ tên của những nhân viên tham gia mọi đề án của công ty.
11. Cho biết mã nhân viên, họ tên của những nhân viên không tham gia đề án nào của công ty.

12. Cho biết mức lương trung bình của nhân viên trong công ty.
13. Cho biết mức lương trung bình của nhân viên nam trong công ty.
14. Cho biết tổng số đề án của công ty.
15. Với mỗi đề án, cho biết tổng số nhân viên tham gia vào đề án.
16. Với mỗi đề án, cho biết tổng số nhân viên nữ tham gia vào đề án.
17. Tăng thời gian tham gia đề án của các nhân viên nam thêm 4giờ/tuần
18. Xóa tất cả những nhân viên có mức lương dưới 500000

## **Bài 2:**

Cho lược đồ CSDL Quản lý sinh viên sau:

SINHVIEN (MaSV, HoSV, TenSV, NgaySinh, DiaChi, Phai, Nam, Khoa)

*Tân từ:* Mỗi sinh viên có Mã sinh viên (MaSV) duy nhất để phân biệt với các sinh viên khác, có họ tên (HoSV, TenSV), ngày sinh (NgaySinh), địa chỉ (DiaChi), phái Nam hoặc Nữ (Phai), năm nhập học (Nam) và thuộc về một khoa (Khoa)

GIANGVIEN (MaGV, HoGV, TenGV, NgaySinh, DiaChi, Phai, ChuyenNganh, Khoa)

*Tân từ:* Mỗi giảng viên có Mã giảng viên (MaGV) duy nhất để phân biệt với các giảng viên khác, có họ tên (HoGV, TenGV), ngày sinh (NgaySinh), địa chỉ (DiaChi), phái Nam hoặc Nữ (Phai), chuyên ngành (ChuyenNganh) và thuộc về một khoa (Khoa)

MONHOC (MaMH, TenMH, STC, Loai, Khoa)

*Tân từ:* Mỗi môn học có mã môn học (MaMH) duy nhất để phân biệt với các môn học khác, có tên môn học (TenMH), số tín chỉ (STC), là loại bắt buộc hay tự chọn (Loai), và do một khoa (Khoa) chịu trách nhiệm giảng dạy.

DIEUKIEN (MaMH, MaMHTruoc)

*Tân từ:* Một số môn học có điều kiện tiên quyết, sinh viên muốn học môn học (MaMH) thì phải đạt được môn tiên quyết của môn học này (MaMHTruoc)

KHOAHOC (MaKH, MaMH, HocKy, NamHoc, MaGV)

*Tân từ:* Một môn học (MaMH) được tổ chức trong học kỳ (HocKy) của một năm học (NamHoc) và do một giảng viên chịu trách nhiệm giảng dạy (MaGV). Lưu ý rằng một môn học có thể được mở nhiều lần (chẳng hạn năm học 2007-2008 mở cho khoá CTK29, năm học 2008-2009 mở cho khoá CTK30). Trong quan hệ này, mã khoá học (MaKH) thể hiện việc một lần tổ chức giảng dạy môn học.

KETQUA (MaSV, MaKH, Diem, KetQua)

*Tân từ:* Khi sinh viên (MaSV) tham gia học môn học tại một khoá học (MaKH) sẽ có điểm đánh giá (Diem) của học viên, từ điểm đánh giá sẽ có kết quả (KetQua) đạt hay không đạt.

Hãy viết các biểu thức đại số quan hệ theo yêu cầu:

1. Cho biết mã sinh viên, họ tên của mọi sinh viên
  2. Cho biết mã môn học, tên môn học và số tín chỉ tương ứng
  3. Cho biết mã môn học, tên môn học phải học trước môn có mã là 'CT101'
  4. Cho biết mã sinh viên, họ tên sinh viên cùng với các môn học mà sinh viên đạt trên 5 điểm.
  5. Cho biết mã sinh viên, họ tên sinh viên học tất cả các khóa học.
  6. Cho biết tổng số sinh viên của mỗi khoa.
  7. Cho biết mã sinh viên, họ tên sinh viên đạt điểm cao nhất trong mỗi khóa học
  8. Cho biết mã sinh viên, họ tên sinh viên và điểm trung bình của sinh viên trong từng học kỳ của từng niên học
  9. Cho biết mã giáo viên, họ tên giáo viên và chuyên ngành của những giáo viên tham gia dạy năm 2004-2005
  10. Tăng số tín chỉ cho những môn học được học trong học kỳ 1, năm 2004-2005
  11. Từ điểm của sinh viên, hãy điền vào cột KetQua thỏa: nếu điểm $\geq$ 5: đạt, ngược lại: không đạt.
-

# Ngôn Ngữ Tân Từ

---

Trong CSDL thực hiện việc mô hình hóa thông tin gồm các sự kiện được liên kết hay biểu diễn một tình trạng thế giới thực. Ngôn ngữ tân từ ứng dụng logic toán để thể hiện truy vấn. Ngôn ngữ tân từ có hai loại, ngôn ngữ tân từ có biến là bộ và ngôn ngữ tân từ có biến là miền giá trị.

## 1. Ngôn ngữ tân từ có biến là bộ

### 1.1. Một số khái niệm

Dạng thức:  $\{t \mid P(t)\}$ , với:

- $t$ : biến bộ, nhận giá trị là một bộ của quan hệ. Khi đó  $t.A$  là giá trị của bộ  $t$  tại thuộc tính  $A$ .
- $P(t)$ : công thức liên quan đến biến bộ  $t$ , phụ thuộc vào giá trị của  $t$  mà  $P(t)$  cho kết quả đúng hay sai.
- Kết quả trả về là tập hợp các bộ  $t$  thỏa  $P(t)$

Ví dụ:

- Tìm tất cả những nhân viên nam

$$\{t \mid NHANVIEN(t) \wedge t.Phai = 'Nam'\}$$

- Tìm tất cả những nhân viên nữ và có lương trên 1500000

$$\{t \mid NHANVIEN(t) \wedge t.Phai = 'Nữ' \wedge t.Luong > 1500000\}$$

Với mỗi biến bộ  $t$ , quan hệ  $R$  mà  $t$  biến thiên trên đó được gọi là quan hệ vùng của biến bộ. Giá trị này được chỉ định bằng điều kiện dạng  $R(t)$ .

### 1.2. Định nghĩa hình thức của phép tính bộ

Dạng tổng quát:  $\{t_1.A_1, t_2.A_2, \dots, t_n.A_n \mid P(t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m})\}$ , với:

- $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$  là các biến bộ
- $A_i$  là thuộc tính của quan hệ mà biến bộ  $t_i$  biến thiên
- $P$  là công thức hay điều kiện của phép tính bộ.
- Công thức được hình thành từ các công thức nguyên tố

### Công thức nguyên tố:

Có 3 dạng:

- $R(t_i)$  là công thức nguyên tố, với
  - $R$  là một quan hệ và  $t_i$  là một biến bộ
  - Công thức này xác định vùng của biến bộ  $t_i$  dưới hình thức quan hệ có tên  $R$
  - Ví dụ: DEAN( $t$ ), NHANVIEN( $x$ )
- $t_i.A \theta t_j.B$  là công thức nguyên tố, với:
  - $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $t_i, t_j$  là các biến bộ
  - $A$  là thuộc tính của quan hệ trên đó bộ  $t_i$  biến thiên,  $B$  là thuộc tính của quan hệ trên đó bộ  $t_j$  biến thiên
  - Ví dụ:  $x.MaPhong = y.Phong$ , với PHONGBAN( $x$ ) và DEAN( $y$ )
- $t_i.A \theta c$  hoặc  $c \theta t_j.B$  là công thức nguyên tố, với:
  - $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $t_i, t_j$  là các biến bộ
  - $A$  là thuộc tính của quan hệ trên đó bộ  $t_i$  biến thiên,  $B$  là thuộc tính của quan hệ trên đó bộ  $t_j$  biến thiên
  - $c$  là giá trị hằng
  - Ví dụ:  $x.MaPhong = PH01$ ,  $y.Luong > 500000$

Nhận thấy rằng mỗi công thức nguyên tố đều mang giá trị đúng hoặc sai. Với (a),  $R(t)$  đúng nếu  $t$  là một bộ thuộc  $R$ , ngược lại mang giá trị sai; với (b, c) giá trị đúng sai phụ thuộc vào kết quả thay thế giá trị thực sự của bộ vào vị trí biến bộ.



Một công thức (hay điều kiện) được hình thành từ một hay nhiều công thức nguyên tố và được nối với nhau bằng các toán tử và, hoặc, phủ định ( $\wedge, \vee, \neg$ ) và được định nghĩa như sau:

- a. Mọi công thức nguyên tố là công thức
- b. Nếu  $F_1$  và  $F_2$  là các công thức thì  $(F_1 \wedge F_2), (F_1 \vee F_2), \neg F_1, \neg F_2$  là công thức và có giá trị:
  - $(F_1 \wedge F_2)$  chỉ đúng khi cả  $F_1$  và  $F_2$  đều đúng
  - $(F_1 \vee F_2)$  chỉ sai khi cả  $F_1$  và  $F_2$  đều sai
  - $\neg F_1$  đúng nếu  $F_1$  sai,  $\neg F_1$  sai nếu  $F_1$  đúng

### 1.3. Lượng từ tồn tại $\exists$ và với mọi $\forall$

#### Biến bộ tự do và biến kết buộc

Một biến bộ  $t$  là *kết buộc* nếu có kèm lượng từ, nghĩa là nó xuất hiện trong mệnh đề  $\forall t \in S(P(t))$  hay  $\exists t \in S(P(t))$ , ngược lại nó được gọi là *biến tự do*.

Nếu  $F$  là công thức nguyên tố thì mọi biến bộ  $t$  trong  $F$  đều là biến tự do.

Tất cả các biến bộ tự do  $t$  trong  $F$  được xem là biến kết buộc trong công thức  $F' = (\forall t)F$  hay  $F' = (\exists t)F$

Đối với các công thức  $F = (F_1 \wedge F_2), F = (F_1 \vee F_2), F = \neg F_1, F = \neg F_2$ , biến  $t$  là tự do hay kết buộc phụ thuộc vào nó là tự do hay kết buộc trong  $F_1$  và  $F_2$

Biến bộ tự do chỉ ra các bộ mà câu truy vấn trả về, nghĩa là được sử dụng trong vế trái. Ngược lại biến bộ kết buộc thường được sử dụng để thực hiện việc khẳng định các bộ trong CSDL, nghĩa là được sử dụng trong vế phải.

Ví dụ:

- Cho biết những nhân viên thuộc phòng có mã số 5 và có lương  $\geq 500000$

$$\{n \mid NHANVIEN(n) \wedge n.Phong = 5 \wedge n.Luong \geq 500000\}$$

- Cho biết những đề án do phòng ‘Quản Lý’ phụ trách

$$\{d \mid DEAN(d) \wedge (\exists p \in PHONGBAN(d.Phong = p.MaPhong \wedge p.TenPhong = 'QuanLy'))\}$$

- Tìm những nhân viên tham gia trong tất cả các đề án của công ty

$$\{n \mid NHANVIEN(n) \wedge (\forall d \in DEAN(\exists p \in PHANCONG(p.MaDA = d.MaDA \wedge p.MaNV = n.MaNV))))\}$$

Chú ý rằng

- $\forall t(F)$  đúng nếu F đúng với mọi bộ t, sai nếu ít nhất một bộ sai.
- $\exists t(F)$  sai nếu F sai với mọi bộ t, đúng nếu ít nhất một bộ đúng.

## 2. Ngôn ngữ tân từ có biến là miền giá trị

Biến miền giá trị nhận giá trị từ miền giá trị của một thuộc tính.

Dạng thức:  $\{x_1, x_2, \dots, x_n \mid P\}$ , với:

- $x_1, x_2, \dots, x_n$  là danh sách các biến miền giá trị
- P là điều kiện hay công thức theo  $x_1, x_2, \dots, x_n$
- Kết quả câu truy vấn là tập hợp các chọn lựa của các bộ  $x_1, x_2, \dots, x_n$  sao cho, với mọi i, giá trị  $x_i$  được thay thế cho các biến tự do  $x_i$  thì điều kiện đúng. Biến  $x_i$  có thể là hằng số, khi đó tất cả các bộ trong tập hợp đều là hằng số trong vị trí i.

Ví dụ:

- Cho biết mã nhân viên thuộc phòng có mã số 5 và có lương  $\geq 500000$

$$\left\{ Ma, Ho, Ten \mid \exists Phg \exists Lg \left( \begin{array}{l} NHANVIEN(Ma, Ho, Ten, NSinh, DChi, Phai, Lg, MaQL, Phg) \\ \wedge Phg = 5 \wedge Lg \geq 500000 \end{array} \right) \right\}$$

Nhận thấy rằng  $Ma, Ho, Ten$  chính là thuộc tính được yêu cầu;  $Phg, Lg$  là biến thực sự xuất hiện trong một câu điều kiện.

- Tìm họ tên, địa chỉ của nhân viên thuộc phòng ‘Quản Lý’

$$\left\{ Ma, Ho, Ten, DChi \mid \exists Phg \left( \begin{array}{l} NHANVIEN(Ma, Ho, Ten, NSinh, DChi, Phai, Lg, MaQL, Phg) \\ \wedge \exists TenPhg (MaPhg, TenPhg, TrPhg, NgayNC) \wedge TenPhg = 'Quản Lý' \end{array} \right) \right\}$$

## Công thức nguyên tố:

Có 3 dạng:

- a.  $R(x_1, x_2, \dots, x_j)$  là công thức nguyên tố, với
- $R$  là một quan hệ bậc  $j$
  - Mỗi  $x_i, 1 \leq i \leq n$  là một biến miền.
  - Công thức này cho biết  $(x_1, x_2, \dots, x_j)$  là một bộ của quan hệ  $R$ , với  $x_i$  là giá trị thứ  $i$  của bộ.
  - Ví dụ:  $\{x_1, x_2, \dots, x_n \mid R(x_1, x_2) \wedge \dots\}$
- b.  $x_i \theta x_j$  là công thức nguyên tố, với:
- $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $x_i, x_j$  là các biến miền
  - Ví dụ:  $\text{MaPhg} = \text{Phg}$ .
- c.  $x_i \theta c$  hoặc  $c \theta x_j$  là công thức nguyên tố, với:
- $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $x_i, x_j$  là các biến miền
  - $c$  là hằng
  - Ví dụ:  $\text{MaPhg} = 1$ .

Một công thức nguyên tố có trị đúng hoặc sai với một tập giá trị cụ thể tương ứng với các biến miền.

Các định nghĩa về công thức dựa trên công thức nguyên tố, các định nghĩa về biến kết buộc và tự do, các lượng từ trong trường hợp phép tính miền cũng tương tự phép tính bộ.

### 3. Bài tập

#### Bài 1:

Hãy viết bằng ngôn ngữ tân từ theo yêu cầu như bài tập 1 trong chương 4.

#### Bài 2:

Hãy viết bằng ngôn ngữ tân từ theo yêu cầu như bài tập 2 trong chương 4.

# ***Ngôn Ngữ Truy Vấn SQL***

---

SQL (Structured Query Language) là ngôn ngữ CSDL quan hệ chuẩn dành cho các CSDL quan hệ thương mại.

Từ những năm 70, SQL được phát triển từ IBM với hệ quản trị CSDL quan hệ SYSTEM-R với ngôn ngữ giao tiếp CSDL là Sequel (Structured English Query Language).

Năm 1986, phiên bản chuẩn được chấp nhận bởi ANSI (American National Standards Institute) gọi là SQL-86 hay SQL1 và năm 1987 được chấp nhận bởi ISO (International Standards Organization). Từ đó các phiên bản đã được đưa ra vào 1992, 1999, 2003.

## **1. Các lệnh hỏi**

Các lệnh hỏi hay còn được gọi là truy vấn rút trích dữ liệu. Lệnh SELECT là lệnh cơ bản để rút trích thông tin từ CSDL.

Chú ý rằng lệnh SELECT không hoàn toàn giống như phép toán chọn trong đại số quan hệ, SQL cho phép một bảng có hai hay nhiều bộ có giá trị giống nhau trên mọi thuộc tính cùng tồn tại

### **1.1. Cú pháp lệnh truy vấn**

Cú pháp lệnh truy vấn:

```
SELECT <danh sách thuộc tính>
FROM <danh sách các bảng>
WHERE <điều kiện>
GROUP BY <các thuộc tính gom nhóm>
HAVING <điều kiện gom nhóm>
ORDER BY <danh sách thuộc tính>
```

với:

- **SELECT** <danh sách thuộc tính>: tên các thuộc tính được lấy giá trị
- **FROM** <danh sách các bảng>: tên các bảng cần để xử lý câu truy vấn
- **WHERE** <điều kiện>: biểu thức điều kiện chọn và điều kiện kết các bộ trong các quan hệ được chỉ ra trong mệnh đề FROM
- **GROUP BY** <các thuộc tính gom nhóm>: chỉ ra các thuộc tính gom nhóm
- **HAVING** <điều kiện gom nhóm>: chỉ ra điều kiện để trích chọn các nhóm
- **ORDER BY** <danh sách thuộc tính>: thứ tự hiển thị kết quả câu truy vấn

Trong câu lệnh, **SELECT** và **FROM** là bắt buộc, những thành phần còn lại có thể không có.

## 1.2. Phép chiếu

Sử dụng mệnh đề **SELECT**, kết quả gần giống với phép chiếu của đại số quan hệ, liệt kê các thuộc tính cần hiển thị trong kết quả câu truy vấn.

Ví dụ:

- Tìm mã nhân viên, họ tên tất cả các nhân viên

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
```

- Tìm mã đề án, tên các đề án

```
SELECT MaDA, TenDA
FROM DEAN
```

## 1.3. Phép chọn

Sử dụng mệnh đề **WHERE**, kết quả gần giống với phép chọn của đại số quan hệ, nêu điều kiện liên quan đến thuộc tính của quan hệ xuất hiện trong mệnh đề FROM. Thường sử dụng **AND**, **OR**, **NOT**, **BETWEEN**, các phép toán so sánh.

Ví dụ:

- Tìm mã nhân viên, họ tên tất cả các nhân viên nam

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
```

WHERE Phai='Nam'

#### 1.4. Phép kết

Trong mệnh đề WHERE thường có điều kiện kết nếu như trong mệnh đề FROM có nhiều hơn hai quan hệ.

Ví dụ:

- Tìm mã nhân viên, họ tên tất cả các nhân viên tham gia vào đề án có mã 'DA01'

```
SELECT MaNV, HoNV, TenNV
```

```
FROM NHANVIEN, PHANCONG
```

```
WHERE NHANVIEN.MaNV= PHANCONG.MaNV AND MaDA='DA01'
```

- Tìm mã nhân viên, họ tên tất cả các nhân viên làm việc tại phòng 'Tài chính'

```
SELECT MaNV, HoNV, TenNV
```

```
FROM NHANVIEN, PHONGBAN
```

```
WHERE Phong=MaPhong AND TenPhong = 'Tài chính'
```

#### 1.5. Một số lưu ý

##### Sử dụng \*

Khi cần lấy thông tin về tất cả các cột của bảng thì sử dụng dấu sao (\*) thay vì phải liệt kê mọi tên thuộc tính.

Ví dụ:

```
SELECT *
```

```
FROM NHANVIEN
```

##### Tên bí danh

Khi câu truy vấn cần tham chiếu tới cùng một quan hệ 2 lần thì dùng bí danh cho tên quan hệ. SQL cho phép đổi tên quan hệ và tên thuộc tính được thực hiện bằng AS (có thể dùng hoặc không) theo quy tắc: *Tên cũ [as] tên mới*

Ví dụ:

```
SELECT MaNV, HoNV as Ho, TenNV as Ten
```

```
FROM NHANVIEN
```

WHERE Phai='Nam'

### **Biểu thức trong mệnh đề SELECT**

Tên thuộc tính có thể kèm theo tên bảng nếu cần làm rõ bằng cách thêm dấu chấm (.) trước tên thuộc tính. Ví dụ: NHANVIEN.MaNV.

Câu lệnh SELECT còn cho phép thực hiện tính toán theo công thức dựa trên các cột của bảng.

Ví dụ:

- Tìm mã nhân viên, họ tên và lương nhân viên, trong đó lương được tăng thêm 20% cho mọi nhân viên

```
SELECT MaNV, HoNV, TenNV, Luong*1.2  
FROM NHANVIEN
```

### **DISTINCT**

Như đã trình bày ở trên, SQL cho phép các bộ trùng nhau trong kết quả câu truy vấn. Để loại bỏ các bộ trùng nhau, sử dụng từ khóa DISTINCT sau SELECT.

Ví dụ:

- Cho biết những mức lương hiện có trong công ty

```
SELECT DISTINCT Luong  
FROM NHANVIEN
```

### **Phép so sánh trên chuỗi**

Sử dụng LIKE để so sánh chuỗi, có 2 ký tự đặc biệt sau:

- % (hoặc \*): thay thế cho mọi ký tự bất kỳ
- \_ (hoặc ?) : thay thế cho một ký tự bất kỳ

Ví dụ:

- Tìm tất cả những nhân viên có tên bắt đầu bằng M như Mai, Minh,...

```
SELECT MaNV, HoNV, TenNV  
FROM NHANVIEN  
WHERE TenNV LIKE 'M%'
```

## Thứ tự hiển thị

Mệnh đề ORDER BY sử dụng để sắp xếp các bộ trong kết quả câu truy vấn dựa trên giá trị các thuộc tính. Trong đó:

- Asc: sắp theo thứ tự tăng dần (mặc định)
- Desc: sắp theo thứ tự giảm dần

Ví dụ:

- Tìm tất cả những nhân viên phòng 1 và lương tương ứng, sắp xếp giảm dần theo lương

```
SELECT MaNV, HoNV, TenNV, Luong
FROM NHANVIEN
WHERE Phong = 1
ORDER BY Luong desc
```

## 2. Câu truy vấn lồng

Ở các ví dụ đã trình bày, mệnh đề WHERE bao gồm thuộc tính đơn, các phép so sánh hằng số. Phần này giới thiệu câu truy vấn lồng, cho phép lồng các câu truy vấn lại với nhau. Khi đó câu truy vấn con thông thường được sử dụng để kiểm tra các tập hợp thành viên, so sánh tập hợp, kiểm tra sự tồn tại.

Khi sử dụng truy vấn con trong mệnh đề WHERE của một truy vấn khác, mệnh đề SELECT trong truy vấn con phải phù hợp với số thuộc tính và kiểu dữ liệu của mệnh đề WHERE trong câu truy vấn ngoài.

Truy vấn con trả về giá trị tập hợp trong mệnh đề WHERE có dạng:

- <biểu thức> [NOT] IN (<truy vấn con>)
- <biểu thức> <phép toán so sánh> ANY (<truy vấn con>)
- <biểu thức> <phép toán so sánh> ALL (<truy vấn con>)

Kiểm tra sự tồn tại:

- [NOT] EXISTS (<truy vấn con>)

Các câu truy vấn con trong một mệnh đề WHERE có thể được kết hợp bằng cách sử dụng các phép nối logic



Ví dụ:

- Tìm tất cả những nhân viên nam làm việc trong phòng ‘Tài chính’

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
WHERE Phong IN (SELECT MaPhong as Phong
                FROM PHONGBAN
                WHERE TenPhong = ‘Tài chính’)
AND Phai = ‘Nam’
```

Trong trường hợp điều kiện ở mệnh đề WHERE của câu truy vấn con tham chiếu tới một thuộc tính của quan hệ được khai báo trong truy vấn cha thì hai câu truy vấn được gọi là tương quan. Các tham chiếu tới các quan hệ và các thuộc tính cha xuất hiện thông qua việc sử dụng bí danh.

Ví dụ:

- Tìm tất cả những nhân viên có cùng tên với người thân

```
SELECT nv.MaNV, nv.HoNV, nv.TenNV
FROM NHANVIEN as nv
WHERE nv.MaNV IN (SELECT MaNV
                  FROM THANNHAN
                  WHERE TenTN = nv.TenNV)
```

- Tìm tất cả những nhân viên có lương lớn hơn mọi nhân viên phòng số 5

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
WHERE Luong > ALL (SELECT Luong
                   FROM NHANVIEN
                   WHERE Phong = 5)
```

- Tìm tất cả những nhân viên có lương lớn hơn ít nhất 1 nhân viên phòng số 5

```
SELECT nv1.MaNV, nv1.HoNV, nv1.TenNV
FROM NHANVIEN as nv1, NHANVIEN as nv2
WHERE nv1.Luong > nv2.Luong AND nv2.Phong = 5
```

Cụm từ lớn hơn ít nhất một có thể diễn đạt bằng >SOME. Ví dụ trên được viết lại như sau:

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
WHERE Luong > SOME (SELECT Luong
                     FROM NHANVIEN
                     WHERE Phong = 5)
```

Hàm EXISTS sử dụng để kiểm tra kết quả của câu truy vấn tương quan có rỗng hay không. Khi đó

- EXISTS (r) mang giá trị true nếu có ít nhất một bộ trong r, false nếu ngược lại.
- NOT EXISTS (r) mang giá trị true nếu không có bộ nào trong r, false nếu ngược lại.

Ví dụ

- Cho biết những nhân viên có cùng tên với thân nhân

```
SELECT nv.MaNV, nv.HoNV, nv.TenNV
FROM NHANVIEN nv
WHERE EXISTS (SELECT *
              FROM THANNHAN
              WHERE MaNV = nv.MaNV AND TenTN= nv.TenNV)
```

### Giá trị null

SQL cho phép sử dụng các giá trị null để chỉ ra giá trị của thuộc tính không biết hay không tồn tại.

Chú ý rằng kết quả của điều kiện ở mệnh đề WHERE là false nếu nó liên quan đến null.

Trong SQL sử dụng IS NULL hay IS NOT NULL để kiểm tra các giá trị rỗng. Trong đó:

- Phép so sánh bằng (=) không dùng được
- Các hàm gom nhóm ngoại trừ COUNT() bỏ qua các giá trị null trong tập các dữ liệu đầu vào
- Nếu có điều kiện kết thì các bộ có giá trị null trên thuộc tính kết sẽ không có trong kết quả.

### 3. Hàm kết hợp và gom nhóm

#### Hàm kết hợp

Hàm kết hợp có đầu vào là một tập giá trị và trả về một giá trị đơn

- Avg(): giá trị trung bình
- Min(): giá trị nhỏ nhất
- Max(): giá trị lớn nhất
- Sum(): tính tổng
- Count(): đếm số mẫu tin

Ví dụ

- Cho biết những mức lương trung bình và cao nhất của các nhân viên phòng có mã là 5

```
SELECT AVG(Luong), MAX(Luong)
FROM NHANVIEN
WHERE Phong = 5
```

#### Gom nhóm

Các hàm gom nhóm được áp dụng trên các nhóm bộ cùng thuộc tính trong quan hệ. Mỗi nhóm bộ bao gồm tập hợp các bộ có cùng giá trị trên các thuộc tính gom nhóm. Trong SQL sử dụng cú pháp như sau:

```
SELECT <danh sách thuộc tính[với hàm kết hợp]>
FROM <danh sách các bảng>
WHERE <điều kiện>
GROUP BY <các thuộc tính gom nhóm>
```

HAVING <điều kiện gom nhóm>

- Mệnh đề GROUP BY chỉ ra các thuộc tính gom nhóm, các thuộc tính trong mệnh đề SELET nằm ngoài một hàm kết hợp phải xuất hiện trong mệnh đề GROUP BY
- Mệnh đề HAVING lấy các giá trị của hàm gom nhóm chỉ trên những nhóm nào thỏa điều kiện nhất định. Mệnh đề HAVING chỉ ra điều kiện lọc trên các nhóm, không phải điều kiện lọc trên từng bộ.

Ví dụ

- Cho biết tên phòng và lương trung bình của các nhân viên trong phòng ban lớn hơn 100000.

```
SELECT TenPhong, AVG(Luong)
FROM NHANVIEN, PHONGBAN
WHERE Phong = MaPhong
GROUP BY TenPhong
HAVING AVG(Luong)>100000
```

#### 4. Các lệnh khai báo cấu trúc CSDL

##### Kiểu dữ liệu

Chọn hệ quản trị CSDL SQL Server để minh họa, Ta có một số kiểu dữ liệu cơ bản:

- Số (number)
- Chuỗi (text, char, varchar, nvarchar)
- Ngày tháng (datetime)
- Logic (bit)

##### Tạo bảng

CREATE TABLE <tên bảng>

(

<thuộc tính 1> <kiểu dữ liệu> [not null][unique] [<RBTV thuộc tính>],  
<thuộc tính 2> <kiểu dữ liệu> [not null][unique] [<RBTV thuộc tính>],

```

...
<thuộc tính n> <kiểu dữ liệu> [not null][unique] [<RBTV thuộc tính>],
[<RBTV bảng>]
)

```

Các thuộc tính được xếp theo thứ tự khi tạo bảng. Các ràng buộc cũng có thể được bổ sung sau bằng cách dùng ALTER TABLE

Ví dụ

```

CREATE TABLE PHONGBAN
(
    MaPhong    char(5)      not null,
    TenPhong   nvarchar(30),
    TruongPhong char(5),
    NgayNhanChuc  datetime,
    PRIMARY KEY (MaPhong),
    UNIQUE (TenPhong)
    FOREIGN KEY (TruongPhong) REFERENCES NHANVIEN (MaNV)
)

```

## RBTV

Một số RBTV trong khi tạo bảng:

- Not null: không được chứa giá trị null
- Khóa chính: không được chứa giá trị null và được xác định bởi PRIMARY KEY <thuộc tính>
- Khóa ngoại: FOREIGN KEY <thuộc tính> REFERENCE <quan hệ> <thuộc tính>
- Tính duy nhất: có thể chứa giá trị null và được xác định bởi UNIQUE <thuộc tính>
- CHECK <điều kiện>: điều kiện đơn giản, không chứa các câu truy vấn hay tham chiếu tới các quan hệ khác

- Trong SQL server, mỗi RBTV có thể được đặt tên bằng cách sử dụng CONSTRAINT <tên ràng buộc> <kiểu ràng buộc>. Chú ý rằng tên ràng buộc phải duy nhất trong một lược đồ CSDL.

Ví dụ:

```
CREATE TABLE PHONGBAN
(
    MaPhong    char(5)        not null,
    TenPhong   nvarchar(30),
    TruongPhong char(5),
    NgayNhanChuc    datetime,
    CONSTRAINT PK_PHONGBAN PRIMARY KEY (MaPhong),
    CONSTRAINT U_PHONGBAN UNIQUE (TenPhong)
    CONSTRAINT PK_PHONGBAN FOREIGN KEY (TruongPhong)
    REFERENCES NHANVIEN (MaNV)
)
```

### **Xóa bảng**

DROP TABLE <tên bảng> xóa bảng

Ví dụ DROP TABLE PHONGBAN

### **Thay đổi cấu trúc**

Thêm cột

```
ALTER TABLE NHANVIEN ADD ChuyenMon    char(40)
```

Xóa cột

```
ALTER TABLE NHANVIEN DROP ChuyenMon
```

Bổ sung, thay đổi RBTV

```
ALTER TABLE NHANVIEN DROP CONSTRAINT FK_NHANVIEN
```

```
ALTER TABLE NHANVIEN ADD CONSTRAINT FK_NHANVIEN FOREIGN
    KEY (MaNQL) REFERENCES NHANVIEN (MaNV)
```

## 5. Các thao tác cập nhật dữ liệu

### 5.1. Thêm

Có thể thêm một bộ vào bảng bằng cách sử dụng:

```
INSERT INTO <bảng> [<thuộc tính 1>, <thuộc tính 2>, ..., <thuộc tính n>] VALUES  
(<giá trị 1>, <giá trị 2>, ..., <giá trị n>)
```

Chú ý rằng thứ tự giá trị trong VALUES là thứ tự các thuộc tính được chỉ ra trong CREATE TABLE

Ví dụ

```
INSERT INTO PHANCONG VALUES ('NV01', 'DA01', 10 )
```

Có thể thêm nhiều bộ vào bảng bằng cách sử dụng:

```
INSERT INTO <bảng> [<thuộc tính 1>, <thuộc tính 2>, ..., <thuộc tính n>]
```

```
SELECT ... FROM... WHERE...
```

### 5.2. Xóa

Có thể xóa một hay nhiều bộ khỏi bảng bằng cách sử dụng

```
DELETE FROM <bảng>
```

```
[WHERE <điều kiện>]
```

Ví dụ

- Xóa những nhân viên có mức lương dưới 100000

```
DELETE FROM NHANVIEN
```

```
WHERE Luong<100000
```

- Xóa những nhân viên làm việc cho phòng 'Nghiên cứu'

```
DELETE FROM NHANVIEN
```

```
WHERE Phong in (SELECT MaPhong
```

```
FROM PHONGBAN
```

```
WHERE TenPhong = 'Nghiên cứu')
```

Chú ý rằng các bộ trong bảng khác có thể bị xóa do ràng buộc tham chiếu. Để giải quyết vấn đề này có thể không cho xóa hoặc xóa luôn những bộ đang tham chiếu đến.

### 5.3. Cập nhật

Cập nhật các giá trị thuộc tính của một hay nhiều bộ bằng cách sử dụng

UPDATE <bảng>

SET <thuộc tính 1>= <giá trị 1>,

<thuộc tính 2>= <giá trị 2>,

...

<thuộc tính n>= <giá trị n>

[WHERE <điều kiện>]

Các bộ thỏa điều kiện sẽ được cập nhật giá trị cho các thuộc tính. Chú ý rằng các bộ trong bảng khác có thể được cập nhật do ràng buộc tham chiếu. Để giải quyết vấn đề này có thể không cho thay đổi hoặc thay đổi luôn những giá trị tham chiếu đến.

Ví dụ

Tăng thêm 100000 cho các nhân viên phòng ‘Nghiên cứu’

UPDATE NHANVIEN

SET Luong = Luong + 100000

WHERE Phong IN (SELECT MaPhong

FROM PHONGBAN

WHERE TenPhong = ‘Nghiên cứu’)

## 6. Bài tập

Thực hiện các câu truy vấn trong bài tập chương 4.



## Chương 7

# *Phụ thuộc hàm, khóa và ràng buộc toàn vẹn của lược đồ quan hệ*

---

Phụ thuộc hàm (functional dependency) dùng để biểu diễn một cách hình thức các ràng buộc toàn vẹn (RBTV). Phụ thuộc hàm có tầm quan trọng rất lớn trong việc giải quyết các bài toán tìm khóa, phủ tối thiểu và chuẩn hóa cơ sở dữ liệu. Nội dung chương cũng trình bày ràng buộc toàn vẹn (RBTV), các yếu tố liên quan đến ràng buộc toàn vẹn nhằm bảo đảm tính đúng đắn của dữ liệu.

### 1. Phụ thuộc hàm

#### 1.1. Khái niệm phụ thuộc hàm

Xét quan hệ DEAN(MaDA, TenDA, DdiemDA, Phong). Nhận thấy rằng quan hệ DEAN có MaDA là khóa, nghĩa là từ MaDA có thể xác định được tất cả các thông tin về tên đề án, địa điểm thực hiện đề án và phòng ban chủ trì đề án. Có thể phát biểu lại như sau:

- MaDA xác định TenDA hay TenDA phụ thuộc hàm vào MaDA
- MaDA xác định DdiemDA hay DdiemDA phụ thuộc hàm vào MaDA
- MaDA xác định Phong hay Phong phụ thuộc hàm vào MaDA

được ký hiệu:

- $\text{MaDA} \rightarrow \text{TenDA}$
- $\text{MaDA} \rightarrow \text{DdiemDA}$
- $\text{MaDA} \rightarrow \text{Phong}$

Một cách tổng quát, ta có:

#### Định nghĩa phụ thuộc hàm

$Q(A_1, A_2, \dots, A_n)$  là quan hệ;  $Q^+ = \{A_1, A_2, \dots, A_n\}$ ; X, Y là hai tập con của  $Q^+$ ;

$t_1, t_2$  là hai bộ bất kỳ của Q.

Khi đó:  $X \rightarrow Y \Leftrightarrow (t_1.X = t_2.X \Rightarrow t_1.Y = t_2.Y)$

Ta nói  $X$  xác định  $Y$  hay  $Y$  phụ thuộc hàm vào  $X$ .  $X$  được gọi là vế trái phụ thuộc hàm,  $Y$  được gọi là vế phải phụ thuộc hàm.

### **Phụ thuộc hàm hiển nhiên**

$X \rightarrow Y$  gọi là phụ thuộc hàm hiển nhiên nếu  $Y \subseteq X$ .

### **Phụ thuộc hàm nguyên tố**

$X \rightarrow Y$  được gọi là phụ thuộc hàm nguyên tố (hoặc nói cách khác  $Y$  được gọi là phụ thuộc đầy đủ vào  $X$ ) nếu  $\forall X' \subsetneq X$  đều không có phụ thuộc hàm  $X' \rightarrow Y$ .

Như vậy các phụ thuộc hàm  $\text{MaDA} \rightarrow \text{TenDA}$ ,  $\text{MaDA} \rightarrow \text{DdiemDA}$ ,  $\text{MaDA} \rightarrow \text{Phong}$  là phụ thuộc hàm nguyên tố.

Xét quan hệ CTIETHOADON (SoHD, MaHang, SoLuong, DonGia, ThanhTien) và các phụ thuộc hàm như sau:

- $\{\text{SoHD}, \text{MaHang}\} \rightarrow \text{SoLuong}$
- $\{\text{SoHD}, \text{MaHang}\} \rightarrow \text{DonGia}$
- $\{\text{SoHD}, \text{MaHang}\} \rightarrow \text{ThanhTien}$

Nhận thấy rằng SoLuong phụ thuộc đầy đủ vào  $\{\text{SoHD}, \text{MaHang}\}$  nhưng DonGia chỉ phụ thuộc vào MaHang (là một thuộc tính khóa) chứ không phụ thuộc đầy đủ vào khóa  $\{\text{SoHD}, \text{MaHang}\}$ . Như vậy, trên một lược đồ quan hệ có thể tồn tại nhiều phụ thuộc hàm.

Tập các phụ thuộc hàm được ký hiệu  $F$

### **1.2. Hệ luật dẫn Amstrong**

Gọi  $F$  là tập phụ thuộc hàm đối với lược đồ quan hệ  $R$  định nghĩa trên tập thuộc tính  $U$  và  $X \rightarrow Y$  là một phụ thuộc hàm;  $X, Y \subseteq U$ . Khi đó ta nói rằng  $X \rightarrow Y$  được suy diễn logic từ  $F$  nếu  $R$  thỏa các phụ thuộc hàm của  $F$  thì cũng thỏa  $X \rightarrow Y$  và ký hiệu là  $F \models X \rightarrow Y$

### **Bao đóng của tập phụ thuộc hàm**

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

Trong trường hợp  $F = F^+$  thì  $F$  là họ đầy đủ của các phụ thuộc hàm.

## Hệ luật dẫn Amstrong

Amstrong đã đưa ra hệ luật tiên đề còn được gọi là hệ luật dẫn Amstrong sau:

Với  $X, Y, Z, W \subseteq U$ ,  $U$  là tập thuộc tính. Phụ thuộc hàm có các tính chất sau:

(1) Tính phản xạ

Nếu  $Y \subseteq X$  thì  $X \rightarrow Y$

(2) Tính tăng trưởng

Nếu  $X \rightarrow Y$  thì  $XZ \rightarrow YZ$

(3) Tính bắc cầu

Nếu  $X \rightarrow Y$  và  $Y \rightarrow Z$  thì  $X \rightarrow Z$

**Từ hệ tiên đề Amstrong suy ra một số tính chất sau:**

(1) Tính phân rã sau

Nếu  $X \rightarrow YZ$  và  $X \rightarrow Y$  thì  $X \rightarrow Z$

(2) Tính kết hợp

Nếu  $X \rightarrow Y$  và  $X \rightarrow Z$  thì  $X \rightarrow YZ$

(3) Tính tựa bắc cầu

Nếu  $X \rightarrow Y$  và  $YZ \rightarrow W$  thì  $XZ \rightarrow W$

## Ví dụ

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

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

Chứng tỏ  $AB \rightarrow E$  được suy diễn từ  $F$ .

(1)  $AB \rightarrow C$

(2)  $AB \rightarrow AB$  (tính phản xạ)

(3)  $AB \rightarrow B$  (tính phân rã)

(4)  $B \rightarrow D$

(5)  $AB \rightarrow D$  (tính bắc cầu 3+4)

(6)  $AB \rightarrow CD$  (tính hợp 1+5)

(7)  $CD \rightarrow E$

(8)  $AB \rightarrow E$  (tính bắc cầu 6+7)

### 1.3. Thuật toán tìm bao đóng của tập thuộc tính

#### Bao đóng của tập thuộc tính

Bao đóng của tập thuộc tính  $X$  đối với tập phụ thuộc hàm  $F$ , ký hiệu là  $X_F^+$  là tập tất cả các thuộc tính  $A$  có thể suy dẫn từ  $X$  nhờ tập bao đóng của các phụ thuộc hàm  $F^+$

$$X_F^+ = \{A \in Q^+ \mid X \rightarrow A \in F^+\}$$

#### Thuật toán tìm bao đóng của tập thuộc tính

Input:  $(Q, F), X \subseteq Q^+$

Output:  $X_F^+$

Bước 1: Tính dãy  $X^{(0)}, X^{(1)}, \dots, X^{(i)}$ :

- $X^{(0)} = X$
- $X^{(i+1)} = X^{(i)} \cup Z, \exists (Y \rightarrow Z) \in F (Y \subseteq X^{(i)})$ , loại  $(Y \rightarrow Z)$  ra khỏi  $F$
- Dừng khi  $X^{(i+1)} = X^{(i)}$  hoặc khi  $X^{(i)} = Q^+$

Bước 2: Kết luận  $X_F^+ = X^{(i)}$ :

#### Ví dụ 1

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

$F = \{ \begin{array}{l} f1: B \rightarrow A, \\ f2: DA \rightarrow CE, \\ f3: D \rightarrow H, \\ f4: GH \rightarrow C, \\ f5: AC \rightarrow D \end{array} \}$

Tìm  $AC_F^+$

Bước 1:  $X_0 = AC$

Bước 2: Từ f1 đến f4 không thỏa, f5 thỏa nên  $X_1 = AC \cup D = ACD$

Lặp lại bước 2:

f1 không thỏa, f2 thỏa nên  $X_2 = ACD \cup CE = ACDE$

f3 thỏa nên  $X_3 = ACDE \cup H = ACDEH$

f4 không thỏa, f5 đã thỏa

Lặp lại bước 2: f2, f3 và f5 đã thỏa, f1 và f4 không thỏa. Nên  $X_4 = X_3 = ACDEH$

Vậy  $AC_F^+ = ACDEH$

#### 1.4. Bài toán thành viên

Cho tập thuộc tính Q, tập phụ thuộc hàm F trên Q và một phụ thuộc hàm  $X \rightarrow Y$  trên Q.

Câu hỏi đặt ra rằng  $X \rightarrow Y \in F^+$  hay không?

Giải quyết:  $X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X^+$

#### Ví dụ

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

$F = \{ \begin{array}{l} f1: B \rightarrow A, \\ f2: DA \rightarrow CE, \\ f3: D \rightarrow H, \\ f4: GH \rightarrow C, \\ f5: AC \rightarrow D \end{array} \}$

Cho biết  $AC \rightarrow E$  có thuộc  $F^+$  không?

Ta có  $AC_F^+ = ACDEH$

Vì  $E \in AC_F^+$  nên  $AC \rightarrow E \in F^+$

#### 1.5. Phủ tối thiểu của một tập phụ thuộc hàm

##### 1.5.1. Hai tập phụ thuộc hàm tương đương

Hai tập phụ thuộc hàm F và G tương đương nếu  $F^+ = G^+$ . Ký hiệu  $F \equiv G$

F được gọi là phủ G nếu  $F^+ \supseteq G^+$

### Thuật toán xác định F và G có tương đương không

Bước 1:

Với mỗi phụ thuộc hàm  $X \rightarrow Y \in F$

Kiểm tra  $X \rightarrow Y$  có là thành viên của G không.

Bước 2:

Với mỗi phụ thuộc hàm  $X \rightarrow Y \in G$

Kiểm tra  $X \rightarrow Y$  có là thành viên của F không.

Bước 3:

Nếu cả bước 1 và 2 đều đúng thì kết luận  $F \equiv G$

### Ví dụ

Cho lược đồ quan hệ R(A, B, C, D, E) và tập phụ thuộc hàm

$F = \{ A \rightarrow BC, A \rightarrow D, CD \rightarrow E \}$

$G = \{ A \rightarrow BCE, A \rightarrow ABD, CD \rightarrow E \}$

Cho biết F và G có tương đương không?

Bước 1:

$$A_G^+ = ABCDE \Rightarrow (A \rightarrow BC \in G^+ \wedge A \rightarrow D \in G^+)$$

Hơn nữa ta có  $(CD \rightarrow E \in F \wedge CD \rightarrow E \in G)$

Vậy mọi phụ thuộc hàm trong F đều là thành viên của G

Bước 2:

$$A_F^+ = ABCDE \Rightarrow (A \rightarrow BCE \in F^+ \wedge A \rightarrow ABD \in F^+)$$

Hơn nữa ta có  $(CD \rightarrow E \in G \wedge CD \rightarrow E \in F)$

Vậy mọi phụ thuộc hàm trong G đều là thành viên của F

Bước 3: Kết luận  $F \equiv G$

### 1.5.2. Phủ tối thiểu của tập phụ thuộc hàm

#### Phụ thuộc hàm có thuộc tính về trái dư thừa

Cho  $F$  là tập các phụ thuộc hàm trên lược đồ quan hệ  $Q$ . Khi đó  $Z \rightarrow Y \in F$  là phụ thuộc hàm có thuộc tính về trái dư thừa nếu tồn tại  $A \in Z$  mà  $F = F - (Z \rightarrow Y) \cup ((Z - A) \rightarrow Y)$

Ngược lại  $Z \rightarrow Y$  là phụ thuộc hàm có thuộc tính về trái không dư thừa hay  $Y$  phụ thuộc đầy đủ vào  $Z$ .  $Z \rightarrow Y$  còn được gọi là phụ thuộc hàm đầy đủ.

Ví dụ

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

$$F = \{ A \rightarrow B, BC \rightarrow D, C \rightarrow D \}$$

Khi đó  $A \rightarrow B, C \rightarrow D$  là những phụ thuộc hàm đầy đủ.  $BC \rightarrow D$  là phụ thuộc hàm có thuộc tính về trái  $B$  dư thừa.

Vấn đề là tìm các phụ thuộc hàm đầy đủ tương ứng bằng cách loại bỏ các thuộc tính dư thừa.

#### Thuật toán tìm các phụ thuộc hàm đầy đủ tương ứng

Thực hiện với mỗi phụ thuộc hàm  $X \rightarrow Y \in F$

Bước 1:  $X = A_1, A_2, \dots, A_n$  ( $n \geq 2$ , với  $n=1$  thì  $X \rightarrow Y$  là đầy đủ)

Đặt  $Z = X$

Bước 2: Với mỗi  $A_i$ , thực hiện

- $Tam = Z \setminus A_i$
- Nếu  $Tam \rightarrow Y \in F^+$  thì  $Z = Tam$

#### Tập phụ thuộc hàm có về phải một thuộc tính

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

Ví dụ:

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

$$F = \{ A \rightarrow BC, A \rightarrow D, CD \rightarrow E \}$$

Khi đó  $G = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, CD \rightarrow E \}$  là tập phụ thuộc hàm có vẻ phải một thuộc tính và  $F \equiv G$

### Tập phụ thuộc hàm không dư thừa

$F$  là tập phụ thuộc hàm không dư thừa

nếu không tồn tại  $F' \subset F$  sao cho  $F' \equiv F$ .

Ngược lại  $F$  được gọi là tập phụ thuộc hàm dư thừa.

### Thuật toán loại những phụ thuộc hàm dư thừa

Với mỗi phụ thuộc hàm  $X \rightarrow Y \in F$

nếu  $X \rightarrow Y$  là thành viên của  $F - \{X \rightarrow Y\}$  thì loại  $X \rightarrow Y$  khỏi  $F$ .

#### 1.5.3. Phủ tối thiểu của tập phụ thuộc hàm

$F$  được gọi là phủ tối thiểu của tập phụ thuộc hàm (hay tập phụ thuộc hàm tối thiểu) nếu thỏa:

- (i)  $F$  là tập phụ thuộc hàm có thuộc tính về trái không dư thừa
- (ii)  $F$  là tập phụ thuộc hàm có vẻ phải một thuộc tính
- (iii)  $F$  là tập phụ thuộc hàm không dư thừa

### Thuật toán tìm phủ tối thiểu của tập phụ thuộc hàm

Bước 1: Loại các thuộc tính có vẻ trái dư thừa của mọi phụ thuộc hàm

Bước 2: Phân rã các phụ thuộc hàm có vẻ phải nhiều thuộc tính thành các phụ thuộc hàm có vẻ phải một thuộc tính

Bước 3: Loại các phụ thuộc hàm dư thừa khỏi  $F$

### Ví dụ

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

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

Tìm phủ tối thiểu của  $F$ .

Bước 1:



- Với  $DA \rightarrow CE$ :
  - Giả sử D thừa thì  $A \rightarrow CE \in F^+$ : ta có  $A_F^+ = A$ ,  $CE \notin A_F^+$  nên D không thừa
  - Giả sử A thừa thì  $B \rightarrow CE \in F^+$ : ta có  $D_F^+ = DH$ ,  $CE \notin D_F^+$  nên A không thừa
- Với  $GH \rightarrow C$ :
  - Giả sử G thừa thì  $H \rightarrow C \in F^+$ : ta có  $H_F^+ = H$ ,  $C \notin H_F^+$  nên G không thừa
  - Giả sử H thừa thì  $G \rightarrow C \in F^+$ : ta có  $G_F^+ = G$ ,  $C \notin G_F^+$  nên H không thừa
- Với  $AC \rightarrow D$ :
  - Giả sử A thừa thì  $C \rightarrow D \in F^+$ : ta có  $C_F^+ = C$ ,  $D \notin C_F^+$  nên A không thừa
  - Giả sử C thừa thì  $A \rightarrow D \in F^+$ : ta có  $A_F^+ = A$ ,  $D \notin A_F^+$  nên C không thừa

Vậy mọi phụ thuộc hàm đều là đầy đủ.

Bước 2: Phân rã về phải ta có  $F = \{ B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$

Bước 3: Loại các phụ thuộc hàm dư thừa:

- Với  $B \rightarrow A$ : ta có  $B_{F \setminus \{B \rightarrow A\}}^+ = B$ ,  $A \notin B_{F \setminus \{B \rightarrow A\}}^+$  nên  $B \rightarrow A$  là không thừa
- Với  $DA \rightarrow C$ : ta có  $DA_{F \setminus \{DA \rightarrow C\}}^+ = DAEH$ ,  $C \notin DA_{F \setminus \{DA \rightarrow C\}}^+$  nên  $DA \rightarrow C$  là không thừa
- Với  $DA \rightarrow E$ : ta có  $DA_{F \setminus \{DA \rightarrow E\}}^+ = DACH$ ,  $E \notin DA_{F \setminus \{DA \rightarrow E\}}^+$  nên  $DA \rightarrow E$  là không thừa
- Với  $D \rightarrow H$ : ta có  $D_{F \setminus \{D \rightarrow H\}}^+ = D$ ,  $H \notin D_{F \setminus \{D \rightarrow H\}}^+$  nên  $D \rightarrow H$  là không thừa
- Với  $GH \rightarrow C$ : ta có  $GH_{F \setminus \{GH \rightarrow C\}}^+ = GH$ ,  $C \notin GH_{F \setminus \{GH \rightarrow C\}}^+$  nên  $GH \rightarrow C$  là không thừa
- Với  $AC \rightarrow D$ : ta có  $AC_{F \setminus \{AC \rightarrow D\}}^+ = AC$ ,  $D \notin AC_{F \setminus \{AC \rightarrow D\}}^+$  nên  $AC \rightarrow D$  là không thừa

Vậy phủ tối thiểu của F:  $PTT(F) = \{ B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$

## 2. Khóa

### 2.1. Định nghĩa

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ ,  $Q^+$  là tập thuộc tính của quan hệ Q, F là tập phụ thuộc hàm trên Q, K là tập con của  $Q^+$ . Khi đó K gọi là một khóa của Q nếu:

$$(i) \quad K_F^+ = Q^+$$

$$(ii) \quad \text{Không tồn tại } K' \subset K \text{ sao cho } K'^+_F = Q^+$$

Thuộc tính A được gọi là thuộc tính khóa nếu  $A \in K$ , trong đó K là khóa của Q. Ngược lại thuộc tính A được gọi là thuộc tính không khóa.

$K'$  được gọi là siêu khóa nếu  $K \subseteq K'$ .

## 2.2. Thuật toán tìm khóa

### 2.2.1. Thuật toán tìm một khóa

Sử dụng đồ thị có hướng để tìm khóa như sau:

Bước 1:

- Mỗi nút của đồ thị là tên một thuộc tính của lược đồ quan hệ R
- Cung nối hai thuộc tính A và B thể hiện phụ thuộc hàm  $A \rightarrow B$
- Thuộc tính chỉ có các mũi tên đi ra (nghĩa là chỉ nằm trong vế trái của phụ thuộc hàm) được gọi là nút gốc
- Thuộc tính chỉ có các mũi tên đi tới (nghĩa là chỉ nằm trong vế phải của phụ thuộc hàm) được gọi là nút lá

Bước 2:

- Xuất phát từ tập các nút gốc (X), dựa trên tập các phụ thuộc hàm F, tìm bao đóng  $X_F^+$ .
- Nếu  $X_F^+ = Q^+$  thì X là khóa, ngược lại bổ sung một thuộc tính không thuộc nút lá vào X rồi thực hiện tìm bao đóng của X. Dừng khi tìm được một khóa của R.

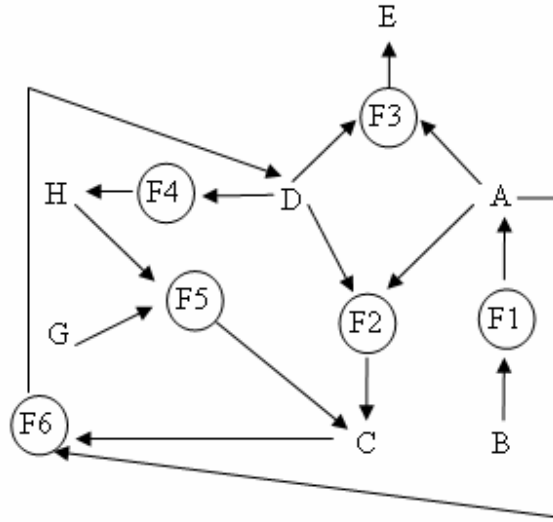
Ví dụ

Cho lược đồ quan hệ R(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 một khóa của R.

Phân rã vế phải ta có  $F = \{ B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$



Nhận thấy từ đồ thị trên, nút B và G là nút gốc. Khóa của R phải chứa thuộc tính B hoặc G, trong ví dụ này chọn B.

$B_F^+ = BA$ , Vì  $B_F^+ \neq Q^+$  nên B không là khóa. Nhận thấy D là thuộc tính ở vế trái của ba phụ thuộc hàm trong F nên bổ sung thuộc tính D vào để xét khóa.

$BD_F^+ = BDACEH$ , vì  $BD_F^+ \neq Q^+$  nên BD không là khóa. Bổ sung thuộc tính G.

$BDG_F^+ = BDGACEH$ , vì  $BDG_F^+ = Q^+$  nên BDG là khóa.

### 2.2.2. Thuật toán tìm tất cả các khóa

Tập thuộc tính nguồn, ký hiệu là N, là tập chứa những thuộc tính chỉ xuất hiện ở vế trái, không nằm bên vế trái và vế phải của mọi phụ thuộc hàm

Tập thuộc tính trung gian, ký hiệu là TG, là tập chứa những thuộc tính vừa xuất hiện ở vế trái, vừa xuất hiện ở vế phải trong các phụ thuộc hàm

#### Thuật toán

Bước 1: Tính tập nguồn N. Nếu  $N_F^+ = Q^+$  thì chỉ có 1 khoá là N, ngược lại qua bước 2

Bước 2: Tính tập trung gian TG. Tính tập tất cả các tập con  $X_i$  của tập TG.

Bước 3: Tìm tập S chứa mọi siêu khóa  $S_i$ :

Với mỗi  $X_i$ , nếu  $(N \cup X_i)_F^+ = Q^+$  thì  $S_i = (N \cup X_i)$

Bước 4: Loại các siêu khóa không tối thiểu:

$\forall S_i, S_j \in S$ , nếu  $S_i \subset S_j$  thì  $S = S - S_j$

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

$$F = \{ AB \rightarrow C, C \rightarrow A \}$$

Tìm mọi khóa của  $R$ .

Bước 1:  $N = \{B\}$ ,  $N_F^+ = B \neq Q^+$

Bước 2:  $TG = \{AC\}$ , tập các tập con trung gian là  $CTG = \{\emptyset, A, C, AC\}$

Bước 3:

N	$X_i$	$N \cup X_i$	$(N \cup X_i)^+$	$(N \cup X_i)^+ = Q^+ ?$
B	$\emptyset$	B	B	Sai
B	A	BA	BAC	Đúng
B	C	BC	BCA	Đúng
B	AC	BAC	BAC	Đúng

Như vậy tập siêu khoá  $S = \{BA, BC, BAC\}$

Bước 4: Loại các siêu khóa không tối thiểu:

Nhận thấy rằng  $BA \subset BAC$  nên loại siêu khóa BAC. Tập các khóa còn lại là  $S = \{BA, BC\}$

### Cải tiến thuật toán

Nhận thấy rằng ở bước 3, khi xét tập một con  $X_i$ , nếu  $(N \cup X_i)^+ = Q^+$ , khi đó  $N \cup X_i$  là một khóa. Trong trường hợp này nếu xét tiếp các tập  $X_j : X_i \subset X_j$ , thì hiển nhiên  $(N \cup X_j)^+ = Q^+$ , và cuối cùng siêu khoá  $N \cup X_j$  cũng sẽ bị loại ở bước 4.

Hơn nữa, bước tính tập con đầu tiên là rỗng cũng không cần tính.

Do đó, khi tính bao đóng của tập thuộc tính, không tính  $X_i = \emptyset$  và nếu  $(N \cup X_i)^+ = Q^+$ , thực hiện loại bỏ các tính toán cho các trường hợp  $X_j : X_i \subset X_j$ . Khi đó kết thúc thuật toán tại bước 3, không cần thực hiện bước 4 vì tất cả các khóa nhận được ở bước 3 đều là tối thiểu.

## 3. Ràng buộc toàn vẹn

### 3.1. Ràng buộc toàn vẹn – các yếu tố của ràng buộc toàn vẹn

Trong CSDL luôn tồn tại nhiều mối liên hệ giữa các thuộc tính, các bộ, các bảng với nhau. Các mối liên hệ này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan đều phải thỏa mãn tại mọi thời điểm. Những điều kiện bất biến đó được gọi là ràng buộc toàn vẹn.

Ràng buộc toàn vẹn (Integrity constraint) viết tắt tiếng việt là RBTV, là một điều kiện được định nghĩa liên quan đến một hay nhiều quan hệ khác nhau. Các mối liên hệ ràng buộc là những điều kiện bất biến mà mọi thể hiện của quan hệ đều phải thỏa mãn trong mọi thời điểm.

Trong thực tế, RBTV là các quy tắc quản lý trong thế giới thực.

Mục đích của RBTV là bảo đảm tính nhất quán của dữ liệu, bảo đảm rằng dữ liệu luôn biểu diễn đúng ngữ nghĩa trong thực tế.

Ví dụ: với giao dịch chuyển tiền, thực hiện qua các bước sau:

- Tại tài khoản của người gửi: trừ tiền
- Tại tài khoản của người nhận: thêm tiền
- Nếu cả hai việc trên đều thành công thì hoàn tất giao dịch, ngược lại quay lui giao dịch.

Qua ví dụ trên, giao dịch thực hiện trừ tiền và tăng tiền trong tài khoản của người gửi và người nhận, nếu không thì không thực hiện gì cả. Việc thực hiện giao dịch cần phải bảo đảm tính đúng của dữ liệu.

Mỗi một RBTV có các yếu tố sau:

### **Bối cảnh**

Bối cảnh là một hay nhiều quan hệ cần phải sử dụng để kiểm tra RBTV. Hay nói cách khác bối cảnh của RBTV là những quan hệ có khả năng bị vi phạm RBTV khi thực hiện các thao tác cập nhật dữ liệu (thêm, xóa, sửa các bộ)

### **Biểu diễn: điều kiện hay nội dung**

Điều kiện được kiểm tra trên mọi thay đổi của thể hiện của các quan hệ cơ sở. Điều kiện của một RBTV có thể được biểu diễn bằng nhiều cách khác nhau, chẳng hạn như ngôn ngữ tự nhiên, ngôn ngữ hình thức (thuật toán, đại số quan hệ,...).

Khi một RBTV bị vi phạm, có thể xử lý bằng cách thông báo cho NSD biết RBTV đã bị vi phạm như thế nào, hoặc từ chối thực hiện thao tác cập nhật dữ liệu và thông báo cho NSD biết thao tác cập nhật bị từ chối trên các quan hệ nào và tại các bước nào.

## Tầm ảnh hưởng

Trong quá trình phân tích thiết kế CSDL, cần thiết phải lập bảng tầm ảnh hưởng cho RBTV nhằm xác định thời điểm cần kiểm tra RBTV, và khi kiểm tra cần kiểm tra trên quan hệ nào.

Bảng tầm ảnh hưởng của một RBTV có dạng sau:

Tên RBTV	Thêm	Xóa	Sửa
R <sub>1</sub>	+	-	+(A)
R <sub>2</sub>	+	-	+(B)
...	...	...	...
R <sub>n</sub>	-	-	+

Trong đó:

Dấu + thể hiện thao tác có thể gây ra vi phạm RBTV. Trong trường hợp +(A) cho biết thao tác sửa có thể gây vi phạm trên thuộc tính A. Dấu - thể hiện thao tác không vi phạm RBTV.

### 3.2. Các loại ràng buộc toàn vẹn

Trong quá trình phân tích thiết kế CSDL, yêu cầu cần thiết là phải tìm được những RBTV tiềm ẩn trong CSDL. Việc phân loại RBTV cho phép người phân tích tìm kiếm đầy đủ, tránh bỏ sót những RBTV. Các loại RBTV được phân thành hai dạng chính như sau:

- RBTV có bối cảnh là một quan hệ
- RBTV có bối cảnh là nhiều quan hệ

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

RBTV có bối cảnh là một quan hệ được chia thành ba loại: RBTV miền giá trị, RBTV liên bộ và RBTV liên thuộc tính

##### 3.2.1.1. Ràng buộc toàn vẹn miền giá trị

Quy định rõ về miền giá trị của một thuộc tính.

Ví dụ. Thời gian phân công tham gia đề án của một nhân viên không quá 40h/tuần

- Bối cảnh: quan hệ PHANCONG

- Biểu diễn:  $\forall pc \in PHANCONG(pc.ThoiGian \leq 40)$
- Bảng tầm ảnh hưởng:

RB1	Thêm	Xóa	Sửa
PHANCONG	+	-	+(ThoiGian)

Ví dụ. Điểm của môn học phải là thang điểm 10

- Bối cảnh: quan hệ KETQUA (MaMH, MaLop, MaKH, Diem )
- Biểu diễn:  $\forall kq \in KETQUA(kq.Diem \geq 0 \wedge kq.Diem \leq 10)$
- Bảng tầm ảnh hưởng:

RB2	Thêm	Xóa	Sửa
KETQUA	+	-	+(Diem)

### 3.2.1.2. Ràng buộc toàn vẹn liên thuộc tính

Quy định các ràng buộc giữa các thuộc tính khác nhau trong cùng một quan hệ

Ví dụ. Ngày trả sách phải là bằng hoặc sau ngày mượn sách

- Bối cảnh: quan hệ MUONSACH(MaSach, MaDocGia, NgayMuon, NgayHenTra, NgayThucTra)
- Biểu diễn:  $\forall ms \in MUONSACH(ms.NgayMuon \leq ms.NgayHenTra \wedge ms.NgayMuon \leq ms.NgayThucTra)$
- Bảng tầm ảnh hưởng:

RB3	Thêm	Xóa	Sửa
MUONSACH	+	-	+(NgayMuon, NgayHenTra, NgayThucTra)

### 3.2.1.3. Ràng buộc toàn vẹn liên bộ

Quy định sự tồn tại của một hoặc nhiều bộ phụ thuộc vào sự tồn tại của một hoặc nhiều bộ khác trong cùng quan hệ.

**RBTV khóa chính là RBTV liên bộ**

Ví dụ. Mỗi đề án trong công ty có một mã duy nhất để phân biệt với các đề án khác

- Bối cảnh: quan hệ DEAN
- Điều kiện:  $\forall da1, da2 \in DEAN : da1 \neq da2 \Rightarrow (da1.MaDA \neq da2.MaDA)$
- Bảng tầm ảnh hưởng:

RB4	Thêm	Xóa	Sửa
DEAN	+	-	+(MaDA)

### RBTV về số lượng các bộ trong một quan hệ.

Ví dụ. Mỗi sinh viên trong một học kỳ được đăng ký không quá 8 môn học

- Bối cảnh: quan hệ DANGKY(MaSV, MaMH)
- Biểu diễn:  $\forall dk1 \in DANGKY : count(dk2 \in DANGKY \mid dk2.MaSV = dk1.MaSV) \leq 8$ , trong đó count() là hàm đếm số bộ của một quan hệ thỏa điều kiện trong ngoặc ().
- Bảng tầm ảnh hưởng:

RB4	Thêm	Xóa	Sửa
DANGKY	+	-	+(MaSV)

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

RBTV có bối cảnh là nhiều quan hệ được chia thành năm loại:

- RBTV tham chiếu
- RBTV liên bộ - liên quan hệ
- RBTV liên thuộc tính - liên quan hệ
- RBTV do thuộc tính tổng hợp
- RBTV do chu trình

#### 3.2.2.1. RBTV tham chiếu

Quy định giá trị xuất hiện của một tập thuộc tính trong một quan hệ phải xuất hiện trong một tập thuộc tính trong một quan hệ khác. RBTV này còn được gọi là RBTV tham chiếu, RBTV phụ thuộc tồn tại hay RBTV khóa ngoại.



### Ví dụ RBTV trên 2 quan hệ

Một nhân viên phải thuộc về một phòng trong công ty, nghĩa là trong quan hệ NHANVIEN, nếu một mã phòng (Phong) mà nhân viên trực thuộc xuất hiện, thì mã phòng này phải xuất hiện trong quan hệ PHONGBAN, cụ thể là thuộc tính (MaPhong). Như vậy:

- Bối cảnh: NHANVIEN, PHONGBAN
- Biểu diễn:  
$$\forall nv \in NHANVIEN ((nv.Phong = NULL) \vee (\exists pb \in PHONGBAN (nv.Phong = pb.MaPhong)))$$
- Bảng tầm ảnh hưởng:

RB5	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(Phong)
PHONGBAN	-	+	+(MaPhong)

### Ví dụ RBTV trên 1 quan hệ

Người quản lý (MaNQL) của một nhân viên cũng phải là một nhân viên trong công ty

- Bối cảnh: NHANVIEN
- Điều kiện:  
$$\forall nv \in NHANVIEN ((nv.MaNQL = NULL) \vee (\exists nv1 \in NHANVIEN (nv.MaNQL = nv1.MaNV)))$$
- Bảng tầm ảnh hưởng:

RB6	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(MaNV, MaNQL)

### Ảnh hưởng của RBTV đối với các thao tác thêm, xóa, sửa dữ liệu

Giả sử  $r_2$  có một khóa ngoại  $\alpha$  tham chiếu đến K trong  $r_1$ , khi đó:  $\pi_\alpha(r_2) \subseteq \pi_K(r_1)$

#### Thêm

Khi thêm một bộ  $t_2$  vào  $r_2$  thì phải bảo đảm tồn tại  $t_1$  trong  $r_1$  sao cho  $t_1[K] = t_2[\alpha]$

## Xóa

Giả sử xóa  $t_1$  khỏi  $r_1$ . Khi đó cần xử lý các bộ trong  $r_2$  tham chiếu tới  $t_1$ , nghĩa là  $s = \sigma_{\alpha=t_1[K]}(r_2)$ . Nếu  $s \neq \emptyset$  thì

- Không thực hiện hành động xóa dữ liệu, hoặc
- Xóa dây chuyền, nghĩa là xóa tất cả các bộ trong  $s$

## Sửa

Trường hợp cập nhật  $t_2$  trong  $r_2$

- Cập nhật  $t_2$  trong  $r_2$ , sửa khóa ngoại  $\alpha$
- Tương tự như trường hợp thêm dữ liệu
- Kiểm tra  $t_2'[\alpha] \in \pi_K(r_1)$

Trường hợp cập nhật  $t_1$  trong  $r_1$

- Cập nhật  $t_1$  trong  $r_1$
- Tương tự như trường hợp xóa dữ liệu
- Kiểm tra  $\sigma_{\alpha=t_1[K]}(r_2) = \emptyset$

### 3.2.2.2. RBTV liên bộ - liên quan hệ

Quy định về từng nhóm các bộ của nhiều quan hệ bối cảnh khác nhau.

Ví dụ Một hóa đơn bán hàng phải có ít nhất một mặt hàng, nghĩa là một chi tiết hóa đơn bán hàng phải có ít nhất một mặt hàng.

- Bối cảnh: HOADON, CTIETHD
- Biểu diễn:  $\forall hd \in HOADON (\exists cthd \in CTIETHD (hd.MaHD = cthd.MaHD))$
- Bảng tầm ảnh hưởng:

RB7	Thêm	Xóa	Sửa
HOADON	+	-	+(MaHD)
CTIETHD	-	+	+(MaHD)

### 3.2.2.3. RBTV liên thuộc tính - liên quan hệ

Quy định về mối liên hệ giữa các thuộc tính trên nhiều quan hệ bối cảnh khác nhau.

Ví dụ Giả sử cho phép thanh toán tiền nhiều lần và thanh toán sau khi mua hàng, khi đó ngày thanh toán tiền theo một hóa đơn mua hàng phải bằng hoặc sau ngày mua hàng.

- Bối cảnh:  $HOADON(\underline{MaHD}, MaKH, NgayHD, TriGia),$   
 $THANHTOAN(\underline{MaHD}, \underline{NgayTToan}, \underline{LanTToan}, SoTienTToan)$

- Biểu diễn:

$$\forall hd \in HOADON (\forall tt \in THANHTOAN (hd.MaHD = tt.MaHD \Rightarrow hd.NgayHD \leq tt.NgayTToan))$$

- Bảng tầm ảnh hưởng:

RB8	Thêm	Xóa	Sửa
HOADON	+	-	+(MaHD, NgayHD)
THANHTOAN	+	-	+(MaHD, NgayTToan)

### 3.2.2.4. RBTV do thuộc tính tổng hợp

Quy định về mối liên hệ giữa các thuộc tính do sự có mặt của thuộc tính tính toán.

Ví dụ Điểm trung bình của sinh viên bằng trung bình của các môn mà sinh viên theo học

- Bối cảnh:  $SINHVIEN(\underline{MaSV}, HoSV, TenSV, Khoa, DTB)$   
 $KETQUA(\underline{MaSV}, \underline{MaMon}, Diem)$

- Biểu diễn:

$$\forall sv \in SINHVIEN (\exists kq \in KETQUA (sv.MaSV = kq.MaSV \Rightarrow sv.DTB = AVG(kq.Diem)))$$

- Bảng tầm ảnh hưởng:

RB9	Thêm	Xóa	Sửa
SINHVIEN	+	-	+(MaSV, DTB)
KETQUA	+	+	+(MaSV, Diem)

### 3.2.2.5. RBTV do chu trình

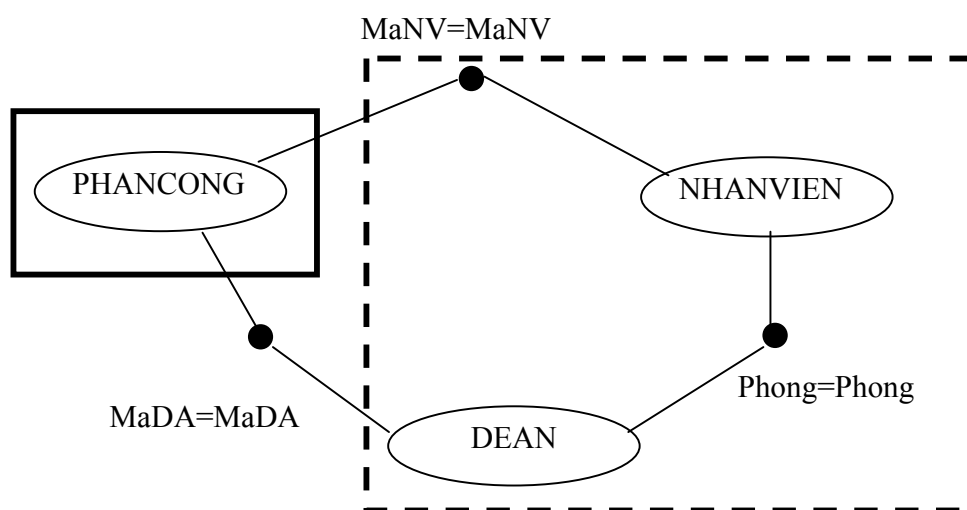
Xảy ra khi có sự hiện diện của chu trình. Để nhận diện chu trình, người ta biểu diễn lược đồ CSDL như sau:

Nút thể hiện lược đồ	
Nút thuộc tính kết	MaNV=MaNV 
Cung nối giữa nút lược đồ và nút thuộc tính kết	

Ví dụ Một nhân viên chỉ được phân công vào các đề án do phòng mình chủ trì

- Bối cảnh: NHANVIEN, DEAN, PHANCONG

Đồ thị thể hiện chu trình như sau:



- Biểu diễn:

$\forall pc \in PHANCONG (\exists nvda \in NV\_DA (nvda.MaNV = pc.MaNV \wedge nvda.MaDA = pc.MaDA))$   
 với:  $NV\_DA \leftarrow NHANVIEN \triangleright \triangleleft_{Phong=MaPhong} DEAN$

- Bảng tầm ảnh hưởng:

RB10	Thêm	Xóa	Sửa
------	------	-----	-----

NHANVIEN	-	+	+(MaNV, Phong)
DEAN	-	+	+(MaDA, Phong)
PHANCONG	+	-	+(MaDA, MaNV)

#### 4. Bài tập

##### Bài tập 1

- Hãy chứng minh 3 tính chất phân rã, kết hợp và tựa bắc cầu.
- Hãy tìm hiểu các tính chất của bao đóng tập thuộc tính, phủ tối thiểu

##### Bài tập 2

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

$$F = \{A \rightarrow C, A \rightarrow EG, B \rightarrow D, G \rightarrow E\}$$

Tìm  $AB_F^+, CGD_F^+, A_F^+$

##### Bài tập 3

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

$$F = \{B \rightarrow C, A \rightarrow EG, B \rightarrow A, G \rightarrow E\}$$

Tìm  $AB_F^+, CGD_F^+, A_F^+$

##### Bài tập 4

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

$$F = \{A \rightarrow C, BC \rightarrow D, D \rightarrow E, E \rightarrow A\}$$

Tìm  $AB_F^+, BD_F^+, D_F^+$

##### Bài tập 5

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

$$F = \{B \rightarrow C, AC \rightarrow D, D \rightarrow G, AG \rightarrow E\}$$

Cho biết  $AC \rightarrow E$  có thuộc  $F^+$  không?

Cho biết  $BD \rightarrow AD$  có thuộc  $F^+$  không?

Với các bài tập 7, 8, 9:

- Tìm một khóa (theo thuật toán tìm một khóa)
- Tìm mọi khóa (theo thuật toán tìm mọi khóa)
- Tìm phủ tối thiểu

#### Bài tập 7

Cho lược đồ quan hệ  $R(A, B, C, D)$  và tập phụ thuộc hàm

$$F = \{AB \rightarrow CD, B \rightarrow C, C \rightarrow D\}$$

#### Bài tập 8

Cho lược đồ quan hệ  $R \{ABCDEFGHIJKLM\}$  và tập phụ thuộc hàm

$$F = \{A \rightarrow B, C \rightarrow D, E \rightarrow F, G \rightarrow AHK, AH \rightarrow G, GLC \rightarrow M\}$$

#### Bài tập 9

Cho lược đồ quan hệ  $R \{ABCDEFGHKL\}$  và tập phụ thuộc hàm

$$F = \{A \rightarrow B, AC \rightarrow D, F \rightarrow G, FK \rightarrow LEH, E \rightarrow FH\}$$

#### Bài tập 10

Với hai bài toán tình huống là quản lý đề án và quản lý ngân hàng, ngoại trừ các ràng buộc khoá chính và khoá ngoại, hãy tìm tất cả các RBTV theo yêu cầu: bối cảnh, biểu diễn, tầm ảnh hưởng. Với những RBTV tìm được, hãy phân theo từng loại RBTV.

## Chương 8

# Dạng chuẩn và chuẩn hóa cơ sở dữ liệu

---

Chương này giới thiệu các dạng chuẩn, phân rã bảo toàn thông tin, bảo toàn phụ thuộc hàm, qua đó cũng trình bày cách phân rã bảo toàn bảo toàn thông tin và bảo toàn phụ thuộc.

### 1. Dạng chuẩn của lược đồ quan hệ

Để dễ dàng trình bày các dạng chuẩn, cần nắm rõ các khái niệm:

#### Thuộc tính khoá:

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ , thuộc tính  $B$  được gọi là **thuộc tính khoá** nếu  $B$  là một thuộc tính thành phần trong một khoá nào đó của  $Q$ , ngược lại  $B$  được gọi là **thuộc tính không khoá**

Ví dụ:  $R(A, B, C, D)$ ,  $F = \{AB \rightarrow C, B \rightarrow D, BC \rightarrow A\}$

Trong ví dụ trên, lược đồ  $R$  có 2 khoá là  $AB, BC$ . Khi đó  $A, B, C$  là thuộc tính khoá,  $D$  là thuộc tính không khoá.

#### Giá trị nguyên tố

Giá trị nguyên tố là giá trị không phân nhỏ được nữa.

Ví dụ: Giá trị ChiTietMua: “Bánh Orion 1 gói, Kẹo mút 2 cây” không phải là giá trị nguyên tố vì có thể phân thành: tên hàng, số lượng, đơn vị tính.

#### 1.1. Dạng chuẩn 1

Lược đồ  $Q$  ở dạng chuẩn 1 nếu mọi thuộc tính đều mang giá trị nguyên tố.

Trong bài toán xét dạng chuẩn, dạng chuẩn thấp nhất là dạng chuẩn 1.

Ví dụ. Cho lược đồ  $HOADON(\underline{MaHD}, MaKH, NgayHD, CtietMua, SoTien)$ , và có thể hiện như sau

MaHD	MaKH	NgàyHD	CtietMua			SoTien
			Tên hàng	Số lượng	ĐVT	
HD01	KH01	15-10-05	Bánh Orion	1	Gói	25.000
			Kẹo mút	2	Cây	2.000
HD02	KH01	18-10-05	Gạo	2	Kg	30.000
HD03	KH02	24-10-05	Đường	1	Kg	15.000
			Bánh AFC	2	Gói	24.000

Nhận thấy rằng CTiếtMua không mang giá trị nguyên tố, do đó HOADON không đạt dạng chuẩn 1.

## 1.2. Dạng chuẩn 2

### 1.2.1. Định nghĩa

Lược đồ Q ở dạng chuẩn 2 nếu thỏa:

- (1) Q đạt dạng chuẩn 1
- (2) Mọi thuộc tính không khóa của Q đều phụ thuộc đầy đủ vào khóa

### 1.2.2. Kiểm tra dạng chuẩn 2

Để kiểm tra dạng chuẩn 2 thực hiện:

Bước 1: Tìm mọi khóa của Q

Bước 2: Với mỗi khóa K, tìm bao đóng của tập tất cả các tập con thực sự  $S_i$  của K

Bước 3: Nếu tồn tại bao đóng  $S_i^+$  chứa thuộc tính không khóa thì Q không đạt dạng chuẩn 2, ngược lại Q đạt dạng chuẩn 2.

Ví dụ 1. Cho Q1 (A, B, C, D),  $F=\{A \rightarrow B, B \rightarrow DC\}$

Lược đồ chỉ có một khóa là A, nên mọi thuộc tính đều phụ thuộc đầy đủ vào khóa. Do vậy Q1 đạt dạng chuẩn 2.

Ví dụ 2. Cho Q2 (A, B, C, D),  $F=\{AB \rightarrow D, C \rightarrow D\}$

Lược đồ có khóa là ABC, ngoài ra còn có  $C \subset ABC$  mà  $C \rightarrow D$ , trong đó D là thuộc tính không khóa (nghĩa là thuộc tính D không phụ thuộc đầy đủ vào khóa). Do vậy Q2 không đạt dạng chuẩn 2.



Ví dụ 3:

Xem ví dụ đơn giản bằng CSDL gồm 2 quan hệ MONHOC, SINHVIEN như sau:

MONHOC (MaMH, TenMH, STC, Loai)

*Tân từ:* Mỗi môn học có mã môn học (MaMH) duy nhất để phân biệt với các môn học khác, có tên môn học (TenMH), số tín chỉ (STC), và loại bắt buộc hay tự chọn (Loai)

MaMH	TenMH	STC	Loai
CT101	Nhập môn tin học	4	BB
CT102	TH kỹ năng máy tính	5	BB
TN311	Xác suất thống kê	3	BB
CT103	Thiết kế Web	4	TC
CT110	Nguyên lý lập trình 2	5	BB

SINHVIEN (MSSV, MaMH, TenSV, DiaChi, Diem)

*Tân từ:* Mỗi sinh viên (MSSV) khi thi một môn học (MaMH) được ghi nhận lại điểm (Diem), ngoài ra còn có thông tin liên quan đến sinh viên như họ tên (TenSV), địa chỉ (DiaChi)

MSSV	MaMH	TenSV	DiaChi	Diem
0310677	CT101	Nguyễn Thị Hoa	11 Nguyễn Công Trứ, Đà Lạt	6
0310678	CT101	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	4
0310679	CT101	Lê Thanh Sơn	2 Nhà Chung, Đà Lạt	7
0310677	CT102	Nguyễn Thị Hoa	11 Nguyễn Công Trứ, Đà Lạt	8
0310678	CT102	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	8
0310678	TN311	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	3
0310679	TN311	Lê Thanh Sơn	2 Nhà Chung, Đà Lạt	6
0310677	CT103	Nguyễn Thị Hoa	11 Nguyễn Công Trứ, Đà Lạt	9
0310678	CT103	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	7

0310000	CT110	Nguyễn Ngọc	1 Lê Hồng Phong, Đà Lạt	9
---------	-------	-------------	-------------------------	---

Ở quan hệ SINHVIEN có thể nhận thấy các phụ thuộc:

MSSV, MaMH → Diem

MSSV → TenSV, DiaChi

Vậy quan hệ SINHVIEN không đạt dạng chuẩn 2 vì thuộc tính TenSV, DiaChi là thuộc tính không khoá chỉ phụ thuộc vào MSSV: như vậy không phụ thuộc đầy đủ vào khoá.

Trong quá trình cập nhật và lưu trữ dữ liệu xuất hiện những vấn đề sau:

- Ở quan hệ SINHVIEN, việc lưu trữ thông tin một sinh viên bị lặp lại (tên, địa chỉ)
- Quá trình cập nhật:
  - Sửa đổi: Khi cần sửa đổi địa chỉ của một sinh viên (ví dụ như Nguyễn Thị Hoa) cần phải sửa đổi 3 lần vì trùng lặp thông tin, hơn nữa, khi sửa đổi thông tin của về một sinh viên lại không liên quan đến thông tin về thi cử.
  - Thêm: Nếu chèn thêm một bộ vào quan hệ SINHVIEN mà sinh viên chưa thi môn nào thì không được vì khoá MSSV, MaMH là không đầy đủ. Vấn đề này chỉ được khắc phục khi loại bỏ những thông tin về kết quả thi cử ra khỏi quan hệ.
  - Xóa: Giả sử rằng cần xóa bỏ môn CT110 mà danh sách sinh viên vẫn giữ nguyên. Khi đó xóa bộ {'CT110', 'Nguyên lý lập trình 2', 5, 'BB'} trong quan hệ MONHOC và xóa bộ {'0310000', 'CT110', 'Nguyễn Ngọc', '1 Lê Hồng Phong, Đà Lạt', 9} trong quan hệ SINHVIEN. Khi đó thông tin về sinh viên sẽ bị mất.

Để khắc phục những bất lợi trên, quan hệ SINHVIEN có thể tách thành 2 quan hệ SINHVIEN (MSSV, TenSV, DiaChi) và KETQUATHI (MSSV, MaMH, Diem). Như vậy, 3 quan hệ trên đều đã ở dạng chuẩn thứ 2.

### 1.3. Dạng chuẩn 3

#### 1.3.1. Định nghĩa

##### Định nghĩa 1

Với một lược đồ  $Q$ ;  $X, Y$  là hai tập con của  $Q^+$ ;  $A$  là một thuộc tính. Khi đó  $A$  được gọi là phụ thuộc bắc cầu vào  $X$  nếu thỏa:

$$(1) X \rightarrow Y, Y \rightarrow A$$

$$(2) Y \not\rightarrow X$$

$$(3) A \notin XY$$

Lược đồ  $Q$  ở dạng chuẩn 3 nếu mọi thuộc tính không khóa đều không phụ thuộc bắc cầu vào khóa.

Việc kiểm tra dạng chuẩn 3 theo định nghĩa trên sẽ khó cài đặt. Ta có thể thực hiện kiểm tra dạng chuẩn 3 theo định nghĩa sau:

##### Định nghĩa 2

Lược đồ  $Q$  ở dạng chuẩn 3 nếu mọi phụ thuộc hàm  $X \rightarrow A \in F^+$ , với  $A \notin X$  đều có:

$$(1) X \text{ là siêu khóa, hoặc}$$

$$(2) A \text{ là thuộc tính khóa}$$

#### 1.3.2. Kiểm tra dạng chuẩn 3

Từ định nghĩa 2, để kiểm tra dạng chuẩn 3 thực hiện các bước sau:

Bước 1: Tìm mọi khóa của  $Q$

Bước 2: Phân rã vế phải của mọi phụ thuộc hàm trong  $F$  để tập  $F$  trở thành tập phụ thuộc hàm có vế phải một thuộc tính

Bước 3: Nếu mọi phụ thuộc hàm  $X \rightarrow A \in F$ , mà  $A \notin X$  đều thỏa

$$(1) X \text{ là siêu khóa (vế trái chứa một khóa), hoặc}$$

$$(2) A \text{ là thuộc tính khóa (vế phải là tập con của khóa)}$$

thì  $Q$  đạt dạng chuẩn 3, ngược lại  $Q$  không đạt dạng chuẩn 3.

**Ví dụ.** Cho  $Q(A, B, C, D), F = \{AB \rightarrow D, C \rightarrow D\}$

Bước 1:  $Q$  có một khóa là  $ABC$

Bước 2: Mọi phụ thuộc hàm trong F đều đã có vế phải một thuộc tính.

Bước 3: Với  $AB \rightarrow D$ , nhận thấy rằng  $D \notin ABC$  có

- Vế trái (AB) không phải là siêu khóa.
- Hơn nữa vế phải (D) không là thuộc tính khóa

Vậy Q không đạt dạng chuẩn 3.

## 1.4. Dạng chuẩn BC (Boyce Codd)

### 1.4.1. Định nghĩa

Lược đồ Q ở dạng chuẩn BC nếu mọi phụ thuộc hàm  $X \rightarrow A \in F^+$ , với  $A \notin X$  đều có X là siêu khóa.

### 1.4.2. Kiểm tra dạng chuẩn BC

Từ định nghĩa, để kiểm tra dạng chuẩn BC thực hiện các bước sau:

Bước 1: Tìm mọi khóa của Q

Bước 2: Phân rã vế phải của mọi phụ thuộc hàm trong F để tập F trở thành tập phụ thuộc hàm có vế phải một thuộc tính

Bước 3: Nếu mọi phụ thuộc hàm  $X \rightarrow A \in F$ , mà  $A \notin X$  đều thỏa X là siêu khóa (vế trái chứa một khóa), thì Q đạt dạng chuẩn BC, ngược lại Q không đạt dạng chuẩn BC.

**Ví dụ.** Cho Q (A, B, C, D, E, I),  $F = \{ACD \rightarrow EBI, CE \rightarrow AD\}$

Bước 1: Q có hai khóa là {ACD, CE}

Bước 2: Phân rã vế phải của các phụ thuộc hàm trong F, ta có:

$$F = \{ACD \rightarrow E, ACD \rightarrow B, ACD \rightarrow I, CE \rightarrow A, CE \rightarrow D\}$$

Bước 3: Mọi phụ thuộc hàm trong F đều có vế trái là một siêu khóa

Vậy Q đạt dạng chuẩn BC.

## 1.5. Kiểm tra dạng chuẩn

### Kiểm tra dạng chuẩn của lược đồ quan hệ Q

Bước 1: Tìm mọi khóa của Q

Bước 2: Kiểm tra dạng chuẩn BC, nếu đúng thì Q đạt dạng chuẩn BC,

ngược lại qua bước 3.

Bước 3: Kiểm tra dạng chuẩn 3, nếu đúng thì Q đạt dạng chuẩn 3,

ngược lại qua bước 4.

Bước 4: Kiểm tra dạng chuẩn 2, nếu đúng thì Q đạt dạng chuẩn 2,

ngược lại Q đạt dạng chuẩn 1.

### Kiểm tra dạng chuẩn của lược đồ CSDL

Dạng chuẩn của một lược đồ CSDL là dạng chuẩn thấp nhất trong các dạng chuẩn của các lược đồ quan hệ con.

## 2. Phép phân rã

Mục tiêu của việc thiết kế CSDL quan hệ là tạo ra một tập các lược đồ quan hệ cho phép chúng ta lưu trữ thông tin không có những dư thừa không cần thiết và truy tìm thông tin một cách dễ dàng, chính xác. Việc phân rã một lược đồ thành những lược đồ con đều mong muốn đạt được bảo toàn thông tin và bảo toàn phụ thuộc.

### 2.1. Phân rã bảo toàn thông tin

Cho lược đồ quan hệ Q (TenNCC, DiaChiNCC, SanPham, DonGia)

Phân rã Q thành Q1 và Q2 như sau:

Q1 (TenNCC, SanPham, DonGia)

Q2 (TenNCC, DiaChiNCC)

Khi đó ta có các thể hiện sau:

Q	TenNCC	DiaChiNCC	SanPham	DonGia
	Nguyễn Mai	10 Nguyễn Công Trứ	Bánh xốp	10.000
	Nguyễn Mai	10 Nguyễn Công Trứ	Kẹo mè	20.000
	Nguyễn Mai	20 Nguyễn Văn Trỗi	Kẹo mè	20.000

Q1	TenNCC	SanPham	DonGia
	Nguyễn Mai	Bánh xốp	10.000

Nguyễn Mai	Kẹo mè	20.000
------------	--------	--------

Q2

TenNCC	DiaChiNCC
Nguyễn Mai	10 Nguyễn Công Trứ
Nguyễn Mai	20 Nguyễn Văn Trỗi

$Q1 \triangleright \triangleleft Q2$

TenNCC	DiaChiNCC	SanPham	DonGia
Nguyễn Mai	10 Nguyễn Công Trứ	Bánh xốp	10.000
Nguyễn Mai	10 Nguyễn Công Trứ	Kẹo mè	20.000
Nguyễn Mai	20 Nguyễn Văn Trỗi	Bánh xốp	10.000
Nguyễn Mai	20 Nguyễn Văn Trỗi	Kẹo mè	20.000

Như vậy kết quả thể hiện Q và  $Q1 \triangleright \triangleleft Q2$  là khác nhau, khi đó ta nói phép phân rã gọi là không bảo toàn thông tin (mất mát thông tin).

### Định nghĩa

Q là lược đồ quan hệ, Q1, Q2 là hai lược đồ con có:

$$Q_1^+ \cap Q_2^+ = X$$

$$Q_1^+ \cup Q_2^+ = Q^+$$

Khi đó Q được phân rã thành hai lược đồ con Q1, Q2 là phép phân rã bảo toàn thông tin nếu với r là thể hiện bất kỳ của Q ta có:

$$r = r.Q_1 \triangleright \triangleleft r.Q_2$$

(r là kết quả của phép kết tự nhiên của các hình chiếu của nó trên Q1, Q2)

### 2.2. Phân rã bảo toàn phụ thuộc hàm

Một vấn đề cần quan tâm khi phân rã lược đồ Q thành các lược đồ con Qi với tập các Fi tương ứng được tính từ tập phụ thuộc hàm F. Phép phân rã bảo toàn phụ thuộc (giữ lại

phụ thuộc) nếu với  $r_i$  là thể hiện của  $Q_i$  thỏa điều kiện:  $r_i$  chỉ thỏa những phụ thuộc hàm  $X \rightarrow Y \in F^+$  với  $XY \subseteq Q_i^+$

### Định nghĩa

Gọi  $Q_1, Q_2, \dots, Q_n$  là phân rã của lược đồ quan hệ  $Q$ , tập phụ thuộc hàm  $F$  trên  $Q$ . Hình chiếu của  $F$  trên một tập các thuộc tính  $Q_i^+$  ký hiệu  $\Pi_{Q_i^+}(F)$  là tập các phụ thuộc hàm  $X \rightarrow Y \in F^+$  với  $XY \subseteq Q_i^+$

$$\Pi_{Q_i^+}(F) = F_i^+ = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \text{ và } XY \subseteq Q_i^+\}$$

Khi đó phân rã là bảo toàn tập phụ thuộc hàm  $F$  nếu

$$F \equiv \cup \Pi_{Q_i^+}(F)$$

## 3. Thiết kế CSDL bằng cách phân rã

### 3.1. Phân rã thành dạng chuẩn BC (hoặc dạng chuẩn 3) bảo toàn thông tin

#### 3.1.1. Thuật toán

Bước 1: Tìm tất cả các khóa của  $Q$

Bước 2: Tìm phụ thuộc hàm  $X \rightarrow Y \in F$  có  $X$  không là siêu khóa và  $Y$  không chứa thuộc tính khóa.

- Nếu tìm thấy thì tách  $Q$  thành  $Q_1$  và  $Q_2$  theo cách:
  - $Q_1 = Q[XY]$ ;  $F_1 \equiv \Pi_{Q_1}(F)$  (tìm bao đóng của tất cả các tập con của  $XY$  để tính  $F_1$ ). Tiếp tục phân rã ( $Q_1, F_1$ ).
  - $Q_2 = Q[Q^+ - Y]$ ;  $F_2 \equiv \Pi_{Q_2}(F)$  (tìm bao đóng của tất cả các tập con của  $(Q^+ - Y)$  để tính  $F_2$ ). Tiếp tục phân rã ( $Q_2, F_2$ ).
- Nếu không tìm thấy thì xét dạng chuẩn  $Q_i$ :
  - Nếu mọi phụ thuộc hàm trong  $F_i$  đều có vế trái là siêu khóa thì  $Q_i$  đạt dạng chuẩn BC
  - Nếu có phụ thuộc hàm trong  $F_i$  có vế trái không là siêu khóa và vế phải là thuộc tính khóa thì  $Q_i$  đạt dạng chuẩn 3

#### 3.1.2. Ví dụ

##### Ví dụ 1

Cho  $Q$  (SDIM),  $F = \{SI \rightarrow D, SD \rightarrow M\}$

Bước 1: Q có một khóa là {SI}

Bước 2: Phụ thuộc hàm  $SD \rightarrow M$  có SD không là siêu khóa nên tách:

$$Q1 = (SDM); F1 = \{SD \rightarrow M\}$$

$$Q2 = (SDI); F2 = \{SI \rightarrow D\}$$

Để tìm tập phụ thuộc hàm F1, F2 cần tính bao đóng của mọi tập con.

*Tìm F1:* với  $Q1 = (SDM)$ : bao đóng của mọi tập con

$$S_F^+ = S$$

$$D_F^+ = D$$

$$M_F^+ = M$$

$$SD_F^+ = SDM$$

$$SM_F^+ = SM$$

$$DM_F^+ = DM$$

$$SDM_F^+ = SDM$$

$$F_1 = \Pi_{Q1}(F) = \{SD \rightarrow M, SD \rightarrow SM, SD \rightarrow DM, SD \rightarrow SDM\}$$

$$\Rightarrow F1 = \{SD \rightarrow M\}$$

*Tìm F2:* với  $Q2 = (SDI)$ : bao đóng của mọi tập con

$$S_F^+ = S$$

$$D_F^+ = D$$

$$I_F^+ = I$$

$$SD_F^+ = SDM$$

$$SI_F^+ = SIDM$$

$$DI_F^+ = DI$$

$$SDI_F^+ = SDIM$$



$$F_2 = \Pi_{Q_2}(F) = \{SI \rightarrow D, SI \rightarrow SD, SI \rightarrow DI, SI \rightarrow SDI\}$$

$$\Rightarrow F_2 = \{SI \rightarrow D\}$$

Bước 3: Mọi phụ thuộc hàm trong F1 và F2 đều có vế trái là một siêu khóa nên Q1 và Q2 đạt dạng chuẩn BC.

## Ví dụ 2

Cho Q (ABCDE),  $F = \{BC \rightarrow A, C \rightarrow D, AE \rightarrow B, B \rightarrow D, B \rightarrow E\}$

Bước 1: 2 khóa là {CB, CAE}

Bước 2: Từ  $C \rightarrow D$  tách thành:

- $Q_1(CD)$ ,  $F_1 = \{C \rightarrow D\}$ , Khóa C. Đạt dạng chuẩn BC.
- $Q_2(CABE)$ ,  $F_2 = \{B \rightarrow E, CB \rightarrow A, AE \rightarrow B\}$

Chi tiết tính F1, F2 như sau:

*Tìm F1:* với  $Q_1 = (CD)$

$$C_F^+ = CD$$

$$D_F^+ = D$$

$$CD_F^+ = CD$$

$$\text{Vậy } F_1 = \{C \rightarrow D\}$$

*Tìm F2:* với  $Q_2(CABE)$ :

$$C_F^+ = CD$$

$$A_F^+ = A$$

$$B_F^+ = BDE$$

$$E_F^+ = E$$

$$CA_F^+ = CAD$$

$$CB_F^+ = CBADE$$

$$CE_F^+ = CED$$

$$AB_F^+ = ABDE$$

$$AE_F^+ = AEBD$$

$$BE_F^+ = BED$$

$$CAB_F^+ = CABDE$$

$$CAE_F^+ = CAEBD$$

$$CBE_F^+ = CBEAD$$

$$ABE_F^+ = ABED$$

$$CABE_F^+ = CABED$$

$$F_2 = \{B \rightarrow E, CB \rightarrow AE, AB \rightarrow E, AE \rightarrow B, CAB \rightarrow E, CAE \rightarrow B, CBE \rightarrow A\}$$

$$\text{Vậy } F_2 = \{B \rightarrow E, CB \rightarrow A, AE \rightarrow B\}$$

Xét dạng chuẩn Q2:

Khoá Q2:  $\{CB, CAE\}$ . Vậy Q<sub>2</sub> đã đạt dạng chuẩn 3.

### 3.1.3. Cải tiến thuật toán

Nhận thấy rằng trong tìm phụ thuộc hàm hình chiếu trên Q<sub>i</sub>, xét tập một con X<sub>i</sub>, nếu  $(X_i)_F^+ = Q^+$ , khi đó nếu tiếp tục xét các tập  $X_j : X_i \subset X_j$ , thì hiển nhiên  $(X_j)_F^+ = Q^+$ , và cuối cùng thì phụ thuộc hàm có vế trái X<sub>j</sub> cũng sẽ bị loại để chỉ chọn phụ thuộc hàm có vế trái X<sub>i</sub>.

Do đó, khi tìm phụ thuộc hàm hình chiếu trên Q<sub>i</sub>, xét tập một con X<sub>i</sub>, nếu  $(X_i)_F^+ = Q^+$ , thực hiện loại bỏ các tính toán cho các trường hợp  $X_j : X_i \subset X_j$ .

**Ví dụ:** với tìm F<sub>2</sub> như ví dụ trên:

$$Q_2(CABE), F = \{BC \rightarrow A, C \rightarrow D, AE \rightarrow B, B \rightarrow D, B \rightarrow E\}$$

$$C_F^+ = CD$$

$$A_F^+ = A$$

$$B_F^+ = BDE$$

$$E_F^+ = E$$

$$CA_F^+ = CAD$$

$$CB_F^+ = CBADE, \text{ loại các tập CAB, CBE, CABE}$$

$$CE_F^+ = CED$$

$$AB_F^+ = ABDE$$

$$AE_F^+ = AEBD$$

$$BE_F^+ = BED$$

$$ABE_F^+ = ABED$$

$$F_2 = \{B \rightarrow E, CB \rightarrow AE, AB \rightarrow E, AE \rightarrow B\}$$

$$\text{Vậy } F_2 = \{B \rightarrow E, CB \rightarrow A, AE \rightarrow B\}$$

### 3.2. Phân rã thành dạng chuẩn 3 vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm

Bước 1: Tìm phủ tối thiểu của F.

Bước 2: Loại bỏ tất cả các thuộc tính của Q không liên quan đến một phụ thuộc hàm nào của PTT(F).

Bước 3: Nếu có một phụ thuộc hàm trong PTT(F) liên quan đến mọi thuộc tính của Q thì không thể phân rã. Ngược lại, qua bước 4.

Bước 4: Gom nhóm những phụ thuộc hàm có cùng vế trái. Với mỗi nhóm phụ thuộc hàm có cùng vế trái, tạo thành một lược đồ con.

Bước 5: Kiểm tra các lược đồ con có thỏa dạng chuẩn 3 chưa, nếu chưa thì áp dụng bước 4 để phân rã tiếp.

**Ví dụ.** Cho Q (CTHRSG),  $F = \{C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R\}$

$$PTT(F) = F = \{C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R\}$$

Ta có kết quả phân rã Q1(CT), Q2(HRC), Q3(HTR), Q4(CSG), Q5(HRS)

## 4. Bài tập

Bài tập 7, 8, 9 trong chương 7, với yêu cầu:

- Phân rã thành dạng chuẩn BC hoặc dạng chuẩn 3 bảo toàn thông tin

- Phân rã thành dạng chuẩn 3 bảo toàn thông tin và bảo toàn phụ thuộc hàm

---

## ***Tài Liệu Tham Khảo***

1. Bài giảng Cơ sở dữ liệu, Nguyễn Hữu Tân, Đại Học Đà Lạt, 2004.
2. Bài tập cơ sở dữ liệu, Nguyễn Xuân Huy – Lê Hoài Bắc, NXB thống kê, 2003.
3. Giáo trình Cơ sở dữ liệu, Nguyễn Đăng Ty - Đỗ Phúc, Đại Học Quốc gia Tp. Hồ Chí Minh, 2001.
4. Nhập môn Cơ sở dữ liệu quan hệ, Lê Tiến Vương, NXB Khoa học và Kỹ thuật, 1994.
5. David Maier, *The theory of Relational Databases*, Computer Science Press, Rockville, 1983.