

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC
TS. NGUYỄN HOÀNG SƠN



GIÁO TRÌNH
CƠ SỞ DỮ LIỆU



NHÀ XUẤT BẢN ĐẠI HỌC HUẾ

Biên mục trên xuất bản phẩm của Thư viện Quốc gia Việt Nam

Nguyễn Hoàng Sơn

Giáo trình Cơ sở dữ liệu / Nguyễn Hoàng Sơn. - Huế : Đại học Huế,
2018. - 257tr. : hình vẽ, bảng ; 24cm

Thư mục: tr. 250-251

1. Cơ sở dữ liệu 2. Giáo trình

005.740711 - dc23

DUH0194p-CIP

Mã số sách: GT/45-2018

Nguyen Hoang Son

Department of Mathematics

College of Sciences, Hue University

77 Nguyen Hue Street, Hue City, Vietnam

Email: nhson@hueuni.edu.vn

Last update, January 2020

LỜI NÓI ĐẦU

Cơ sở dữ liệu là lĩnh vực chuyên nghiên cứu về các mô hình dữ liệu, nguyên lý và các phương pháp tổ chức dữ liệu trên các đối tượng mang thông tin. Cơ sở dữ liệu đầu tiên được xây dựng dựa trên các mô hình dữ liệu mạng và mô hình dữ liệu phân cấp vào giữa những năm 1960. Đây được xem như là thế hệ thứ nhất của cơ sở dữ liệu. Sau đó, thế hệ thứ hai tốt hơn của cơ sở dữ liệu ra đời, đó là dựa trên mô hình dữ liệu quan hệ do E. F. Codd đề xuất vào những năm 1970. Hai thế hệ này của cơ sở dữ liệu đã giải quyết hầu hết các mục tiêu đặt ra của cơ sở dữ liệu như tổ chức dữ liệu, truy cập và cập nhật một khối lượng lớn dữ liệu một cách thuận lợi, an toàn và hiệu quả. Đáp ứng được khá nhiều nhu cầu về thu thập và tổ chức dữ liệu quản lý của các xí nghiệp, tổ chức. Tuy vậy, từ những năm 1990 trở lại đây, trong thực tế xuất hiện những đối tượng mới có cấu trúc phức tạp (văn bản, âm thanh, hình ảnh) và động (các chương trình, mô phỏng) mà các thế hệ trước của cơ sở dữ liệu chưa thể đáp ứng được, dẫn đến đòi hỏi một thế hệ khác nữa của cơ sở dữ liệu ra đời. Mô hình dữ liệu mới đáp ứng cho cơ sở dữ liệu thế này là mô hình dữ liệu hướng đối tượng. Lúc này, cơ sở dữ liệu có khả năng hỗ trợ cho các ứng dụng đa phương tiện.

Quá trình hình thành và phát triển của cơ sở dữ liệu được mô tả ngắn gọn như trên cũng đủ cho thấy được tầm quan trọng của cơ sở dữ liệu trong môi trường tính toán hiện đại. Mục đích của giáo trình này nhằm trình bày các khái niệm cơ bản, những tính chất đặc trưng, ứng dụng và cũng như các thuật toán cơ sở quan trọng của lý thuyết cơ sở dữ liệu. Tuy vậy, bên cạnh đó một số nghiên cứu hiện đại, sâu hơn gần đây của lý thuyết cơ sở dữ liệu theo hướng tổ hợp như tập đóng, khóa, phản khóa, chuyển dịch lược đồ quan hệ, họ các tập tối tiểu của thuộc tính, mở rộng phụ thuộc hàm hay tìm các mô tả tương đương của phụ thuộc hàm cũng được giới thiệu. Phần lớn các kết quả theo các cách tiếp cận này được giáo trình phát biểu và chứng minh lại theo hướng

ngắn gọn, súc tích. Nhiều kết quả trong các hướng này được sử dụng làm công cụ cho các hướng nghiên cứu thời sự gần đây như khai phá dữ liệu, trí tuệ nhân tạo, tập thô, tập mờ ... Chính vì vậy, ngoài sinh viên là đối tượng chính cho giáo trình này thì các học viên cao học, nghiên cứu sinh và những ai quan tâm muốn phát triển sâu về lý thuyết cơ sở dữ liệu hoặc xem lý thuyết cơ sở dữ liệu như là một công cụ cơ sở thì có thể đọc thêm về các nội dung này.

Có thể dễ dàng kể ra những nhà toán tin học và khoa học máy tính có nhiều đóng góp quan trọng vào lĩnh vực này trong những năm đầu tiên (1970) như A. V. Aho, W. W. Armstrong, C. Beeri, E. F. Codd, R. Fagin, C. L. Lucchesi, D. Maier, J. D. Ullman ... và những năm 1985 cho đến nay như J. Demetrovics, T. Eiter, Y. Huhtala, Nguyễn Xuân Huy, G. Gottlob, G. O. H. Katona, L. Libkin, H. Mannila, K. J. Räihä, Vũ Đức Thi, Hồ Thuần ... Bản thân nhóm nghiên cứu của tác giả (cùng với Vũ Đức Thi [16]) theo hướng tổ hợp cũng có những đóng góp nhất định trong lĩnh vực này. Tuy nhiên, với khuôn khổ của một giáo trình 3 tín chỉ dành cho sinh viên chuyên ngành toán tin ứng dụng Trường Đại học Khoa học, Đại học Huế, tác giả không thể trình bày sâu hơn nữa các kết quả nghiên cứu của lĩnh vực này theo tiếp cận bằng các công cụ toán tổ hợp như siêu đồ thị, họ trù mật, hệ Sperner, tập thô ... Với những đề cập và phân tích như trên, nội dung của giáo trình được chia làm năm chương. Chương 1 tập trung khái quát về hệ cơ sở dữ liệu, trình bày các khái niệm cơ bản nhất như hệ cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu, mô hình dữ liệu, kiến trúc ba mức trừu tượng của hệ cơ sở dữ liệu, ... Chương 2 giới thiệu hai mô hình dữ liệu cơ bản và quan trọng là mô hình dữ liệu thực thể-mối quan hệ và mô hình dữ liệu quan hệ. Ngôn ngữ SQL được giới thiệu trong Chương 3. Đây là ngôn ngữ văn tin quan hệ, được sử dụng rộng rãi trong các hệ cơ sở dữ liệu thương mại. Chương 4 trình bày quá trình thiết kế một cơ sở dữ liệu quan hệ. Đầu tiên phân tích khi nào một cơ sở dữ liệu là kém, sau đó các bước để có được một cơ sở dữ liệu tốt bằng cách giới thiệu khái niệm phụ thuộc hàm và những vấn đề liên quan đến phụ thuộc hàm như

hệ tiên đề Armstrong, phủ phụ thuộc hàm, khóa và phản khóa, chuẩn hóa lược đồ quan hệ và cuối cùng là phụ thuộc đa trị. Dựa trên kinh nghiệm giảng dạy và nghiên cứu trong nhiều năm về lĩnh vực này nên nội dung các kết quả trong chương này được tác giả sắp xếp, phát biểu và chứng minh lại hơi khác so với các giáo trình cơ sở dữ liệu trước đây. Cấu trúc trình bày như thế này càng rõ hơn ở Chương 5, đó là trình bày những hướng nghiên cứu quan trọng gần đây liên quan đến thiết kế một cơ sở dữ liệu hiệu quả, chặt chẽ về mặt toán học như các mô tả tương đương của phụ thuộc hàm, mở rộng phụ thuộc hàm theo tiếp cận tập thô, phụ thuộc hàm xấp xỉ, họ các tập tối tiểu của thuộc tính và chuyển dịch lược đồ quan hệ.

Trong quá trình biên soạn, giáo trình không thể tránh khỏi những thiếu sót. Vì vậy, chúng tôi rất mong nhận được sự đóng góp ý kiến của bạn đọc để giáo trình được hoàn thiện hơn trong lần tái bản sau.

Nguyễn Hoàng Sơn

MỤC LỤC

Lời nói đầu	1
Chương 1. Khái quát về cơ sở dữ liệu	9
1.1. Hệ cơ sở dữ liệu	9
1.2. Kiến trúc ba mức trừu tượng của một hệ cơ sở dữ liệu .	11
1.3. Lược đồ và thể hiện của cơ sở dữ liệu	12
1.4. Sự độc lập của dữ liệu	13
1.5. Các cách tiếp cận cơ sở dữ liệu	15
1.5.1. Mô hình dữ liệu logic	15
1.5.2. Mô hình dữ liệu vật lý	16
1.6. Hệ quản trị cơ sở dữ liệu	16
1.6.1. Khái niệm	16
1.6.2. Các chức năng của hệ quản trị cơ sở dữ liệu . . .	18
1.6.3. Kiến trúc của hệ quản trị cơ sở dữ liệu	19
1.7. Vai trò của con người trong hệ cơ sở dữ liệu	20
1.7.1. Người quản trị cơ sở dữ liệu	20
1.7.2. Người thiết kế cơ sở dữ liệu	21
1.7.3. Người lập trình ứng dụng	22
1.7.4. Người sử dụng đầu cuối	22
Bài tập Chương 1	23
Chương 2. Các mô hình dữ liệu	25
2.1. Mô hình dữ liệu thực thể-mối quan hệ	25
2.1.1. Mô hình dữ liệu bậc cao và quá trình thiết kế cơ sở dữ liệu	26
2.1.2. Các thành phần cơ bản của mô hình dữ liệu thực thể-mối quan hệ	28
2.1.3. Mối quan hệ <i>isa</i>	31
2.1.4. Ràng buộc trên các mối quan hệ	32
2.1.5. Sơ đồ thực thể-mối quan hệ	34

2.1.6.	Thiết kế lược đồ thực thể-mối quan hệ	35
2.1.7.	Một ví dụ về thiết kế lược đồ thực thể-mối quan hệ	36
2.2.	Mô hình dữ liệu quan hệ	39
2.2.1.	Quan hệ	40
2.2.2.	Các tính chất cơ bản của quan hệ	42
2.2.3.	Khóa của quan hệ	43
2.2.4.	Thuật toán tìm khóa của quan hệ	45
2.2.5.	Cập nhật dữ liệu trên quan hệ	51
2.2.6.	Đại số quan hệ	53
2.2.7.	Các phép toán quan hệ bổ sung	60
2.2.8.	Một số ví dụ về đại số quan hệ	62
2.2.9.	Phép tính quan hệ	64
2.2.10.	Khung nhìn	65
2.2.11.	Nhận xét về mô hình dữ liệu quan hệ	66
Bài tập Chương 2		69
Chương 3. Ngôn ngữ SQL		71
3.1.	Một số quy ước ký pháp	72
3.2.	Định nghĩa dữ liệu	74
3.2.1.	Các kiểu miền trong SQL	74
3.2.2.	Định nghĩa bảng trong SQL	75
3.2.3.	Tạo lập các chỉ mục	79
3.3.	Thao tác dữ liệu	80
3.3.1.	Cập nhật cơ sở dữ liệu	80
3.3.2.	Truy vấn cơ sở dữ liệu	82
3.4.	Cấp phát và thu hồi quyền truy cập cơ sở dữ liệu	99
3.5.	SQL nhúng	100
Bài tập Chương 3		103
Chương 4. Thiết kế cơ sở dữ liệu quan hệ		105
4.1.	Dư thừa dữ liệu và các dị thường cập nhật	105
4.2.	Phụ thuộc hàm	107
4.2.1.	Định nghĩa	107
4.2.2.	Suy diễn theo quan hệ	110

4.2.3.	Hệ tiên đề Armstrong	110
4.2.4.	Bao đóng của thuộc tính	114
4.2.5.	Một số thuật toán cơ bản	119
4.2.6.	Bao đóng của thuộc tính trên quan hệ và thuật toán	121
4.3.	Phủ của phụ thuộc hàm	124
4.3.1.	Định nghĩa	124
4.3.2.	Phủ không dư	126
4.3.3.	Phủ rút gọn	128
4.3.4.	Phủ tối thiểu	132
4.4.	Khóa của lược đồ quan hệ	134
4.4.1.	Định nghĩa	134
4.4.2.	Một số tính chất cơ bản	135
4.4.3.	Thuật toán tìm khóa	139
4.5.	Phản khóa của lược đồ quan hệ	143
4.5.1.	Định nghĩa	143
4.5.2.	Một số đặc trưng cơ bản	144
4.6.	Chuẩn hóa lược đồ quan hệ	146
4.6.1.	Phép tách lược đồ quan hệ	147
4.6.2.	Các dạng chuẩn của lược đồ quan hệ	153
4.6.3.	Một số đặc trưng cơ bản của các dạng chuẩn	158
4.6.4.	Chuẩn hóa lược đồ quan hệ qua phép tách	162
4.7.	Phụ thuộc đa trị và dạng chuẩn bốn	168
4.7.1.	Khái niệm	168
4.7.2.	Một số tính chất cơ bản	174
4.7.3.	Hệ tiên đề của phụ thuộc hàm và phụ thuộc đa trị	177
4.7.4.	Cơ sở phụ thuộc	183
4.7.5.	Chuẩn hóa lược đồ quan hệ về dạng chuẩn bốn	188
Bài tập Chương 4		193
Chương 5. Những vấn đề liên quan đến thiết kế		197
5.1.	Chuyển dịch lược đồ quan hệ	197
5.1.1.	Định nghĩa	197

5.1.2.	Bao đóng qua phép chuyển dịch lược đồ quan hệ	198
5.1.3.	Khóa qua phép chuyển dịch lược đồ quan hệ . . .	200
5.2.	Các mô tả tương đương của phụ thuộc hàm	204
5.2.1.	Toán tử bao đóng	205
5.2.2.	Hệ bao đóng	209
5.2.3.	Hàm chọn	211
5.3.	Họ các tập tối tiểu của thuộc tính	213
5.3.1.	Định nghĩa	214
5.3.2.	Thuật toán tìm họ các tập tối tiểu	214
5.3.3.	Họ các tập tối tiểu và quan hệ Armstrong	216
5.4.	Mở rộng phụ thuộc hàm	219
5.4.1.	Cơ sở tập thô	219
5.4.2.	Phụ thuộc cấp k	228
5.4.3.	Phụ thuộc hàm xấp xỉ	235
Bài tập Chương 5		249
Tài liệu tham khảo		253
Danh mục chữ viết tắt, ký hiệu		256
Chỉ mục		258

Chương 1.

KHÁI QUÁT VỀ CƠ SỞ DỮ LIỆU

Chương này chủ yếu trình bày về các khái niệm cơ bản của cơ sở dữ liệu như hệ cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu, mô hình dữ liệu, kiến trúc ba mức trừu tượng của hệ cơ sở dữ liệu, lược đồ và thể hiện cơ sở dữ liệu cũng như sự độc lập dữ liệu. Cuối cùng của chương trình bày về các vai trò của con người trong một hệ cơ sở dữ liệu.

1.1. Hệ cơ sở dữ liệu

Cơ sở dữ liệu (CSDL) là một tập hợp các dữ liệu có liên quan với nhau chứa thông tin về một tổ chức nào đó, chẳng hạn như trường đại học, ngân hàng, công ty, ... được lưu trữ trên các thiết bị nhớ thứ cấp nhằm đáp ứng nhu cầu khai thác thông tin của nhiều người sử dụng (NSD) với nhiều mục đích khác nhau.

Như vậy, CSDL phải phản ánh được thông tin về hoạt động của một tổ chức nhất định. Thông tin lưu trữ trong CSDL phải được chia sẻ cho nhiều NSD cũng như nhiều ứng dụng khác nhau. Trước đây khoảng đầu những năm 60, khi khái niệm CSDL chưa ra đời, mỗi chương trình ứng dụng đều có một tệp dữ liệu tương ứng và mỗi khi chương trình cần được sửa đổi thì tệp dữ liệu tương ứng cũng phải thay đổi theo. Việc lưu trữ thông tin của một tổ chức trong một hệ xử lý tệp dữ liệu như vậy có những nhược điểm chính như sau:

- *Dư thừa dữ liệu và dữ liệu không nhất quán*: một dữ liệu có thể được lưu trữ trong nhiều tệp dữ liệu khác nhau, khi đó việc cập nhật dữ liệu có thể sẽ bỏ sót và dẫn đến dữ liệu không nhất quán.
- *Truy cập dữ liệu khó khăn*: môi trường xử lý tệp dữ liệu truyền thống không cho phép dữ liệu được tìm kiếm theo cách thức thuận

tiện và hiệu quả.

- *Dữ liệu cô lập*: dữ liệu nằm rải rác trong nhiều tệp và các tệp có thể có cấu trúc khác nhau nên khó viết các chương trình ứng dụng mới để tìm các dữ liệu thích hợp.
- *Các vấn đề về toàn vẹn dữ liệu*: khi thêm các ràng buộc mới, khó có thể thay đổi các chương trình để có thể tuân thủ chúng.
- *Các vấn đề về tính nguyên tố của các giao dịch*: với hệ xử lý tệp dữ liệu truyền thống, khó có thể đảm bảo được tính chất “hoặc thực hiện hoàn toàn hoặc không thực hiện gì” và khó đưa được hệ thống trở về trạng thái ban đầu khi xảy ra sự cố.
- *Các vấn đề về an toàn dữ liệu*: thường thì mỗi NSD CSDL chỉ được phép truy cập một phần của CSDL, điều này làm cho dữ liệu trong CSDL được an toàn. Tuy nhiên, đối với hệ xử lý tệp dữ liệu truyền thống tính chất này thường không được đảm bảo.

Những vấn đề trên sẽ không còn tồn tại nữa khi khái niệm CSDL ra đời.

Một phần mềm cho phép NSD giao tiếp với CSDL, cung cấp một môi trường thuận lợi và hiệu quả để tìm kiếm và lưu trữ thông tin của CSDL được gọi là *hệ quản trị CSDL*. Một CSDL cùng với hệ quản trị CSDL để truy cập CSDL đó thường được gọi là *hệ CSDL*. Mục đích chính của một hệ CSDL là cung cấp cho NSD một cách nhìn trùu tượng về dữ liệu, nghĩa là hệ thống che dấu những chi tiết phức tạp về cách dữ liệu được lưu trữ và bảo trì. Chính vì vậy, trong cuộc sống hiện đại ngày nay việc sử dụng CSDL trở nên phổ biến và quen thuộc đến mức nhiều lúc NSD xem đó là hiển nhiên. Chẳng hạn, khi chúng ta muốn đặt chỗ cho chuyến bay sắp tới của mình, nhân viên đại lý bán vé hàng không sẽ nhanh chóng cung cấp những thông tin cần thiết giúp chúng ta có thể quyết định chọn chuyến bay để đăng ký và họ cũng sẽ ghi nhận sự đăng ký của chúng ta như một thông tin cập nhật vào tập hợp dữ liệu được lưu trữ. Hoặc khi đến thư viện tìm mượn sách, nhờ máy tính ít nhất chúng ta có thể biết được thông tin chi tiết về sách của thư viện,

thông tin về sách đã có người xếp hàng đặt mượn. Sự phát triển mạnh mẽ của Internet ở thập kỷ cuối thế kỷ 20 đã làm số người truy nhập và khai thác thông tin trong các CSDL tăng lên nhanh chóng. Với các giao diện Web, người ta có thể đăng ký các khóa học ở một trường đại học, có thể xem số dư trong tài khoản của mình ở một ngân hàng, có thể tìm hiểu chi tiết về một mặt hàng nào đó, ... càng ngày việc truy xuất thông tin trong các CSDL càng trở thành một bộ phận thiết yếu trong cuộc sống của mỗi người.

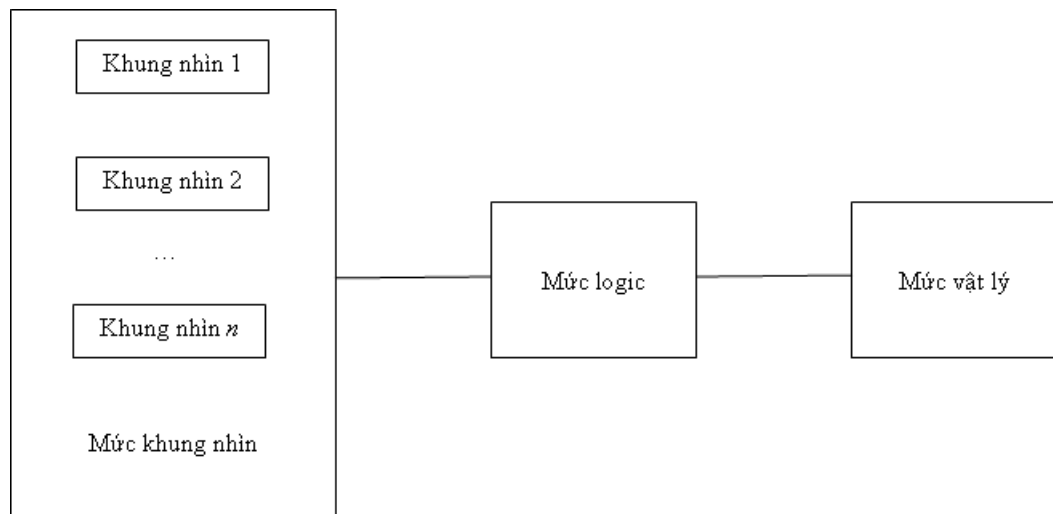
Vì nhiều NSD CSDL không thuộc giới chuyên tin, nên những người phát triển hệ thống đã che dấu không cho người dùng biết sự phức tạp của việc lưu trữ và bảo trì dữ liệu thông qua các mức trừu tượng hóa dữ liệu, nhằm làm đơn giản những tương tác của người dùng với hệ thống.

1.2. Kiến trúc ba mức trừu tượng của một hệ cơ sở dữ liệu

Theo ANSI-PARC (American National Standards Institute-Planning and Requirements Committee: Viện tiêu chuẩn quốc gia Mỹ-Ban nhu cầu và kế hoạch Mỹ), có ba mức trừu tượng hóa dữ liệu trong một hệ CSDL như sau:

- *Mức vật lý* (hay *mức trong*): mức này mô tả dữ liệu được thực sự lưu trữ như thế nào trong CSDL. Đây là mức trừu tượng thấp nhất, thể hiện các cài đặt có tính chất vật lý của CSDL. Nó cũng phản ánh các cấu trúc dữ liệu, các tổ chức tệp được dùng cho việc lưu trữ dữ liệu trên các thiết bị nhớ thứ cấp.
- *Mức logic* (hay *mức khái niệm*): cho biết dữ liệu nào được lưu trữ trong CSDL và các mối quan hệ giữa chúng. Cụ thể biểu diễn các thực thể, các thuộc tính và các mối quan hệ giữa các thực thể đó. Mức này chỉ quan tâm đến cái gì được lưu trữ trong CSDL chứ không quan tâm đến cách thức lưu trữ chúng.
- *Mức khung nhìn* (hay *mức ngoài*): mức này thể hiện một phần

CSDL mà NSD cần khai thác. Đây là mức trừu tượng cao nhất. Mức này bao gồm một số khung nhìn của NSD trong toàn bộ thông tin của hệ CSDL mà NSD cần quan tâm. Khung nhìn này gồm các thực thể, thuộc tính và các mối quan hệ giữa thực thể. Một số khung nhìn có thể chứa các dữ liệu suy diễn ra được hay tính toán được, những dữ liệu này vốn không được thực sự lưu trữ trong CSDL.



Hình 1.1: Ba mức trừu tượng hóa dữ liệu trong CSDL

Tóm lại, mức khung nhìn là cách cảm nhận của NSD về dữ liệu, mức vật lý là cách cảm nhận của hệ CSDL và hệ điều hành về dữ liệu. Cuối cùng mức logic là cách cảm nhận của toàn thể cộng đồng NSD về dữ liệu. Tại mức logic tồn tại hai ánh xạ đến hai mức còn lại, tạo nên một sự độc lập đối với nhau của hai mức đó.

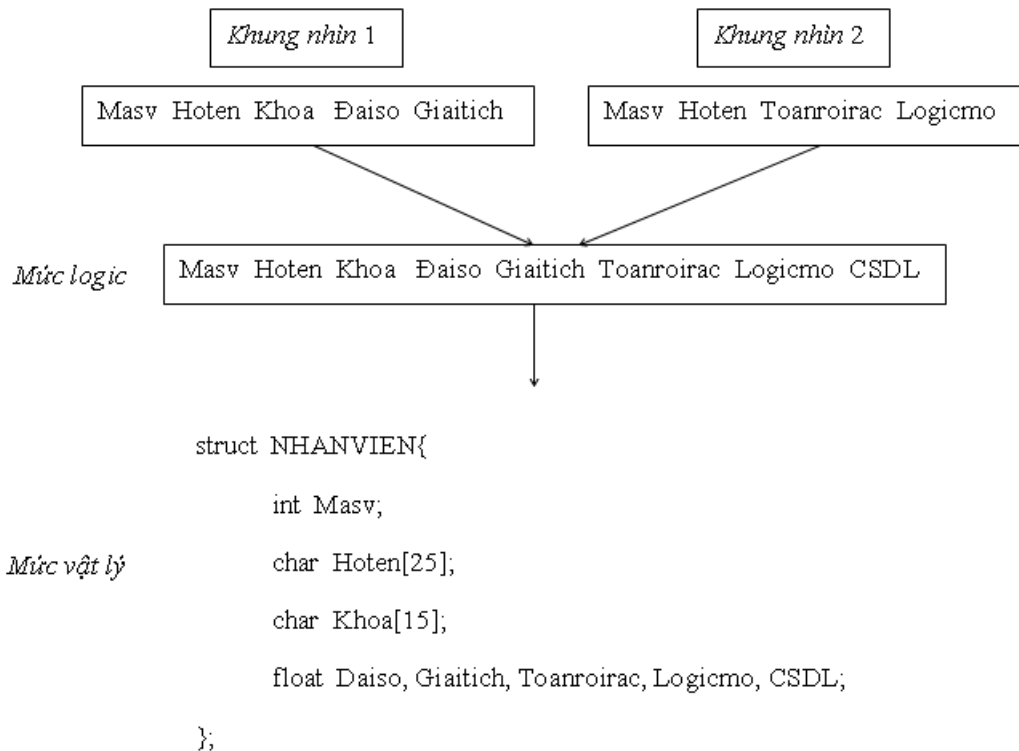
1.3. Lược đồ và thể hiện của cơ sở dữ liệu

Lược đồ CSDL là toàn bộ mô tả của CSDL. Tương ứng với ba mức trừu tượng hóa dữ liệu chúng ta có ba loại lược đồ. Ở mức khung nhìn ta có nhiều *lược đồ con* thường gọi là *lược đồ ngoài*, ở mức logic ta có *lược đồ*

khái niệm và cuối cùng ở mức vật lý ta có *lược đồ trong*. Thường thì các hệ CSDL hỗ trợ một lược đồ trong, một lược đồ khái niệm và nhiều lược đồ con.

Lược đồ CSDL được xác định trong quá trình thiết kế CSDL và thông thường người ta không muốn nó thường xuyên thay đổi. Trong khi đó, bản thân CSDL sẽ thay đổi theo thời gian do dữ liệu thêm vào, xóa đi hay sửa đổi. Toàn bộ dữ liệu được lưu trữ trong CSDL tại một thời điểm nhất định được gọi là một *thể hiện* của CSDL. Như vậy, với một lược đồ CSDL có thể có cùng nhiều thể hiện của CSDL.

Ví dụ 1.1. Sau đây là các lược đồ trong CSDL:



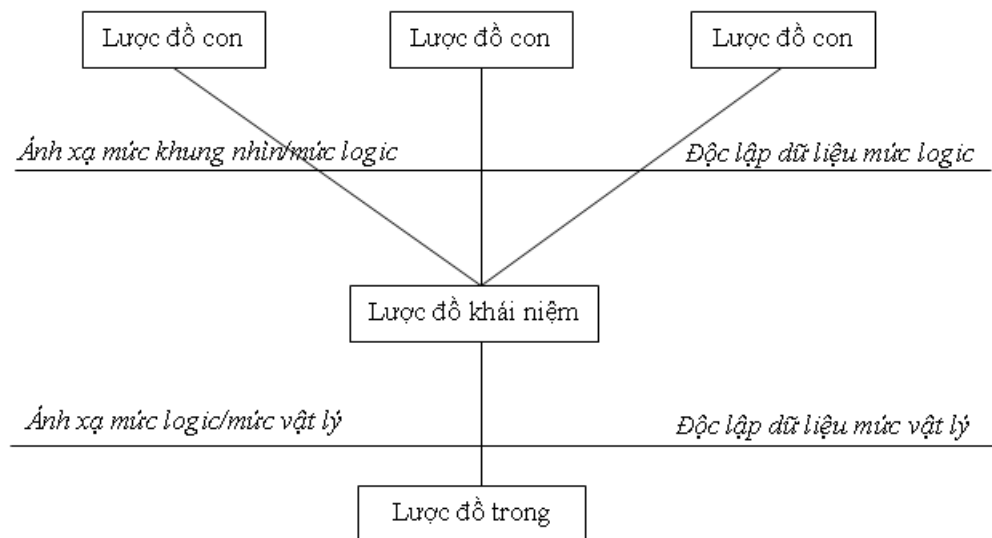
1.4. Sự độc lập của dữ liệu

Mục đích của kiến trúc ba mức trừu tượng hóa dữ liệu trong CSDL còn có thể hiểu theo nghĩa là sự độc lập của dữ liệu, đó là các lược đồ ở mức

trên không bị ảnh hưởng khi có sự thay đổi các lược đồ ở mức dưới. Chúng ta có hai loại độc lập dữ liệu: độc lập dữ liệu vật lý và độc lập dữ liệu logic.

- *Độc lập dữ liệu vật lý* là khả năng sửa đổi lược đồ trong mà không làm thay đổi lược đồ khái niệm và do đó không đòi hỏi phải viết lại các chương trình ứng dụng. Để tăng hiệu quả nhiều khi chúng ta cần có những thay đổi ở mức vật lý, chẳng hạn tổ chức tệp khác trước, thay đổi các chỉ mục hay thay đổi thuật toán băm, ...
- *Độc lập dữ liệu logic* là khả năng sửa đổi lược đồ khái niệm mà không làm thay đổi các lược đồ ngoài (khung nhìn) và do đó cũng có nghĩa không đòi hỏi phải viết lại các chương trình ứng dụng. Các sửa đổi ở mức logic là cần thiết mỗi khi cấu trúc logic của CSDL cần phải thay đổi, chẳng hạn cần thêm hay bớt một thực thể nào đó, các thuộc tính hay các mối quan hệ của chúng.

Sau đây là kiến trúc ba mức ANSI-SPARC về sự độc lập dữ liệu:



Hình 1.2: Kiến trúc ba mức về sự độc lập dữ liệu

Độc lập dữ liệu logic khó thực hiện hơn độc lập dữ liệu vật lý, vì các chương trình ứng dụng phụ thuộc nhiều vào cấu trúc logic của dữ liệu.

liệu mà chúng truy cập. Khái niệm độc lập dữ liệu trong nhiều khía cạnh nó tương tự với khái niệm *kiểu dữ liệu trừu tượng* trong các ngôn ngữ lập trình hiện đại. Đó là cả hai đều che dấu NSD những chi tiết cài đặt, chỉ cho phép NSD tập trung vào cấu trúc chung hơn là tập trung vào chi tiết cài đặt ở mức thấp.

1.5. Các cách tiếp cận cơ sở dữ liệu

Trên thực tế, một lược đồ được viết trong ngôn ngữ định nghĩa dữ liệu của một hệ quản trị CSDL cụ thể. Tuy nhiên, để mô tả các yêu cầu về dữ liệu của một tổ chức sao cho mô tả đó dễ hiểu đối với NSD thì ngôn ngữ đó lại ở mức quá thấp. Điều này dẫn đến cần phải có mô tả lược đồ ở mức cao hơn, mô tả như thế này được hiểu là một mô hình dữ liệu. Một cách hình thức *mô hình dữ liệu* là một hệ toán học gồm hai phần:

- (i) Một tập hợp các ký hiệu để mô tả dữ liệu.
- (ii) Một tập hợp các phép toán thao tác trên dữ liệu đó.

Mô hình dữ liệu (MHDL) được sử dụng trong việc thiết kế CSDL với mục đích biểu diễn dữ liệu sao cho dễ hiểu. Đã có nhiều MHDL được đề xuất, người ta thường chia làm hai nhóm: MHDL logic và MHDL vật lý. Trong đó, MHDL logic tập trung vào bản chất logic của biểu diễn dữ liệu, tập trung vào cái được biểu diễn trong CSDL và thường xem là các MHDL bậc cao. Còn MHDL vật lý tập trung vào những chi tiết cho biết dữ liệu được lưu trữ như thế nào và thường được xem là các MHDL bậc thấp.

1.5.1. Mô hình dữ liệu logic

MHDL logic được dùng trong việc mô tả dữ liệu ở các mức logic và mức khung nhìn. Người ta thường chia MHDL logic làm hai loại: MHDL logic dựa trên cơ sở đối tượng và MHDL logic dựa trên cơ sở bản ghi. Trong đó, MHDL logic dựa trên cơ sở đối tượng cung cấp các khả năng cấu trúc rất mềm dẻo và cho phép các ràng buộc được đặc tả một cách

tường minh. Các MHDL này thường được hay nói đến là MHDL thực thể-mối quan hệ, MHDL đối tượng, MHDL ngữ nghĩa và MHDL chức năng. Còn MHDL logic dựa trên cơ sở bản ghi được dùng để đặc tả cấu trúc logic tổng thể của CSDL, đồng thời cung cấp một mô tả ở mức cao hơn của sự cài đặt. MHDL này đơn giản hơn so với MHDL logic dựa trên cơ sở đối tượng, và thường bao gồm các MHDL là MHDL quan hệ, MHDL mạng và MHDL phân cấp.

1.5.2. Mô hình dữ liệu vật lý

MHDL vật lý được dùng trong việc mô tả dữ liệu ở mức vật lý, cụ thể MHDL vật lý mô tả dữ liệu được lưu trữ trong máy tính như thế nào, mô tả các cấu trúc bản ghi, thứ tự các bản ghi và cách truy cập. Các MHDL vật lý thường chỉ có ý nghĩa với các chuyên gia máy tính, không cần thiết đối với NSD thông thường.

1.6. Hệ quản trị cơ sở dữ liệu

1.6.1. Khái niệm

Hệ quản trị CSDL là một phần mềm cho phép NSD tương tác với CSDL, một cách hình thức hệ quản trị CSDL chính là một tập hợp các chương trình cho phép NSD định nghĩa, tạo lập, bảo trì các CSDL và cung cấp các truy nhập có điều khiển đến các CSDL này. Để làm điều này hệ quản trị CSDL cung cấp các công cụ sau:

(1) Ngôn ngữ định nghĩa dữ liệu: cho phép NSD định nghĩa CSDL, đó là đặc tả các kiểu và các cấu trúc dữ liệu, đặc tả các ràng buộc trên các dữ liệu lưu trữ trong CSDL. Kết quả chúng ta có một tập các bảng được lưu trữ trong một tệp đặc biệt thường được gọi là từ điển dữ liệu.

(2) Ngôn ngữ thao tác dữ liệu: cho phép NSD thao tác hay truy nhập dữ liệu được dễ dàng hơn. Các thao tác dữ liệu bao gồm:

- Tìm kiếm thông tin lưu trữ trong CSDL.

- Thêm thông tin mới vào CSDL.
- Xoá thông tin khỏi CSDL.
- Thay đổi thông tin được lưu trữ trong CSDL.

Có hai kiểu ngôn ngữ thao tác dữ liệu là thủ tục và phi thủ tục. Ngôn ngữ thao tác dữ liệu kiểu thủ tục đòi hỏi NSD vừa xác định dữ liệu họ cần, vừa phải xác định cách thức để có được dữ liệu đó. Còn ngôn ngữ thao tác dữ liệu kiểu phi thủ tục chỉ yêu cầu NSD xác định dữ liệu họ cần, chứ không yêu cầu NSD xác định cách thức để có được dữ liệu đó. Các ngôn ngữ thao tác dữ liệu phi thủ tục dễ học và sử dụng hơn ngôn ngữ thao tác dữ liệu thủ tục. Tuy nhiên, vì NSD không phải xác định cách thức nào để có được dữ liệu nên các ngôn ngữ này có thể sinh mã không hiệu quả như mã được sinh bởi ngôn ngữ thao tác dữ liệu thủ tục. Để khắc phục những khó khăn này, người ta thường dùng đến các kỹ thuật tối ưu hóa câu hỏi. Ngôn ngữ thông dụng nhất của kiểu ngôn ngữ thao tác dữ liệu phi thủ tục là ngôn ngữ SQL.

(3) Các kiểm soát, các điều khiển đối với việc truy nhập CSDL, chẳng hạn như:

- Hệ thống an ninh: ngăn cấm sự cố tình truy nhập vào CSDL một cách không được phép.
- Hệ thống ràng buộc toàn vẹn: duy trì tính nhất quán của dữ liệu được lưu trữ.
- Hệ thống điều khiển tương tranh: cho phép các truy nhập xảy ra đồng thời.
- Hệ thống tự thích ứng: cho phép khôi phục CSDL khi có sự cố về phần cứng hay phần mềm.
- Từ điển dữ liệu: NSD có thể truy nhập được.

Các hệ quản trị CSDL hiện nay có trên thị trường là Dbase, Sybase, Informix, SQL Server, Oracle, Access, Foxpro, Paradox, ...

1.6.2. Các chức năng của hệ quản trị cơ sở dữ liệu

Một hệ quản trị CSDL thường cung cấp các chức năng cơ bản sau:

- Cung cấp cho NSD khả năng lưu trữ, truy xuất và cập nhật CSDL. Khi đó hệ quản trị CSDL đã che dấu NSD những chi tiết cài đặt mang tính chất vật lý.
- Cung cấp cho NSD một từ điển dữ liệu, đó là các mô tả về dữ liệu được lưu trữ và NSD truy cập được vào từ điển dữ liệu này. Dữ liệu trong từ điển dữ liệu là dữ liệu về dữ liệu nên còn được gọi là *siêu dữ liệu*. Thông tin trong từ điển dữ liệu và cách thức sử dụng thông tin này là khác nhau trong các hệ quản trị CSDL khác nhau. Thông thường, một từ điển dữ liệu cất giữ tên, kiểu, kích thước các bản ghi, tên của các mối quan hệ, các ràng buộc toàn vẹn trên dữ liệu, tên những NSD truy cập vào CSDL, các lược đồ trong, lược đồ khái niệm, lược đồ ngoài và các ánh xạ giữa chúng ...
- Hỗ trợ các giao dịch bằng cách cung cấp một cơ chế đảm bảo rằng hoặc tất cả các cập nhật trong một giao dịch làm việc được thực hiện hoặc không có thao tác cập nhật nào trong giao dịch này được thực hiện, để đảm bảo tính nhất quán của dữ liệu.
- Cung cấp các dịch vụ điều khiển tương tranh để đảm bảo tính nhất quán của dữ liệu khi có nhiều phiên làm việc với CSDL, có nhiều người đồng thời truy nhập vào CSDL, đặc biệt những truy nhập làm thay đổi thông tin lưu trữ trong đó.
- Cung cấp cơ chế để khôi phục dữ liệu khi xảy ra một sự cố làm hỏng CSDL theo một kiểu nào đó.
- Cung cấp các dịch vụ bản quyền.
- Hỗ trợ cho truyền thông dữ liệu, tức là hệ quản trị CSDL phải có khả năng tích hợp được với các phần mềm truyền thông.
- Cung cấp các dịch vụ đảm bảo tính toàn vẹn dữ liệu để đảm bảo sự chính xác và nhất quán của dữ liệu được lưu trữ.

Một số chức năng khác của hệ quản trị CSDL mà NSD thường quan

tâm mong muốn có thêm:

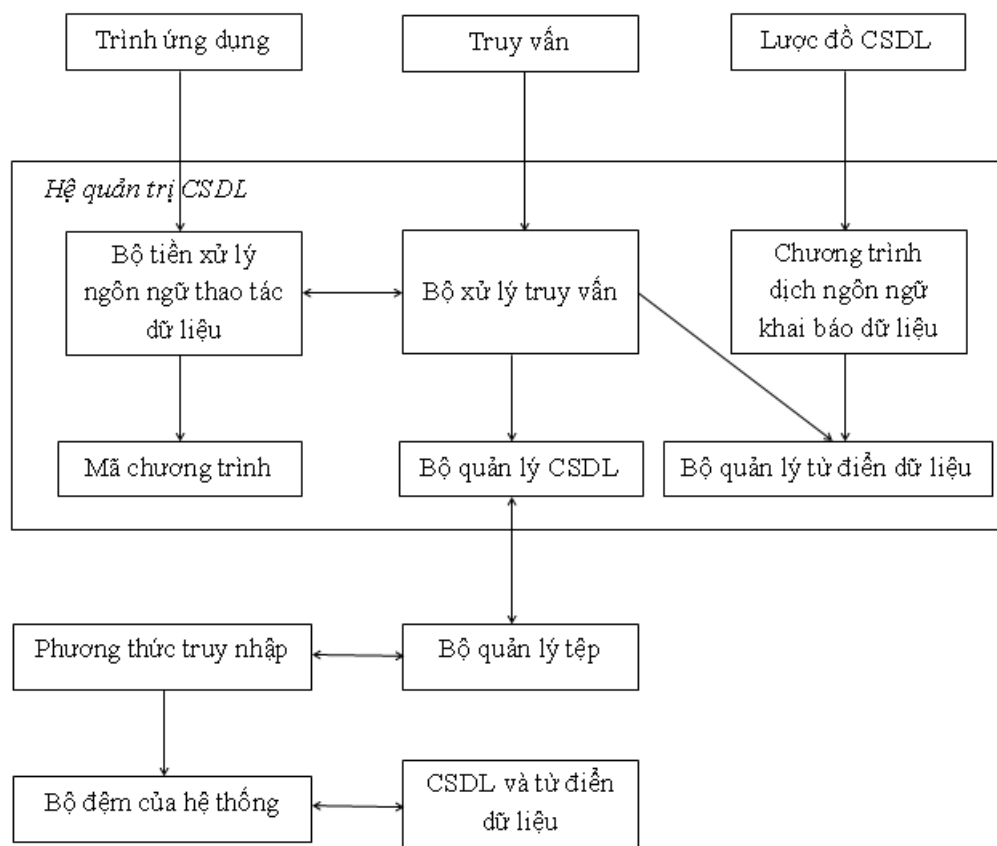
- Cung cấp các dịch vụ hỗ trợ cho tính độc lập dữ liệu.
- Cung cấp một số dịch vụ tiện ích giúp người quản trị CSDL quản lý CSDL được thuận tiện và hiệu quả hơn.

1.6.3. Kiến trúc của hệ quản trị cơ sở dữ liệu

Để có thể cung cấp các dịch vụ nói trên, hệ quản trị CSDL được chia làm một số thành phần, trong đó mỗi thành phần là một mô đun phần mềm chịu trách nhiệm cụ thể. Chú ý, hệ điều hành chỉ cung cấp một số dịch vụ cơ bản nhất và hệ quản trị CSDL phải xây dựng trên nó, do đó việc thiết kế một hệ quản trị CSDL phải tạo ra các giao diện giữa hệ quản trị CSDL và hệ điều hành.

Sau đây là các thành phần chính của một hệ quản trị CSDL, với một số thành phần chính của hệ quản trị CSDL có chức năng như sau:

- Bộ xử lý truy vấn: dịch các câu lệnh trong ngôn ngữ truy vấn thành các câu lệnh ở mức thấp chuyển cho bộ quản lý CSDL.
- Bộ quản lý CSDL: giao tiếp với các chương trình ứng dụng của NSD và các câu lệnh truy vấn, đó là nó tiếp nhận các yêu cầu truy vấn và kiểm tra các lược đồ con và lược đồ khái niệm để xác định những bản ghi logic nào đã được yêu cầu.
- Bộ tiền xử lý ngôn ngữ thao tác dữ liệu: chuyển các câu lệnh ngôn ngữ thao tác dữ liệu nhúng trong một chương trình ứng dụng thành các lời gọi thủ tục chuẩn trong ngôn ngữ chủ.
- Bộ biên dịch ngôn ngữ khai báo dữ liệu: chuyển các câu lệnh ngôn ngữ khai báo dữ liệu sang thành một tập các bảng chứa siêu dữ liệu, các bảng này được lưu trữ trong từ điển dữ liệu.
- Bộ quản lý từ điển dữ liệu: quản lý các truy nhập đến từ điển dữ liệu và bảo trì hệ thống từ điển dữ liệu.



Hình 1.3: *Kiến trúc một hệ quản trị CSDL*

1.7. Vai trò của con người trong hệ cơ sở dữ liệu

Với các CSDL nhỏ chỉ cần một người tạo lập và thao tác trên đó, nhưng với CSDL lớn thì có thể cần phải có sự tham gia của nhiều người vào việc xây dựng, bảo trì và sử dụng nó. Những người này bao gồm người quản trị CSDL, người thiết kế CSDL, người lập trình ứng dụng và NSD đầu cuối.

1.7.1. Người quản trị cơ sở dữ liệu

Đây là người có trách nhiệm quản lý các tài nguyên của hệ CSDL (CSDL, hệ quản trị CSDL và các phần mềm liên quan) cụ thể như: thiết kế và

cài đặt CSDL về mặt vật lý, cấp phát quyền truy nhập CSDL, cấp phần mềm và phần cứng theo yêu cầu, duy trì các hoạt động hệ thống đảm bảo thỏa mãn yêu cầu của các trình ứng dụng và NSD.

Như vậy, người quản trị CSDL phải là người có những hiểu biết chi tiết và kỹ năng về mặt kỹ thuật trong lĩnh vực CSDL, hệ quản trị CSDL và môi trường hệ thống.

1.7.2. Người thiết kế cơ sở dữ liệu

Trong các dự án thiết kế các CSDL lớn, người ta thường chia các người thiết CSDL làm hai nhóm: nhóm người thiết kế CSDL logic và nhóm người thiết kế CSDL vật lý.

- Người thiết kế CSDL logic: có trách nhiệm xác định dữ liệu lưu trữ trong CSDL (các thực thể và thuộc tính), xác định mối quan hệ giữa dữ liệu, các ràng buộc trên dữ liệu. Như vậy, người thiết kế CSDL logic phải am hiểu về dữ liệu của tổ chức và các luật làm việc của tổ chức đó. Do đó, để quá trình thiết kế có hiệu quả, người thiết kế CSDL logic phải giao tiếp với những NSD CSDL trong tương lai, hiểu được nhu cầu sử dụng của họ để có thể đưa ra một thiết kế phù hợp.
- Người thiết kế CSDL vật lý: lựa chọn MHDL logic và quyết định nó được thực hiện về mặt vật lý như thế nào. Như vậy, người thiết kế CSDL vật lý liên quan đến những vấn đề như ánh xạ MHDL logic vào tập các bảng và các ràng buộc toàn vẹn, lựa chọn cấu trúc lưu trữ và phương thức truy nhập dữ liệu để đạt hiệu quả cao khi thực hiện các thao tác trên CSDL, thiết kế các hệ thống an ninh cho dữ liệu.

Tóm lại, có thể hiểu người thiết kế CSDL logic liên quan đến câu hỏi “cái gì được lưu trữ trong CSDL” và người thiết kế CSDL vật lý liên quan đến câu hỏi “lưu trữ nó như thế nào”.

1.7.3. Người lập trình ứng dụng

Khi CSDL được cài đặt, người lập trình ứng dụng viết các chương trình ứng dụng phục vụ các nhu cầu khai thác CSDL của NSD. Thông thường, người lập trình ứng dụng thể hiện các đặc tả của người phân tích thiết kế hệ thống thành chương trình. Mỗi chương trình bao gồm các câu lệnh yêu cầu hệ quản trị CSDL thực hiện một số thao tác trên CSDL như truy xuất dữ liệu, thêm, xóa hay sửa đổi dữ liệu. Các chương trình ứng dụng này có thể được viết trong ngôn ngữ thế hệ ba hay thế hệ bốn.

1.7.4. Người sử dụng đầu cuối

NSD chính là khách hàng của CSDL bởi CSDL được thiết kế, cài đặt vào bảo trì để cung cấp những thông tin họ cần. Theo cách sử dụng CSDL, có thể chia NSD đầu cuối thành hai nhóm: nhóm NSD đơn giản và nhóm NSD tinh tế.

- NSD đơn giản: là người không có hiểu biết sâu về CSDL, họ truy cập vào CSDL thông qua các chương trình ứng dụng để thực hiện các thao tác đơn giản. Họ yêu cầu thực hiện các thao tác này bằng cách đưa vào những câu lệnh đơn giản hay lựa chọn các mục trên bảng chọn. Nói một cách khác họ sử dụng các giao dịch định sẵn.
- NSD tinh tế: là những người hiểu biết về cấu trúc CSDL, những tiện ích mà hệ quản trị CSDL cung cấp. Họ có thể sử dụng ngôn ngữ truy vấn bậc cao như SQL để thực hiện những thao tác cần thiết. Thậm chí một số người sử dụng tinh tế có thể viết thêm các chương trình ứng dụng thỏa mãn yêu cầu của họ.

BÀI TẬP CHƯƠNG 1

1.1. Phân biệt các khái niệm sau:

- Dữ liệu
- CSDL
- Hệ quản trị CSDL
- Hệ CSDL.

1.2. CSDL được ứng dụng rộng rãi trong cuộc sống. Hãy nêu một số ứng dụng khác của CSDL ngoài 2 ứng dụng trong lĩnh vực thư viện và hàng không.

1.3. Hãy nêu các đặc điểm khác nhau chính giữa một hệ xử lý tệp với một hệ quản trị CSDL.

1.4. Mô tả các thành phần chính của một hệ quản trị CSDL và chỉ ra các chức năng của chúng.

1.5. Giải thích khái niệm độc lập dữ liệu và sự quan trọng của tính độc lập dữ liệu trong môi trường của một hệ CSDL. Hãy giải thích sự khác nhau giữa tính độc lập dữ liệu vật lý và độc lập dữ liệu logic.

1.6. Cho biết mục đích của việc mô hình hóa các CSDL. Nêu các cách tiếp cận khác nhau trong việc mô hình hóa CSDL. Chỉ ra sự khác nhau giữa MHDL logic và MHDL vật lý.

1.7. Vì sao các hệ CSDL theo MHDL quan hệ được sử dụng phổ biến.

1.8. Phân biệt vai trò của những người sau đây đối với một hệ CSDL:

- Người quản trị CSDL
- Người thiết kế CSDL logic
- Người thiết kế CSDL vật lý
- Người lập trình ứng dụng
- NSD đầu cuối.

Chương 2.

CÁC MÔ HÌNH DỮ LIỆU

Như chúng ta đã biết ở Chương 1, MHDL được dùng như một hệ thống các ký hiệu cho dữ liệu trong một CSDL và làm hệ thống ký hiệu cho ngôn ngữ thao tác dữ liệu. Ngoài ra, MHDL giúp giải quyết vấn đề dư thừa dữ liệu, đó là nó giúp NSD tránh lưu trữ cùng một giá trị quá một lần. Vì nếu không nó sẽ làm lãng phí bộ nhớ và cũng như đồng thời làm dữ liệu mất tính nhất quán (giá trị có thể thay đổi ở chỗ này nhưng lại không thay đổi ở chỗ khác).

Có nhiều MHDL đã đề xuất, tuy nhiên phải khẳng định rằng không có MHDL nào được xem là tốt nhất. Điều quan trọng là chúng được sử dụng tốt nhất ở đâu và vào lúc nào. Chương này sẽ giới thiệu các MHDL cơ bản như MHDL thực thể-mối quan hệ, MHDL quan hệ. Hai MHDL mạng và MHDL phân cấp hiện nay rất ít được sử dụng và ngoài ra cũng dễ dàng chuyển đổi giữa hai MHDL này với MHDL quan hệ, do đó trong chương này chúng ta sẽ không trình bày về hai MHDL này.

2.1. Mô hình dữ liệu thực thể-mối quan hệ

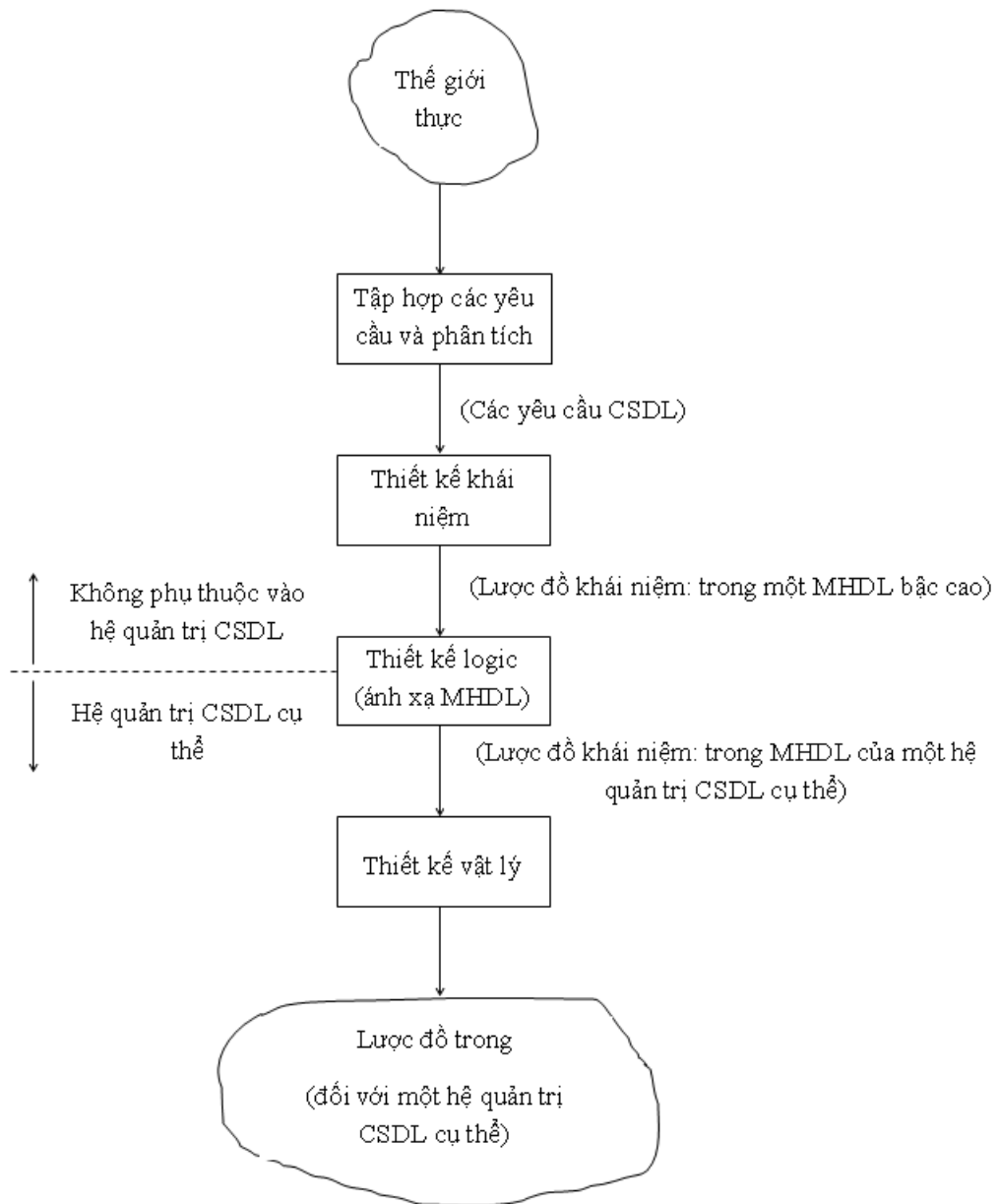
MHDL thực thể-mối quan hệ hay còn gọi *MHDL ER* (Entity-Relationship data model). Đây là MHDL khái niệm bậc cao hỗ trợ cho việc thiết kế CSDL. Nhiều công cụ thiết kế CSDL đã sử dụng khái niệm của MHDL này. MHDL thực thể-mối quan hệ dựa trên cơ sở nhận thức của thế giới thực, đó là nó cố gắng biểu diễn các ngữ nghĩa của các dữ liệu trong thế giới thực. Vì vậy, nó còn được xem như là MHDL ngữ nghĩa.

2.1.1. Mô hình dữ liệu bậc cao và quá trình thiết kế cơ sở dữ liệu

Quá trình thiết kế một CSDL thường được chia làm bốn bước như sau:

1. Tập hợp các yêu cầu và phân tích. Kết quả của bước này là một tập hợp các yêu cầu của NSD được ghi lại dưới dạng súc tích, với đặc tả càng chi tiết và đầy đủ thì càng tốt.
2. Thiết kế khái niệm. Ở bước này, người thiết kế lựa chọn một MHDL và dùng các khái niệm của MHDL này để chuyển những đặc tả yêu cầu của NSD sang thành một lược đồ khái niệm, với lược đồ khái niệm là một mô tả cô đọng về yêu cầu dữ liệu của NSD như mô tả chi tiết các kiểu dữ liệu, các liên kết và các ràng buộc. Lược đồ khái niệm bậc cao sẽ đảm bảo kết quả của quá trình thiết kế CSDL đáp ứng tất cả các yêu cầu của NSD và những yêu cầu đó là không mâu thuẫn. Trong bước này người thiết kế chỉ quan tâm đến việc đặc tả các tính chất của dữ liệu mà chưa cần quan tâm về các chi tiết lưu trữ.
3. Thiết kế logic hay còn gọi ánh xạ MHDL. Bước này người thiết kế cài đặt CSDL bằng một hệ quản trị CSDL. Hầu hết các hệ quản trị CSDL dùng một MHDL thể hiện (chẳng hạn MHDL quan hệ hay MHDL hướng đối tượng, ...). Do đó lược đồ khái niệm được chuyển đổi từ MHDL bậc cao sang MHDL thể hiện. Kết quả của bước này là một lược đồ CSDL dưới dạng một MHDL thể hiện của hệ quản trị CSDL.
4. Thiết kế vật lý. Bước này đặc tả về mặt vật lý của CSDL như các cấu trúc lưu trữ bên trong và kiểu tổ chức tệp cho CSDL.

Có thể hình dung quá trình thiết kế một CSDL qua Hình 2.1 sau:



Hình 2.1: *Quá trình thiết kế một CSDL*

2.1.2. Các thành phần cơ bản của mô hình dữ liệu thực thể-mối quan hệ

MHDL thực thể-mối quan hệ cho phép chúng ta thiết kế lược đồ khái niệm của một tổ chức mà không cần quan tâm đến tính hiệu quả hoặc vấn đề thiết kế CSDL ở mức vật lý. Sau đó, nó được cài đặt trong một MHDL của một hệ quản trị CSDL nào đó. Điều này dẫn đến MHDL thực thể-mối quan hệ không có hệ thống ký hiệu cho các phép toán trên dữ liệu, thậm chí có tác giả còn đề nghị không xem nó là một MHDL. MHDL thực thể-mối quan hệ sử dụng các đối tượng cơ bản sau: tập thực thể và tập mối quan hệ.

2.1.2.1. Tập thực thể

Thực thể là một sự vật hay một đối tượng trong thế giới thực tồn tại và phân biệt được. Các đặc tính xác định một thực thể được gọi là các *thuộc tính* của thực thể đó. Với mỗi thuộc tính ta có một tập các giá trị tương ứng. Mỗi thực thể cụ thể sẽ có một giá trị cho các thuộc tính của nó. Các giá trị thuộc tính mô tả mỗi thực thể sẽ là một phần chính của các dữ liệu được lưu trữ trong CSDL. Lưu ý, không có định nghĩa hình thức cho khái niệm thực thể, chúng được định nghĩa bằng cách mô tả về các đặc trưng của chúng. Tính phân biệt ở trên chính là khái niệm cơ bản của thực thể, được hiểu như là đặc tính nhận dạng của các đối tượng.

Chẳng hạn, mỗi sinh viên là một thực thể, các môn học hay các dự án là các thực thể. Với thực thể sinh viên có thể mô tả bởi các thuộc tính như *mã số sinh viên, họ tên, giới tính, ngày sinh, quê quán, khoa, số điện thoại*. Giá trị tương ứng của các thuộc tính là “K1001”, “Nguyễn Tiến Trung”, “Nam”, “10/05/1995”, “Huế”, “Toán”, “0914111222”. Như vậy, chúng ta đã có một thực thể sinh viên cùng với các giá trị thuộc tính của nó.

Một tập hợp các thực thể cùng kiểu, nghĩa là cùng được thể hiện

bởi một tập đặc trưng hay thuộc tính được gọi là *tập thực thể*. Một tập thực thể thường được tham chiếu đến bằng cách dùng tên.

Chẳng hạn, tập tất cả các sinh viên trường Đại học Khoa học Huế là một tập thực thể, trong đó mỗi sinh viên là một thực thể và các thực thể sinh viên này đều được thể hiện qua tập các thuộc tính (như *mã số sinh viên, họ tên, giới tính, ngày sinh, quê quán, khoa, số điện thoại*), tuy nhiên mỗi thực thể có các giá trị riêng cho các thuộc tính. Tập thực thể gồm các thực thể sinh viên này có thể được tham chiếu bằng tên như *SINHVIEN*. Như vậy, tên *SINHVIEN* vừa là tên của một tập thực thể chỉ đến tập hợp hiện tại của tất cả các thực thể sinh viên trong CSDL đang đề cập.

Sự mô tả một tập thực thể được gọi là một *lược đồ tập thực thể*, tức là nó đặc tả một cấu trúc chung của các thực thể trong tập thực thể này. Như vậy, một lược đồ tập thực thể xác định tên tập thực thể, tên và ngữ nghĩa của mỗi thuộc tính, và mọi ràng buộc mà các thực thể thuộc tập đó phải tuân theo. Ngoài ra, trong một CSDL có thể xác định nhiều tập thực thể và các tập thực thể này không nhất thiết phải phân biệt (theo nghĩa các tập thực thể khác nhau có thể có chung các thực thể). Chẳng hạn, xét tập thực thể *SINHVIEN* ở trên và tập thực thể *CANBO* gồm các thực thể là cán bộ trong trường, khi đó có thể có một thực thể cán bộ nào đó (ở phòng ban) đồng thời cũng là thực thể sinh viên (học tại chức). Lưu ý, lược đồ của tập thực thể sẽ không thay đổi thường xuyên bởi nó mô tả cấu trúc của các thực thể cụ thể.

Trong MHDL thực thể-mối quan hệ, khái niệm thuộc tính được chia làm nhiều loại như thuộc tính đơn, thuộc tính phức hợp, thuộc tính đơn trị và thuộc tính đa trị.

- *Thuộc tính đơn* là thuộc tính không thể phân chia được thành những thành phần nhỏ hơn. Chẳng hạn, thuộc tính mã số sinh viên của tập thực thể *SINHVIEN* ở trên là thuộc tính đơn.
- *Thuộc tính phức hợp* là thuộc tính có thể phân chia thành những thành phần nhỏ hơn. Chẳng hạn, thuộc tính họ tên của tập thực

thể *SINHVIEN* ở trên có thể phân chia thành họ, tên đệm và tên. Do đó, nó là thuộc tính phức hợp. Trên thực tế, trong nhiều tính huống việc sử dụng thuộc tính phức hợp trong thiết kế lược đồ là rất hữu ích, đặc biệt khi NSD chỉ cần tham khảo đến một phần của giá trị đó.

- *Thuộc tính đơn trị* là thuộc tính có một giá trị duy nhất cho một thực thể cụ thể. Chẳng hạn, thuộc tính mã số sinh viên hay thuộc tính giới tính của tập thực thể *SINHVIEN* là các thuộc tính đơn trị.
- *Thuộc tính đa trị* là thuộc tính có thể có một tập hợp các giá trị cho một thực thể. Chẳng hạn, thuộc tính số điện thoại của tập thực thể *SINHVIEN* có thể là thuộc tính đa trị bởi một sinh viên có thể có nhiều hơn một số điện thoại để liên lạc.

Ngoài ra, trong một số trường hợp, một số thuộc tính có liên quan đến nhau theo kiểu biết giá trị của thuộc tính *a* thì suy ra được giá trị của thuộc tính *b*. Khi đó *a* còn được gọi là *thuộc tính lưu trữ*, *b* còn được gọi là *thuộc tính suy diễn*. Chẳng hạn, với *a* là thuộc tính ngày sinh và *b* là thuộc tính tuổi.

MHDL thực thể-mối quan hệ cũng cho phép một thực thể cụ thể tại một thời điểm nào đó có thể thiếu một giá trị cho một thuộc tính nào đó. Giá trị thiếu này sẽ được ký hiệu bằng *null*.

Thuộc tính của tập thực thể mà giá trị của nó xác định duy nhất các thực thể trong tập thực thể được gọi là *khóa tối thiểu* của tập thực thể đó. Ràng buộc về tính duy nhất trên các thuộc tính như thế này còn được gọi là ràng buộc khóa. Trong một số trường hợp khóa tối thiểu có thể là một tập hợp gồm nhiều thuộc tính. Cũng có thể một tập thực thể có nhiều hơn một khóa tối thiểu. Chẳng hạn, thuộc tính *mã số sinh viên* là khóa tối thiểu của tập thực thể *SINHVIEN*. Về nguyên tắc, chúng ta luôn giả thiết mỗi thực thể đều có thể phân biệt được với các thực thể khác, do đó mỗi tập thực thể phải có ít nhất một khóa tối thiểu khác rỗng.

2.1.2.2. Tập mối quan hệ

Một mối quan hệ giữa các tập thực thể là một sự kết hợp nào đó giữa các thực thể của các tập thực thể đó. Một cách hình thức mối quan hệ được định nghĩa như sau. Cho k tập thực thể $E_1, E_2, \dots, E_k$. Một *mối quan hệ* R giữa các tập thực thể $E_1, E_2, \dots, E_k$ là một danh sách có thứ tự của các tập thực thể này. Tức là $R = (F_1, F_2, \dots, F_k)$ trong đó $F_i \in \{E_1, E_2, \dots, E_k\}$, $i = 1, 2, \dots, k$. Một k -bộ $(e_1, e_2, \dots, e_k)$, với $e_i \in F_i$ ($i = 1, 2, \dots, k$), được gọi là một *thể hiện* của R . Số nguyên dương k được gọi là *cấp* của mối quan hệ R . Phần nhiều các mối quan hệ đều có cấp 2, tuy nhiên cũng có những mối quan hệ cấp lớn hơn 2. Trong thực tế, người ta hạn chế dùng những mối quan hệ có cấp lớn hơn 3.

Chẳng hạn, xét hai tập thực thể *NHANVIEN* (nhân viên) và *PHONG* (phòng). Giả sử có một mối quan hệ cấp 2 *LAMVIECCHO* (làm việc cho) giữa hai tập thực thể này theo nghĩa một thực thể nhân viên sẽ kết hợp với một thực thể phòng, chẳng hạn thực thể nhân viên e_1, e_3 làm việc cho thực thể phòng d_1 , thực thể nhân viên e_2 làm việc cho thực thể phòng d_2 , ...

Một mối quan hệ cũng có thể có các thuộc tính để mô tả nó. Chẳng hạn, xét mối quan hệ *GUI TIEN* (gửi tiền) giữa hai tập thực thể *KHACHHANG* (khách hàng) và *TAIKHOAN* (tài khoản). Khi đó dễ thấy mối quan hệ *GUI TIEN* cần có thuộc tính *ngày truy cập* để ghi nhận lần cuối cùng (tức ngày gần nhất) khách hàng truy cập vào tài khoản này.

2.1.3. Mối quan hệ *isa*

Một tập thực thể có thể chứa một số tập con các thực thể phân biệt với các thực thể khác theo một cách nào đó. Cụ thể hơn, một tập con của một tập thực thể có thể có thêm một số thuộc tính mà những thực thể ngoài tập con này là không có. Việc thiết kế các nhóm con trong một

tập thực thể như thế này được gọi là sự đặc biệt hóa. Mỗi quan hệ giữa tập thực thể và các nhóm con của nó được gọi là mối quan hệ *isa*. Một cách hình thức mỗi quan hệ này được mô tả như sau. Xét hai tập thực thể E_1 và E_2 . Ta nói E_1 *isa* E_2 , đọc là “ E_1 là một E_2 ”, nếu E_1 là một loại tập thực thể đặc biệt của E_2 hoặc E_2 là sự tổng quát hóa của E_1 . Trường hợp ngược lại của sự đặc biệt hóa được gọi là sự khái quát hóa.

Như vậy, mục đích chính của việc sử dụng thêm mối quan hệ *isa* này là tập thực thể E_1 kế thừa các thuộc tính của tập thực thể E_2 nhưng có thể có thêm các thuộc tính khác mà một số thực thể trong tập thực thể E_2 không có. Chẳng hạn, xét tập thực thể *TAIKHOAN* (tài khoản) có các thuộc tính như *số hiệu tài khoản* và *số dư*. Trên thực tế, một tài khoản còn được phân biệt theo một trong hai loại sau:

TAIKHOANTIETKIEM (tài khoản tiết kiệm) và
TAIKHOANKIEMTRA (tài khoản kiểm tra).

Các loại tài khoản này đều có hai thuộc tính *số hiệu tài khoản* và *số dư*, ngoài ra còn có thêm một số thuộc tính riêng cho nó, chẳng hạn các thực thể tài khoản tiết kiệm có thêm *thuộc tính tỷ suất* (tỷ số lãi suất) còn các thực thể tài khoản kiểm tra có thêm *thuộc tính số vượt trội* (số lượng rút vượt quá số lượng gửi). Khi đó, ta có các mối quan hệ:

TAIKHOANTIETKIEM isa TAIKHOAN và
TAIKHOANKIEMTRA isa TAIKHOAN.

2.1.4. Ràng buộc trên các mối quan hệ

Để có thể mô hình hóa thế giới thực một cách thích hợp, người ta đưa ra các ràng buộc trên mối quan hệ giữa các tập thực thể. Có hai loại ràng buộc như vậy: ràng buộc về tỷ số lực lượng và ràng buộc về sự tham gia. Các ràng buộc như thế này còn được gọi chung là *ràng buộc cấu trúc*.

2.1.4.1. Ràng buộc về tỷ số lực lượng

Tỷ số lực lượng của mỗi quan hệ ở đây là cho biết số lượng các thực thể từ mỗi tập thực thể tham gia vào trong mỗi quan hệ. Có tất cả ba loại mỗi quan hệ như thế: mỗi quan hệ một-một, mỗi quan hệ nhiều-một và mỗi quan hệ nhiều-nhiều. Một cách hình thức các mỗi quan hệ này được định nghĩa như sau. Cho hai tập thực thể E_1 và E_2 . Ta nói mỗi quan hệ R là:

- *Mỗi quan hệ một-một* (hay 1-1) giữa E_1 và E_2 nếu mỗi thực thể của E_1 kết hợp với nhiều nhất một thực thể của E_2 và ngược lại.
- *Mỗi quan hệ nhiều-một* (hay $n-1$) từ E_1 đến E_2 nếu mỗi thực thể trong E_2 kết hợp với nhiều thực thể trong E_1 và mỗi thực thể trong E_1 kết hợp với nhiều nhất một thực thể trong E_2 .
- *Mỗi quan hệ nhiều-nhiều* (hay $n-m$) giữa E_1 và E_2 nếu mỗi thực thể của E_1 kết hợp với nhiều thực thể của E_2 và ngược lại.

2.1.4.2. Ràng buộc về sự tham gia

Ràng buộc về sự tham gia trên một mỗi quan hệ R cho biết sự tồn tại của một thực thể có phụ thuộc vào mỗi quan hệ R giữa nó với một thực thể khác hay không. Sự tham gia của các thực thể trên các mỗi quan hệ ở đây rõ ràng được chia làm hai loại: *sự tham gia toàn phần* và *sự tham gia bộ phận*.

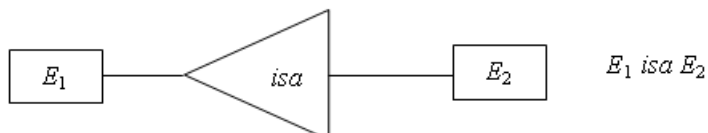
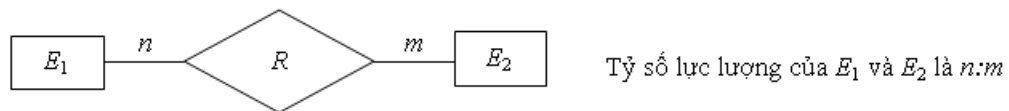
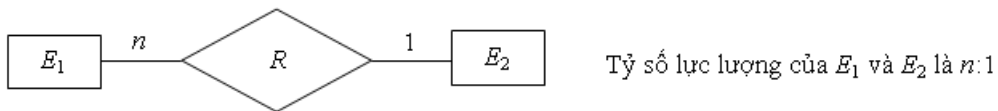
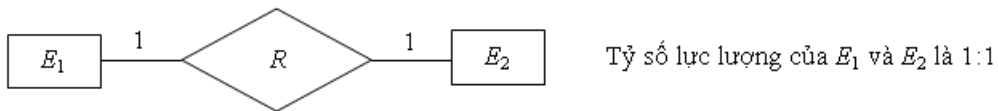
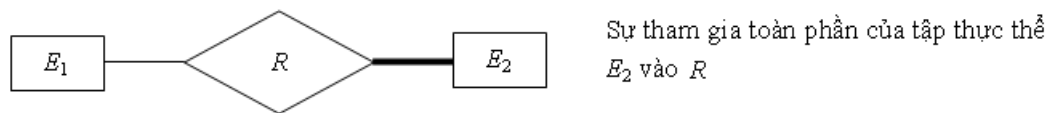
Chẳng hạn, xét mỗi quan hệ *QUANLY* giữa hai tập thực thể *NHANVIEN* và *PHONG*. Chúng ta biết rằng, không phải nhân viên nào cũng là người quản lý một phòng, điều này có nghĩa chỉ có một bộ phận nào đó của tập thực thể *NHANVIEN* tham gia vào mỗi quan hệ *QUANLY*. Lúc này, ta nói sự tham gia của tập thực thể *NHANVIEN* vào mỗi quan hệ *QUANLY* là sự tham gia bộ phận. Còn chẳng hạn, nếu quy định của công ty là mỗi nhân viên phải làm việc cho một phòng nào đó, thì khi đó mỗi thực thể nhân viên trong tập thực thể *NHANVIEN* chỉ có thể tồn tại nếu nó có tham gia vào một mỗi quan

hệ *LAMVIECCHO*. Lúc này, ta nói sự tham gia của tập thực thể *NHANVIEN* vào mối quan hệ *LAMVIECCHO* là sự tham gia toàn phần.

2.1.5. Sơ đồ thực thể-mối quan hệ

Sơ đồ thực thể-mối quan hệ hay còn gọi sơ đồ ER là sơ đồ tóm tắt những thông tin cần lưu trữ trong một CSDL đối với một thiết kế. Việc sử dụng các sơ đồ ER này là rất có ích. Trong sơ đồ thực thể-mối quan hệ người ta thường dùng các ký hiệu sau đây:

- Hình chữ nhật: biểu diễn các tập thực thể.



- Hình elip: biểu diễn các thuộc tính, chúng được nối với các tập

thực thể bằng các cạnh không có hướng, trong đó các thuộc tính của khóa tối thiểu sẽ được gạch chân, thuộc tính đa trị được biểu diễn bằng hai hình elip lồng nhau, thuộc tính suy diễn được biểu diễn bằng hình elip với nét không liền, thuộc tính phức hợp được biểu diễn bằng nhiều hình elip (tương ứng với các thành phần của thuộc tính phức hợp) nối với hình elip chính (tương ứng thuộc tính phức hợp).

- Hình thoi: biểu diễn các mối quan hệ, chúng được nối với các tập thực thể bằng các cạnh không có hướng. Cụ thể xét hai tập thực thể E_1 , E_2 và mối quan hệ R giữa E_1 và E_2 . Khi đó:

2.1.6. Thiết kế lược đồ thực thể-mối quan hệ

MHDL thực thể-mối quan hệ cho chúng ta sự linh hoạt quan trọng trong việc thiết kế lược đồ CSDL nhằm mô hình hóa một thế giới thực nhỏ. Để có được một thiết kế như vậy, chúng ta cần phải xem xét và quyết định những yếu tố sau:

- Dùng một thuộc tính hay một tập thực thể để biểu diễn một đối tượng.
- Dùng một tập thực thể hay một mối quan hệ để biểu diễn một khái niệm của thế giới thực.
- Dùng một mối quan hệ cấp ba hay một cặp mối quan hệ cấp hai.
- Có nên dùng đặc biệt hóa hay không (trong sơ đồ thực thể-mối quan hệ người ta không phân biệt đặc biệt hóa hay khái quát hóa).

Lưu ý, trong thực tiễn các mối quan hệ n - m xuất hiện khá nhiều, bởi vậy chúng ta cần phải thận trọng cách diễn đạt những mối quan hệ này. Nhiều MHDL không cho phép biểu diễn trực tiếp các mối quan hệ n - m , thay vào đó yêu cầu chúng phải được tách thành nhiều mối quan hệ n -1.

2.1.7. Một ví dụ về thiết kế lược đồ thực thể-mối quan hệ

Để minh họa cho việc thiết kế một MHDL thực thể-mối quan hệ, mục này chúng ta xem xét một ví dụ về xây dựng một mô hình CSDL của một công ty.

2.1.7.1. Mô tả bài toán

Giả sử sau bước tập hợp các yêu cầu và phân tích, người thiết kế CSDL mô tả lại công ty (thế giới nhỏ) đó bằng các câu như sau:

1. Công ty được tổ chức thành các phòng (đơn vị). Mỗi phòng có một tên duy nhất, một mã số duy nhất và một trưởng phòng. Việc một nhân viên được giao làm trưởng phòng được ghi lại ngày mà nhân viên đó được bổ nhiệm. Một phòng có thể có nhiều địa điểm.
2. Một phòng quản lý một số dự án. Mỗi dự án có một tên duy nhất, một mã số duy nhất và một địa điểm.
3. Với mỗi nhân viên của công ty cần lưu trữ những thông tin như sau: họ tên, mã số bảo hiểm xã hội, ngày sinh, địa chỉ và lương. Một nhân viên được phân công vào một phòng nào đó nhưng có thể làm việc với một số dự án mà không nhất thiết tất cả các dự án này đều do phòng của anh ta quản lý. Lưu trữ số giờ làm việc/tuần của mỗi nhân viên trên mỗi dự án và thông tin người giám sát trực tiếp của mỗi nhân viên.

2.1.7.2. Xác định các thực thể, thuộc tính và các mối quan hệ

Theo mô tả bài toán như trên, chúng ta có thể xác định ba tập thực thể:

1. Tập thực thể *PHONG* với các thuộc tính *Tên phòng*, *Mã phòng*, *Địa điểm*, *Trưởng phòng*, *Ngày bổ nhiệm*. Các thuộc tính *Tên phòng* và *Mã phòng* là hai khóa tối thiểu. *Địa điểm* là một thuộc tính đa trị.

2. Tập thực thể *DUAN* với các thuộc tính *Tên dự án*, *Mã dự án*, *Địa điểm*, *Phòng quản lý*. Trong đó, *Tên dự án* và *Mã dự án* đều là khóa tối thiểu.
3. Tập thực thể *NHANVIEN* với các thuộc tính *Họ tên*, *Mã BHXH* (mã số bảo hiểm xã hội), *Ngày sinh*, *Địa chỉ*, *Lương*, *Tên phòng*. *Họ tên* là thuộc tính phức hợp. Thuộc tính *Mã BHXH* là khóa tối thiểu.

Giữa các tập thực thể trên ta phát hiện được các mối quan hệ sau:

1. Mỗi phòng có một nhân viên lãnh đạo (trưởng phòng) nên giữa tập thực thể *NHANVIEN* và tập thực thể *PHONG* có một mối quan hệ, chúng ta đặt tên là *QUANLY*. Mỗi nhân viên chỉ có thể làm trưởng phòng của một phòng và một phòng không có nhiều hơn một trưởng phòng nên có một mối quan hệ 1-1. *PHONG* tham gia toàn phần vào mối quan hệ này (phòng nào cũng có trưởng phòng) còn *NHANVIEN* chỉ tham gia bộ phận (không phải nhân viên nào cũng làm trưởng một phòng nào đó). Mối quan hệ *QUANLY* có thuộc tính *Ngày bổ nhiệm* (ngày bổ nhiệm nhân viên nào đó làm trưởng phòng).
2. Mỗi nhân viên đều được phân công vào một phòng, một phòng có nhiều nhân viên nhưng không nhân viên nào được phân công vào hơn một phòng. Điều này có nghĩa có một mối quan hệ $n-1$ từ tập thực thể *NHANVIEN* đến tập thực thể *PHONG*. Cả hai tập thực thể tham gia trong mối quan hệ này đều tham gia toàn phần bởi phòng nào cũng có nhân viên và không có nhân viên nào lại không được phân công vào một phòng cụ thể. Chúng ta đặt tên mối quan hệ này là *LAMVIECCHO*.
3. Giữa tập thực thể *PHONG* và tập thực thể *DUAN* có một mối quan hệ, có thể gọi là mối quan hệ này là *DIEUHANH* (do phòng điều hành các dự án). Đây là mối quan hệ $n-1$ từ *DUAN* đến *PHONG*, vì một phòng có thể điều hành nhiều dự án và mỗi dự án chỉ chịu sự điều hành của một phòng. Tập thực thể *DUAN*

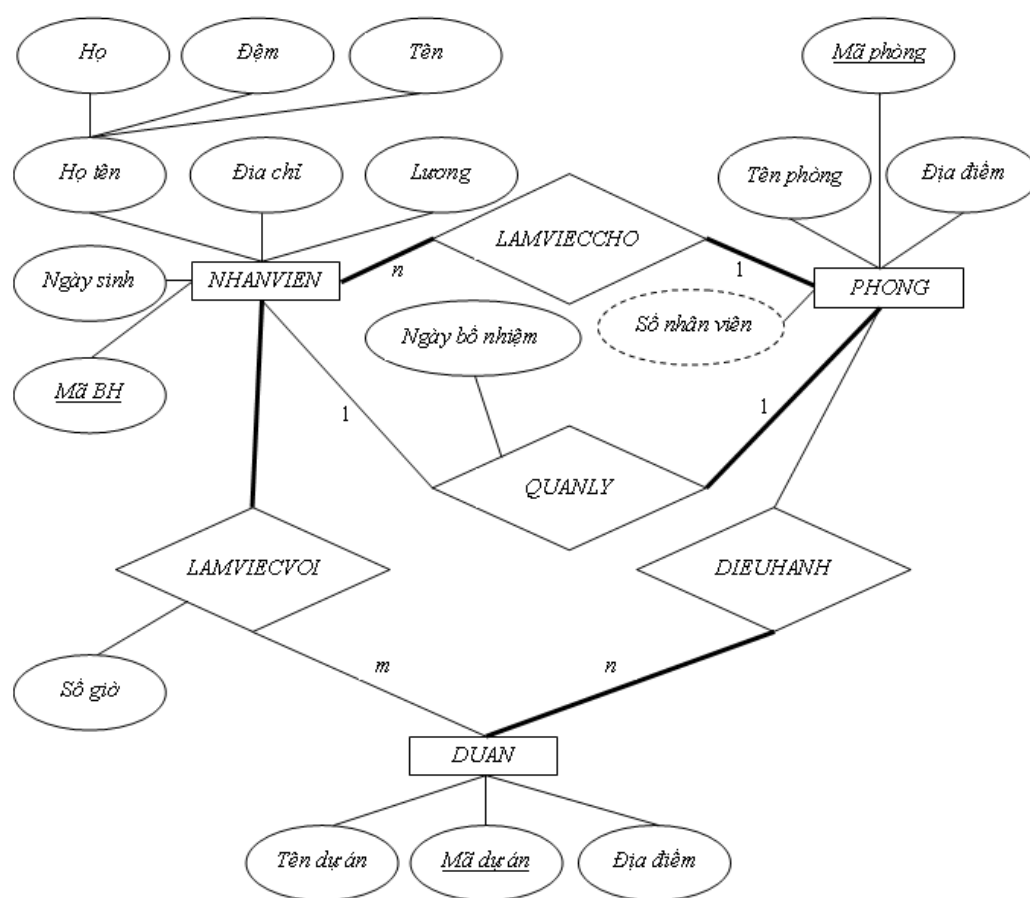
tham gia toàn phần vào mỗi quan hệ *DIEUHANH* bởi dự án nào nào cũng phải chịu sự điều hành của một phòng nào đó. Trong khi đó, tập thực thể *PHONG* tham gia bộ phận hay toàn phần tùy thuộc vào có hay không một phòng có chức năng điều hành dự án.

4. Một số nhân viên có tham gia vào các dự án, như vậy có một mối quan hệ trên hai tập thực thể *NHANVIEN*, *DUAN*. Do một nhân viên có thể tham gia nhiều dự án và mỗi dự án có sự tham gia của một số nhân viên nên mỗi quan hệ này là mối quan hệ $n-m$. Chúng ta đặt tên mối quan hệ này là *LAMVIECVOI*. Nếu có nhân viên không tham gia một dự án nào và có dự án nhưng không có nhân viên nào tham gia thì sự tham gia của hai tập thực thể thành phần *NHANVIEN* và *DUAN* vào mối quan hệ *LAMVIECVOI* đều là bộ phận. Mối quan hệ này có thuộc tính *Số giờ* cho biết số giờ công/tuần tương ứng với cặp thông tin (nhân viên, dự án).

2.1.7.3. Sơ đồ thực thể-mối quan hệ

Sau các phân tích như trên, trong mỗi tập thực thể chúng ta cần loại bỏ các thuộc tính được xem như các mối quan hệ. Chúng ta sẽ loại bỏ thuộc tính *Trưởng phòng*, *Ngày bổ nhiệm* khỏi tập thực thể *PHONG*; loại bỏ thuộc tính *Phòng quản lý* khỏi tập thực thể *DUAN*; loại bỏ thuộc tính *Tên phòng* khỏi tập thực thể *NHANVIEN*.

Từ đây, chúng ta có sơ đồ thực thể-mối quan hệ như hình vẽ sau:



Hình 2.2: Sơ đồ thực thể-mối quan hệ

2.2. Mô hình dữ liệu quan hệ

MHDL quan hệ do E. F. Codd đề xuất vào năm 1970 tại công ty IBM ở Sanjoe, Mỹ [3]. Đây là MHDL sử dụng các quan hệ toán học để mô tả dữ liệu và các phép toán đại số quan hệ để thao tác dữ liệu. Chính sự chặt chẽ của lý thuyết toán học về quan hệ nên dữ liệu trong MHDL này được mô tả một cách rõ ràng, logic và uyển chuyển. Cho tới nay đã có hàng trăm hệ quản trị CSDL theo MHDL quan hệ dùng cho các máy tính lớn và máy tính nhỏ. Chúng ta có thể hình dung MHDL quan hệ là một tập các quan hệ, với quan hệ được xem như là một bảng giá trị

gồm các hàng và các cột. Mỗi hàng trong bảng là một tập các giá trị có liên quan với nhau, các giá trị này biểu thị một sự kiện tương ứng với một thực thể hay một mối quan hệ trong thế giới thực. Ngoài ra, mỗi bảng và cột đều có tên với mục đích giúp cho việc hiểu ý nghĩa của mỗi hàng trong bảng.

Ví dụ 2.1. Sau đây là quan hệ *SINHVIEN* cho dưới dạng một bảng các giá trị:

<i>Masv</i>	<i>Hoten</i>	<i>Gioitinh</i>	<i>Ngaysinh</i>	<i>Quequan</i>	<i>Khoa</i>
K401	Lê Văn Sơn	Nam	10/02/1995	Huế	Toán
K402	Trần Thị Bình	Nữ	28/10/1995	Huế	Lý
K403	Trần Tiến	Nam	01/10/1996	Đà Nẵng	Toán
K4004	Lê Văn Hải	Nam	04/04/1997	Quảng Trị	Hóa
K405	Đặng Tuấn	Nam	25/08/1996	Hà Tĩnh	Văn

Bảng có tên *SINHVIEN* vì mỗi hàng trong bảng cho thông tin về một sinh viên cụ thể. Tên của mỗi cột trong bảng giúp chúng ta hiểu ý nghĩa giá trị trong mỗi ô được tạo nên bởi các hàng và cột này. Chẳng hạn, cột thứ nhất trong bảng được đặt tên *Masv* (Mã sinh viên), điều này giúp chúng ta hiểu giá trị “K401” chính là mã số sinh viên của sinh viên được mô tả trong hàng đầu tiên của bảng. Thông thường, tên của mỗi cột thể hiện một thuộc tính cần nói tới của các thực thể khi mà thông tin về các thực thể này được thể hiện trong bảng. Mọi giá trị trong một cột thường có cùng một kiểu dữ liệu.

Trong lý thuyết về MHDL quan hệ, người ta dùng thuật ngữ *quan hệ* để chỉ một bảng, *thuộc tính* để chỉ tên của một cột, *miền* để chỉ kiểu dữ liệu của một cột và *bộ* để chỉ một hàng trong bảng. Định nghĩa hình thức của các thuật ngữ này sẽ được trình bày ngay sau.

2.2.1. Quan hệ

Cho một tập hữu hạn khác rỗng các thuộc tính $U = \{a_1, a_2, \dots, a_n\}$. Mỗi thuộc tính a_i ($i = 1, 2, \dots, n$) có một miền giá trị tương ứng $dom(a_i)$.

Mỗi tập con của tích Descartes $dom(a_1) \times dom(a_2) \times \dots \times dom(a_n)$ được gọi là một *quan hệ* trên U và ký hiệu $R(U)$. Trong trường hợp tập thuộc tính luôn mặc định là U thì để cho gọn chúng ta sẽ ký hiệu lại $R(U)$ bằng R .

Như vậy, quan hệ $R(U)$ là một tập các n -bộ có dạng như sau:

$$R(U) = \{(x_1, x_2, \dots, x_n) : x_i \in dom(a_i), i = 1, 2, \dots, n\}.$$

Một quan hệ không chứa bộ nào được gọi là *quan hệ rỗng*. Hai quan hệ có cùng tập thuộc tính và các thuộc tính cùng tên có cùng miền giá trị được gọi là *tương thích* với nhau. Ký hiệu $Rel(U)$ là tập tất các quan hệ $R(U)$. Với mỗi bộ $t \in R(U)$ và mỗi tập con thuộc tính $X \subseteq U$, ta ký hiệu $t(X)$ là bộ t chỉ chứa các giá trị của các thuộc tính trong X và ký hiệu $t(a_i)$ để chỉ giá trị của bộ t tương ứng với thuộc tính a_i .

Định nghĩa trên cho thấy, quan hệ R có hình ảnh một bảng trong đó mỗi cột ứng với một thuộc tính và mỗi hàng ứng với một bộ. Điều này dẫn đến mọi thông tin trong CSDL quan hệ đều có thể biểu diễn bởi các giá trị trong bảng.

Trong lý thuyết CSDL, chúng ta luôn giả sử tập thuộc tính U là hữu hạn và khác rỗng. Một cặp $S = (U, F)$, trong đó U là một tập hữu hạn các thuộc tính và F là tập các điều kiện giữa các thuộc tính, được gọi là một *lược đồ quan hệ*. Một ràng buộc trên tập thuộc tính U là một tính chất (hay một tân từ) trên tập tất cả các quan hệ xác định trên U . Khi đó tập điều kiện F còn được gọi là *tập các ràng buộc toàn vẹn*.

Một lược đồ quan hệ (LĐQH) dùng để mô tả về cấu trúc và các ràng buộc toàn vẹn của một quan hệ. Một quan hệ có thể thay đổi theo thời gian do việc thực hiện các phép toán cập nhật trên các bộ của quan hệ như thêm, xóa và sửa bộ. Còn LĐQH dùng để chỉ những nội dung của quan hệ tồn tại lâu hơn, đó là tính bất biến của quan hệ theo một nghĩa tương đối. Theo nghĩa này, các quan hệ thỏa LĐQH S , tức có cấu trúc và ràng buộc toàn vẹn được mô tả bởi S , được gọi là một *thể hiện* của LĐQH S .

Ví dụ 2.2. Xét quan hệ *TKB* (thời khóa biểu) của một trường học:

<i>Ngày</i>	<i>Tiết</i>	<i>Mon</i>	<i>Phong</i>	<i>Giaovien</i>
1/9	1	Logic mờ	E201	Nguyễn Văn Hùng
1/9	2	Logic mờ	E201	Nguyễn Văn Hùng
1/9	1	Anh văn 1	H101	Lê Thị Hòa
1/9	2	Anh văn 1	H102	Trần Văn Hiệp
1/9	3	Anh văn 2	E201	Hoàng Thị Ngọc

Quan hệ này có tập thuộc tính $U = \{Ngày, Tiết, Môn, Phong, Giaovien\}$. Bộ thứ nhất của quan hệ là ("1/9", 1, "Logic mờ", "E201", "Nguyễn Văn Hùng"). Nếu tập thuộc tính $X = \{Tiết, Môn, Phong\}$ và t là bộ thứ nhất, thì bộ $t(X) = (1, "Logic mờ", "E201")$. Dữ liệu trong bảng trên sẽ có thể thay đổi theo thời gian nhưng cấu trúc của bảng được dùng trong một khoảng thời gian nào đó để thể hiện thời khóa biểu. Đồng thời, dữ liệu trong bảng cũng luôn phải thỏa mãn các ràng buộc để đảm bảo tính đúng đắn của thời khóa biểu, chẳng hạn:

- Một bộ (n, t, m, p, g) có ý nghĩa là vào tiết thứ t của ngày n giáo viên g dạy môn m tại phòng p .
- Miền của thuộc tính *Tiết* là tập hợp các số tự nhiên $\{1, 2, \dots, 10\}$ (cơ sở đào tạo ở đây chia một ngày học làm 10 tiết).
- Vào một tiết của một ngày, một giáo viên cụ thể chỉ có thể xuất hiện tại nhiều nhất một phòng học.

Một tập hữu hạn các LĐQH $\{S_1, S_2, \dots, S_k\}$ được gọi là một *lược đồ CSDL quan hệ* và ký hiệu là DS . Khi đó một *thể hiện* của DS (hay còn gọi một *CSDL quan hệ*) là một tập các thể hiện $DB = \{R_1, R_2, \dots, R_k\}$ sao cho mỗi R_i là một thể hiện của S_i và các thể hiện trong DB thỏa mãn các ràng buộc toàn vẹn trong DS .

2.2.2. Các tính chất cơ bản của quan hệ

Các tính chất cơ bản (về mặt toán học) sau của quan hệ $R \in Rel(U)$ là rõ ràng:

- Thứ tự các bộ trong R là không quan trọng và các bộ là phân biệt.
- Thứ tự các thành phần trong mỗi bộ của R là quan trọng.

Ngoài ra, trong ngữ cảnh của MHDL, các quan hệ (bảng) ở đây cần có thêm một số đặc trưng sau:

- Mỗi quan hệ cần có một tên phân biệt với tên của các quan hệ khác.
- Mỗi thuộc tính trong quan hệ có một tên phân biệt.
- Thứ tự các thuộc tính trong quan hệ là không quan trọng.
- Mỗi giá trị trong quan hệ phải là *giá trị nguyên tố*, đó là giá trị không phân chia được thành các thành phần trong phạm vi của MHDL quan hệ. Sau này chúng ta xem đây là điều kiện để quan hệ đạt dạng chuẩn một. Nói một cách khác, trong MHDL quan hệ không cho phép có các thuộc tính phức hợp hay thuộc tính đa trị.
- Các giá trị của một thuộc tính trong quan hệ thuộc cùng một miền.
- Ở thời điểm hiện hành (đang xét), giá trị của một số bộ tại một số thuộc tính có thể chưa biết, và ký hiệu bằng *null* cho trường hợp này.
- Ở mức logic (phù hợp với nhu cầu tự nhiên), thứ tự các thuộc tính và thứ tự các giá trị trong mỗi bộ không nên thực sự quan trọng. Để đảm bảo điều này cần có một sự tương ứng giữa các thuộc tính với các giá trị. Một trong những cách là định nghĩa quan hệ bằng tập hữu hạn các ánh xạ từ tập tất cả các thuộc tính vào hợp các miền giá trị của các thuộc tính đó.
- Ở mức vật lý, khi cài đặt quan hệ như một tệp thì các bản ghi (bộ) được lưu trữ một cách vật lý trên đĩa, lúc này chúng ta cần phải có một thứ tự giữa các bản ghi.

2.2.3. Khóa của quan hệ

Cho quan hệ $R \in Rel(U)$. Một *khóa* của R là một tập con thuộc tính $K \subseteq U$ sao cho với hai bộ bất kỳ $t_1, t_2 \in R$, thì tồn tại một thuộc

tính $a \in K$ để $t_1(a) \neq t_2(a)$. Trường hợp nếu với mọi tập con thật sự $K' \subset K$, K' không phải là khóa của R thì K được gọi là *khóa tối tiểu*¹ của R .

Như vậy, khóa tối tiểu của quan hệ dùng để nhận biết các bộ trong quan hệ. Điều này có ý nghĩa, khi cần truy tìm một bộ t nào đó trong quan hệ thì chúng ta chỉ cần biết giá trị của các thành phần khóa tối tiểu của t là đủ để tìm kiếm và nhận biết được hoàn toàn bộ t .

Dễ thấy U là một khóa của quan hệ R , và một quan hệ có thể có nhiều hơn một khóa tối tiểu. Người ta còn gọi mỗi khóa tối tiểu là *khóa dự tuyển*. Việc xác định khóa dự tuyển đòi hỏi chúng ta cần phải hiểu ngữ nghĩa của các thuộc tính mà ta định chọn để tạo thành khóa dự tuyển. Thông thường, có một khóa dự tuyển được chọn làm *khóa chính*. Việc lựa chọn một khóa dự tuyển làm khóa chính là tùy ý, tuy nhiên thường thì người ta hay chọn khóa chính là khóa dự tuyển có ít thuộc tính nhất, chẳng hạn tập chỉ có duy nhất một thuộc tính. Khi quan hệ biểu diễn các thực thể tồn tại khách quan (như sinh viên, môn học, mặt hàng, ...), các khóa dự tuyển thường có hơn một thuộc tính, khi đó người thiết kế CSDL có thể gán thêm cho những thực thể này một thuộc tính gọi là mã số để làm khóa chính (như mã số sinh viên, mã môn học, mã mặt hàng, ...). Khóa chính chỉ thật sự có ý nghĩa trong quá trình khai thác CSDL, còn trên phương diện lý thuyết khóa chính không có vai trò gì khác so với các khóa dự tuyển khác. Một số hệ quản trị CSDL như Access, Oracle, Informix, ... có cài đặt cơ chế tự động kiểm tra tính nhận diện khóa chính. Với cơ chế này, giá trị của các bộ ở các thuộc tính trong khóa chính phải khác giá trị *null*. Khi đó, CSDL thỏa mãn yêu cầu này được gọi là thỏa *ràng buộc toàn vẹn thực thể*.

Ký hiệu $Key(R)$ là tập tất cả các khóa tối tiểu của quan hệ R . Bài toán tìm $Key(R)$ là bài toán rất khó, bài toán này có độ phức tạp hàm mũ theo số thuộc tính. Tuy nhiên, tìm một khóa tối tiểu của quan hệ thì đơn giản hơn nhiều. Độ phức tạp của bài toán này chỉ là đa thức

¹Trong một số tài liệu thuật ngữ khóa tối tiểu được dùng theo nghĩa *khóa* và thuật ngữ khóa được dùng theo nghĩa *siêu khóa*.

theo kích thước của R .

Chẳng hạn, tập thuộc tính $\{Tiet, Monhoc\}$ là một khóa tối thiểu của quan hệ TKB ở Ví dụ 2.2 trên. Để rõ chúng ta xét thêm một ví dụ minh họa nữa.

Ví dụ 2.3. Xét quan hệ $R = \{t_1, t_2, t_3\}$ trên tập thuộc tính $U = \{a, b, c\}$:

$$R = \begin{array}{c|ccc} & a & b & c \\ \hline t_1 & 0 & 1 & 1 \\ t_2 & 1 & 2 & 0 \\ t_3 & 2 & 1 & 1 \end{array}$$

Dễ kiểm chứng được $\{a\}$ là khóa tối thiểu duy nhất của quan hệ R .

Lưu ý, sau này để gọn chúng ta sẽ không viết ra cột t_1, t_2, t_3 trong quan hệ. Một *khóa ngoài* của quan hệ $R \in Rel(U)$ là một tập con thuộc tính $K \subseteq U$ sao cho K là khóa tối thiểu của một quan hệ $R' \in Rel(U')$. Khóa ngoài đóng vai trò quan trọng khi thực hiện các thao tác dữ liệu. Một CSDL có nhiều LĐQH thường có nhiều khóa ngoài để biểu diễn các mối liên kết giữa các quan hệ, các mối liên kết này được gọi là *ràng buộc toàn vẹn tham chiếu*. Một CSDL không chỉ phải thỏa mãn các ràng buộc toàn vẹn mà còn có thể phải thỏa mãn một số ràng buộc khác nữa. Chẳng hạn, như “lương của cán bộ giảng dạy hợp đồng không thể cao hơn lương của cán bộ giảng dạy biên chế”. Những điều kiện kiểu như vậy thường được gọi chung là *ràng buộc toàn vẹn ngữ nghĩa*.

Tất cả các ràng buộc toàn vẹn cần được đặc tả trên lược đồ CSDL quan hệ nếu ta muốn bảo đảm các ràng buộc này trên mọi thể hiện của CSDL. Do vậy, ngôn ngữ định nghĩa dữ liệu trong một số hệ quản trị CSDL quan hệ cho phép đặc tả một số loại ràng buộc khác nhau để có thể kiểm soát duy trì các ràng buộc này một cách tự động.

2.2.4. Thuật toán tìm khóa của quan hệ

Mục này giới thiệu các thuật toán tìm một khóa tối thiểu và tìm tất cả các khóa tối thiểu của quan hệ. Các kết quả và thuật toán trong mục này

được đề xuất trong công trình [16]. Chúng ta bắt đầu với một số khái niệm cơ sở như sau. Xét một tập hợp hữu hạn khác rỗng bất kỳ V và $\mathcal{S}$ là một họ các tập con của V . Ký hiệu:

$$\begin{aligned}\min(\mathcal{S}) &= \{A \in \mathcal{S} : (\forall B \in \mathcal{S}, B \subseteq A) \Rightarrow (B = A)\}, \\ \max(\mathcal{S}) &= \{A \in \mathcal{S} : (\forall B \in \mathcal{S}, B \supseteq A) \Rightarrow (B = A)\}.\end{aligned}$$

Họ $\mathcal{S}$ thỏa điều kiện như sau:

$$\forall A, B \in \mathcal{S} : A \subseteq B \Rightarrow A = B$$

được gọi là một *hệ Sperner* (hay *họ Sperner*) trên V . Kết quả sau là cơ bản trong lý thuyết Sperner [8].

Định lý 2.1. *Số phần tử tối đa của một hệ Sperner $\mathcal{S}$ trên tập V là*

$$\binom{|V|}{\lfloor \frac{|V|}{2} \rfloor}$$

Ở đây ký hiệu $\lfloor x \rfloor$ số nguyên lớn nhất không vượt quá x . Dễ thấy các họ $\min(\mathcal{S})$, $\max(\mathcal{S})$ và $Key(R)$ là các hệ Sperner.

Hệ quả 2.1.

$$|Key(R)| \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

Bây giờ ta xét quan hệ $R = \{t_1, t_2, \dots, t_m\} \in Rel(U)$. Khi đó họ

$$E(R) = \{E_{ij} : 1 \leq i < j \leq m\},$$

với $E_{ij} = \{a \in U : t_i(a) = t_j(a)\}$, được gọi là *hệ bằng nhau* của quan hệ R . Họ $\max(E(R))$ còn được gọi là *hệ bằng nhau cực đại* của quan hệ R và ký hiệu lại là $\max(R)$. Trên cơ sở hệ bằng nhau cực đại, thuật toán tìm một khóa tối thiểu của quan hệ được xây dựng như sau:

Thuật toán 2.1 (Thuật toán tìm một khóa tối tiểu quan hệ)

Vào: Quan hệ $R = \{t_1, t_2, \dots, t_m\} \in Rel(U)$ với $U = \{a_1, a_2, \dots, a_n\}$.

Ra: Một khóa tối tiểu K của R .

Phương pháp:

Bước 1. Từ R , xây dựng hệ bằng nhau $E(R)$.

Bước 2. Từ $E(R)$, tính hệ bằng nhau cực đại $\max(R)$.

Bước $i + 2$. Xây dựng dãy tập thuộc tính $K_0, K_1, \dots, K_n$ theo qui tắc sau:

$$K_0 := U$$

$$K_i := \begin{cases} K_{i-1} \setminus \{a_i\} & \text{nếu } \forall E_{ij} \in \max(R) : K_{i-1} \setminus \{a_i\} \not\subseteq E_{ij} \\ K_{i-1} & \text{ngược lại} \end{cases}$$

với $i = 1, 2, \dots, n$.

Bước $n + 3$. Đặt $K := K_n$.

Có thể kiểm chứng được Thuật toán 2.1 cho ra đúng một khóa tối tiểu và độ phức tạp thời gian của thuật toán này là đa thức theo m và n . Ngoài ra, dễ thấy nếu thay đổi thứ tự các thuộc tính trong U thì chúng ta có thể tìm được một khóa tối tiểu khác. Hơn nữa, trong Bước $i + 2$ không nhất thiết đặt $K_0 = U$, chúng ta có thể đặt bằng một khóa nào đó quan hệ R .

Ví dụ 2.4. Xét lại quan hệ R trong Ví dụ 2.3.

Ta có $E_{12} = \emptyset, E_{13} = \{b, c\}, E_{23} = \emptyset$. Suy ra $E(R) = \{\emptyset, \{b, c\}\}$ và do đó $\max(R) = \{\{b, c\}\}$. Khi đó dãy tập các thuộc tính $K_0, \dots, K_3$ được xây dựng như sau:

$$K_0 = \{a, b, c\}$$

$$K_1 = K_0$$

$$K_2 = \{a, c\}$$

$$K_3 = \{a\}.$$

Vậy, tập thuộc tính $\{a\}$ là một khóa tối tiểu của quan hệ R .

Bây giờ chúng ta giới thiệu thuật toán tìm tất cả khóa tối tiểu của

quan hệ [16]. Để bắt đầu chúng ta cần các khái niệm sau. Một tập con bất kỳ $T \subseteq V$ được gọi là một *đại diện* của hệ Sperner $\mathcal{S}$ nếu T thỏa các điều kiện sau:

- i) Với mọi $A \in \mathcal{S}$ ta có $T \cap A \neq \emptyset$.
- ii) Không tồn tại một tập con $T' \subset T$ sao cho T' thỏa điều kiện (i).

Ký hiệu $Tran(\mathcal{S})$ là tập tất cả các đại diện của hệ Sperner $\mathcal{S}$, và *phần bù* của $Tran(\mathcal{S})$ là:

$$\overline{Tran(\mathcal{S})} = \{\bar{T} : T \in Tran(\mathcal{S})\}.$$

Chúng ta có kết quả sau [16]:

$$\overline{Tran(\mathcal{S})} = AntiSperner(\mathcal{S})$$

với

$$AntiSperner(\mathcal{S}) = \{A \subset V : (T \in \mathcal{S} \Rightarrow T \not\subseteq A) \text{ và } (A \subset B \Rightarrow (\exists T \in \mathcal{S})(T \subseteq B))\}.$$

Họ $AntiSperner(\mathcal{S})$ còn được gọi là *phản hệ Sperner* của $\mathcal{S}$. Như vậy, $AntiSperner(\mathcal{S})$ gồm tất cả các tập cực đại sao cho không bao hàm bất kỳ tập nào của hệ Sperner $\mathcal{S}$. Lưu ý, tương quan giữa $\mathcal{S}$ và $AntiSperner(\mathcal{S})$ là chúng xác định duy nhất lẫn nhau.

Tìm một đại diện của hệ Sperner $\mathcal{S} = \{E_1, E_2, \dots, E_m\}$ trên $V = \{a_1, a_2, \dots, a_n\}$ là thủ tục khá đơn giản, có thể thực hiện bằng quy nạp như sau:

$$\begin{aligned} T_0 &= V, \\ T(i) &= \begin{cases} T(i-1) \setminus \{a_i\} & \text{nếu } \forall E_i \in \mathcal{S} : (T(i-1) \setminus \{a_i\}) \cap E_i \neq \emptyset \\ T(i-1) & \text{ngược lại} \end{cases} \end{aligned}$$

với mọi $i = 1, 2, \dots, n$. Dễ dàng chứng minh được ngay tập $T(n)$ thu được như trên là một đại diện của hệ Sperner $\mathcal{S}$. Độ phức tạp thời gian của thủ tục này là $O(n^2m)$. Tuy nhiên, tìm tất cả các đại diện của một hệ Sperner thì phức tạp hơn, độ phức tạp thời gian là hàm mũ theo n .

Thuật toán 2.2 (Thuật toán tìm tất cả các đại diện)

Vào: Hệ Sperner $\mathcal{S} = \{E_1, E_2, \dots, E_m\}$ trên V .

Ra: Tập tất cả các đại diện $Tran(\mathcal{S})$.

Phương pháp:

Bước 1. Tính $\mathcal{L}_1 := \{\{a\} : a \in E_1\}$. Rõ ràng $\mathcal{L}_1 = Tran(\{E_1\})$.

Bước $q + 1$ ($q = 1, 2, \dots, m - 1$). Giả sử rằng

$$\mathcal{L}_q = \mathcal{S}_q \cup \{B_1, B_2, \dots, B_{t_q}\}$$

với $B_i \cap E_{q+1} = \emptyset$ ($i = 1, 2, \dots, t_q$) và $\mathcal{S}_q = \{A \in \mathcal{L}_q : A \cap E_{q+1} \neq \emptyset\}$.

Với mỗi i ($i = 1, 2, \dots, t_q$), xây dựng họ $\{B_i \cup \{b\} : b \in E_{q+1}\}$ và ký hiệu các phần tử của họ là $A_1^i, A_2^i, \dots, A_{r_i}^i$ ($i = 1, 2, \dots, t_q$). Đặt

$$\mathcal{L}_{q+1} := \mathcal{S}_q \cup \{A_p^i : \forall A \in \mathcal{S}_q \Rightarrow A \not\subseteq A_p^i, 1 \leq i \leq t_q, 1 \leq p \leq r_i\}.$$

Bước $m + 1$. Đặt $Tran(\mathcal{S}) := \mathcal{L}_m$.

Có thể kiểm chứng được $\mathcal{L}_m = Tran(\mathcal{S})$. Để ý, quá trình xác định $Tran(\mathcal{S})$ bằng Thuật toán 2.2 không phụ thuộc vào thứ tự của $E_1, E_2, \dots, E_m$. Bây giờ chúng ta sẽ phân tích độ phức tạp thời gian của Thuật toán 2.2. Giả sử số phần tử của $\mathcal{L}_q$ là l_q . Khi đó, độ phức tạp thời gian xấu nhất của Thuật toán 2.2 là

$$O(n^2 \sum_{q=1}^{m-1} t_q u_q)$$

ở đây

$$u_q = \begin{cases} l_q - t_q & \text{nếu } l_q > t_q \\ 1 & \text{nếu } l_q = t_q. \end{cases}$$

Dễ thấy các họ $\mathcal{L}_q$ tính được trong mỗi bước của thuật toán là một

hệ Sperner, và do đó số phần tử tối đa của $\mathcal{L}_q$ là

$$\binom{n}{\lfloor \frac{n}{2} \rfloor}$$

Hơn nữa

$$\binom{n}{\lfloor \frac{n}{2} \rfloor} \cong \frac{2^{n+\frac{1}{2}}}{\sqrt{\pi n}}.$$

Suy ra độ phức tạp thời gian của Thuật toán 2.2 không thể lớn hơn hàm mũ theo n . Tuy nhiên, trong trường hợp $l_q \leq l_m$ ($q = 1, 2, \dots, m-1$) chúng ta có thể thấy ngay độ phức tạp thời của Thuật toán 2.2 không lớn hơn $O(n^2 m |Tran(\mathcal{S})|)$. Do đó, trong trường hợp này độ phức tạp thời gian của Thuật toán 2.2 là đa thức theo n, m và $|Tran(\mathcal{S})|$. Khi đó, nếu số phần tử của $\mathcal{S}$ là nhỏ thì Thuật toán 2.2 rất hiệu quả, nó chỉ yêu cầu thời gian đa thức theo n . Hơn nữa, ta có kết quả sau [16].

Định lý 2.2. Cho quan hệ $R \in Rel(U)$. Khi đó

$$Key(R) = Tran(\min(\overline{E(R)})).$$

Do đó, trên cơ sở Định lý 2.2 và Thuật toán 2.2, thuật toán tìm tất cả khóa tối thiểu của một quan hệ được xây dựng như sau:

Thuật toán 2.3 (Thuật toán tìm tất cả khóa tối thiểu của quan hệ)

Vào: Quan hệ $R = \{t_1, t_2, \dots, t_m\} \in Rel(U)$.

Ra: Tập tất cả khóa tối thiểu $Key(R)$.

Phương pháp:

Bước 1. Từ R tính hệ bằng nhau $E(R)$.

Bước 2. Tính phần bù của $E(R)$:

$$\overline{E(R)} = \{\overline{E_{ij}} : E_{ij} \in E(R)\}$$

và $\min(\overline{E(R)})$.

Bước 3. Tính tập các đại diện $Tran(\min(\overline{E(R)}))$.

Bước 4. Đặt $Key(R) := Tran(\min(\overline{E(R)}))$. $Key(R)$ là tập tất cả các khóa tối thiểu của R .

Độ phức tạp thời gian của Thuật toán 2.3 chính là độ phức tạp thời gian của Thuật toán 2.2. Do đó, trong nhiều trường hợp Thuật toán 2.3 rất hiệu quả.

Ví dụ 2.5. Xét quan hệ R trên tập thuộc tính $U = \{a, b, c\}$ như sau:

$$R = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 0 \\ \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$$

Ta có $E_{12} = \{a, b\}$, $E_{13} = \{c\}$, $E_{14} = \emptyset$, $E_{23} = \emptyset$, $E_{24} = \{c\}$, $E_{34} = \{a, b\}$. Suy ra $E(R) = \{\{a, b\}, \{c\}, \emptyset\}$ và do đó $\overline{E(R)} = \{\{c\}, \{a, b\}, U\}$. Suy ra $\min(\overline{E(R)}) = \{\{c\}, \{a, b\}\}$. Khi đó, tập các đại diện của họ $\min(\overline{E(R)})$ được tính như sau:

$$\mathcal{L}_1 = \{\{c\}\}, \mathcal{L}_2 = \{\{a, c\}, \{b, c\}\}.$$

$$\text{Suy ra } Tran(\min(\overline{E(R)})) = \{\{a, c\}, \{b, c\}\}.$$

$$\text{Như vậy } Key(R) = \{\{a, c\}, \{b, c\}\}.$$

2.2.5. Cập nhật dữ liệu trên quan hệ

Các thao tác cơ bản hay còn gọi các phép toán cơ bản trên các quan hệ được chia làm hai loại: loại phép toán cập nhật và loại phép toán tìm kiếm thông tin. Các phép toán cập nhật nhằm tạo ra một quan hệ mới đúng đắn, trong khi đó các phép toán tìm kiếm thông tin nhằm đặc tả các thao tác lấy thông tin để trả ra theo yêu cầu. Mục này chỉ đề cập đến các phép toán cập nhật dữ liệu, đó là các phép toán làm CSDL thay đổi và vì vậy một số ràng buộc trên lược đồ CSDL có thể bị vi phạm mỗi khi thực hiện một phép cập nhật.

Có ba phép toán cập nhật dữ liệu cơ bản là thêm, xóa và sửa đổi. Phép toán thêm bộ được dùng để thêm một hay nhiều bộ mới vào một quan hệ, phép toán xóa bộ được dùng để loại bỏ các bộ trong một quan hệ và phép toán sửa đổi bộ được dùng để thay đổi các giá trị tại một số

thuộc tính của một số bộ. Mỗi khi thực hiện các phép toán này một số ràng buộc trên lược đồ CSDL có thể bị vi phạm. Sự vi phạm này được giải quyết cụ thể theo từng phép toán như sau:

- Với phép toán thêm bộ: nếu phép toán này vi phạm một hoặc nhiều ràng buộc thì thông thường phép toán này sẽ bị loại bỏ, và các hệ quản trị CSDL sẽ có thể giải thích cho NSD biết lý do. Cũng có thể hệ quản trị CSDL cố gắng kiểm tra lại nguyên nhân vi phạm và yêu cầu NSD khắc phục vi phạm này và chấp nhận phép toán thêm bộ nếu NSD làm đúng yêu cầu.

- Với phép toán xóa bộ: sự vi phạm ở đây chỉ là vi phạm ràng buộc tham chiếu đó là bộ được chỉ định để xóa được tham chiếu bởi một khóa ngoài từ các bộ khác trong CSDL. Có tất cả ba phương pháp xử lý cho trường hợp này. Phương pháp thứ nhất loại bỏ phép toán này, phương pháp thứ hai là *xóa lan truyền* tức là xóa các bộ tham chiếu đến bộ được chỉ định xóa ban đầu, và phương pháp cuối cùng là thay đổi các giá trị của các thuộc tính tham chiếu gây nên sự vi phạm bằng cách mỗi giá trị như vậy hoặc đặt lại bằng giá trị *null* hoặc đặt lại bằng một giá trị khác cho tham chiếu đến một bộ hợp lệ khác. Lưu ý, nếu một thuộc tính tham chiếu gây ra vi phạm là một thành phần của khóa chính thì không thể đặt lại bằng giá trị *null*, vì như vậy sẽ vi phạm ràng buộc toàn vẹn thực thể.

- Với phép toán sửa đổi bộ: nếu sửa đổi giá trị của các thuộc tính không phải khóa chính và cũng không phải khóa ngoài thì thường không gây nên sự vi phạm các ràng buộc toàn vẹn, khi đó hệ quản trị CSDL chỉ cần kiểm tra để đảm bảo rằng các giá trị mới thuộc miền và kiểu dữ liệu đúng đắn. Còn nếu giá trị tại khóa chính bị biến đổi thì điều này tương đương với việc xóa một bộ và thêm một bộ khác vào quan hệ (đã xét ở trên). Cuối cùng, nếu giá trị tại khóa ngoài bị biến đổi (nhận giá trị khác *null*) thì hệ quản trị CSDL phải đảm bảo rằng giá trị mới đã tham chiếu đến một bộ có mặt trong quan hệ được tham chiếu.

2.2.6. Đại số quan hệ

Đại số quan hệ là tập hợp các phép toán đại số trên các quan hệ với ý nghĩa dùng để tạo ra các quan hệ mới, quan hệ mới này là thông tin trả ra theo yêu cầu của NSD trong việc xử lý thông tin. Do đó, đại số quan hệ được xem là “ngôn ngữ cơ sở” quan trọng của ngôn ngữ bậc cao được sử dụng để thao tác trên các quan hệ. Các phép toán cơ bản của đại số quan hệ thường được chia thành hai nhóm. Nhóm thứ nhất gồm các phép toán tập hợp thông thường như hợp, giao, hiệu, tích Descartes và chia, bắt nguồn từ cách nhìn mỗi quan hệ là tập hợp các bộ. Nhóm thứ hai gồm các phép toán đặc biệt trên quan hệ như chọn, chiếu và kết nối. Các phép toán đặc biệt này là một cách tiếp cận khác trong việc biểu diễn truy vấn của ngôn ngữ đại số quan hệ, chúng liên quan đến các thao tác cơ bản làm thay đổi quan hệ như thêm, xóa, sửa đổi các bộ giá trị của một quan hệ.

2.2.6.1. Phép hợp, giao và hiệu

Cho hai quan hệ tương thích $R_1, R_2 \in Rel(U)$. *Hợp* của R_1 và R_2 , ký hiệu $R_1 \cup R_2$, là một quan hệ trên U gồm các bộ của mỗi quan hệ thành phần R_1 và R_2 . *Giao* của R_1 và R_2 , ký hiệu $R_1 \cap R_2$, là một quan hệ trên U gồm các bộ xuất hiện đồng thời trong cả hai quan hệ thành phần R_1 và R_2 . *Hiệu* của R_1 và R_2 , ký hiệu $R_1 \setminus R_2$, là một quan hệ trên U gồm các bộ của quan hệ R_1 không có trong R_2 .

Như vậy

$$R_1 \cup R_2 = \{t : t \in R_1 \text{ hoặc } t \in R_2\}$$

$$R_1 \cap R_2 = \{t : t \in R_1, t \in R_2\}$$

$$R_1 \setminus R_2 = \{t : t \in R_1, t \notin R_2\}$$

Ví dụ 2.6. Xét hai quan hệ tương thích trên tập thuộc tính $U = \{a, b, c\}$ như sau:

$$R_1 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 2 \\ \hline 0 & 1 & 3 \\ \hline 0 & 2 & 3 \\ \hline \end{array}, R_2 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 2 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Ta có

$$R_1 \cup R_2 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 2 \\ \hline 0 & 1 & 3 \\ \hline 0 & 2 & 3 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

$$R_1 \cap R_2 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 2 \\ \hline \end{array}$$

$$R_1 \setminus R_2 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 3 \\ \hline 0 & 2 & 3 \\ \hline \end{array}$$

Dễ thấy phép hợp và giao của hai quan hệ có tính giao hoán, còn phép hiệu thì không có tính chất này. Hai phép toán hợp và giao cũng còn có tính kết hợp khi tác động lên nhiều quan hệ. Phép giao (tương ứng hợp) phân phối đối với phép hợp (tương ứng giao). Chứng minh các tính chất này được quy về thành bài tập ở cuối chương. Ngoài ra, để cho gọn nhẹ giáo trình, một số nhận xét về các tính chất khác ở trong các mục sau cũng được chuyển về làm bài tập ở cuối chương.

2.2.6.2. Tích Descartes và phép chia

Ngoài ba phép toán hợp, giao và hiệu xác định trên các quan hệ tương thích ở trên, chúng ta còn có thêm hai phép toán tích Descartes và chia cũng thuộc nhóm các phép toán tập hợp của đại số quan hệ. Để bắt đầu, chúng ta giới thiệu phép toán dán bộ sau đây. Xét hai quan hệ

$R_1 \in \text{Rel}(U_1)$ và $R_2 \in \text{Rel}(U_2)$. Phép *dán bộ* của hai bộ $t_1 \in R_1$ và $t_2 \in R_2$, ký hiệu $t_1 \star t_2$, là một bộ t trên $U_1 \cup U_2$ sao cho $t(U_1) = t_1$ và $t(U_2) = t_2$.

Rõ ràng nếu $U_1 \cap U_2 \neq \emptyset$ thì

$$t_1(U_1 \cap U_2) = t_2(U_1 \cap U_2) = t(U_1 \cap U_2).$$

Bây giờ, giả sử $U_1 \cap U_2 = \emptyset$. Khi đó, *tích Descartes* của R_1 và R_2 , ký hiệu $R_1 \times R_2$, là một quan hệ trên tập thuộc tính $U_1 \cup U_2$ gồm các bộ $t_1 \star t_2$ sao cho $t_1 \in R_1$ và $t_2 \in R_2$.

Một cách hình thức

$$R_1 \times R_2 = \{t_1 \star t_2 : t_1 \in R_1, t_2 \in R_2\}.$$

Như vậy, thực chất phép tích Descartes của hai quan hệ là phép dán bộ của hai quan hệ đó với số bộ của quan hệ $R_1 \times R_2$ là $|U_1| \cdot |U_2|$.

Ví dụ 2.7. Xét hai quan hệ $R_1 \in \text{Rel}(U_1 = \{a, b, c\})$ và $R_2 \in \text{Rel}(U_2 = \{d, e\})$ như sau:

$$R_1 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 2 \\ \hline 0 & 1 & 3 \\ \hline 0 & 2 & 3 \\ \hline \end{array}, R_2 = \begin{array}{|c|c|} \hline d & e \\ \hline 0 & 1 \\ \hline 1 & 1 \\ \hline \end{array}$$

Ta có

$$R_1 \times R_2 = \begin{array}{|c|c|c|c|c|} \hline a & b & c & d & e \\ \hline 0 & 1 & 2 & 0 & 1 \\ \hline 0 & 1 & 2 & 1 & 1 \\ \hline 0 & 1 & 3 & 0 & 1 \\ \hline 0 & 1 & 3 & 1 & 1 \\ \hline 0 & 2 & 3 & 0 & 1 \\ \hline 0 & 2 & 3 & 1 & 1 \\ \hline \end{array}$$

Bây giờ, giả sử tiếp $R_2 \neq \emptyset$ và $U_2 \subseteq U_1$. Khi đó *chia* R_1 cho R_2

(hay *thương* của R_1 và R_2), ký hiệu $R_1 : R_2$, là một quan hệ trên tập thuộc tính $U_1 \setminus U_2$ gồm các bộ $t(U_1 \setminus U_2)$ sao cho $t \in R_1$ và với mọi $t_2 \in R_2$ thì $t(U_1 \setminus U_2) \star t_2 \in R_1$.

Như vậy

$$R_1 : R_2 = \{t(U_1 \setminus U_2) : t \in R_1, (\forall t_2 \in R_2 \Rightarrow t(U_1 \setminus U_2) \star t_2 \in R_1)\}.$$

Ví dụ 2.8. Xét hai quan hệ $R_1 \in Rel(U_1 = \{a, b\})$ và $R_2 \in Rel(U_2 = \{b\})$ như sau:

$$R_1 = \begin{array}{|c|c|} \hline a & b \\ \hline 1 & 0 \\ \hline 1 & 1 \\ \hline 2 & 0 \\ \hline 2 & 2 \\ \hline 3 & 0 \\ \hline 3 & 1 \\ \hline \end{array}, R_2 = \begin{array}{|c|} \hline b \\ \hline 0 \\ \hline 1 \\ \hline \end{array}$$

Ta có

$$R_1 : R_2 = \begin{array}{|c|} \hline a \\ \hline 1 \\ \hline 3 \\ \hline \end{array}$$

2.2.6.3. Phép chọn, chiếu và kết nối

Một cách tiếp cận khác trong việc biểu diễn câu hỏi của ngôn ngữ đại số quan hệ là qua các phép toán chọn, chiếu và kết nối. Chúng liên quan đến các thao tác cơ bản làm thay đổi quan hệ như thêm, xóa và sửa đổi các bộ giá trị. Xét quan hệ $R \in Rel(U)$. Giả sử F là một biểu thức logic. Khi đó, *chọn* (hay *lọc*) của R theo điều kiện F , ký hiệu $\sigma_F(R)$, là một quan hệ trên U gồm các bộ của R thỏa F .

Một cách hình thức

$$\sigma_F(R) = \{t \in R : F(t) \text{ đúng}\}.$$

Như vậy, phép chọn được dùng để xây dựng một tập con các bộ của quan hệ đã cho thoả mãn một điều kiện xác định. Biểu thức logic F còn được gọi là *biểu thức chọn* hay *biểu thức lọc*. Biểu thức logic F là một tổ hợp Boole ($\wedge, \vee, \neg$) của các toán hạng, trong đó mỗi toán hạng là một phép so sánh đơn giản giữa hai biến (thuộc tính) hay so sánh giữa một biến và một hằng. Biểu thức logic F sẽ cho giá trị *đúng/sai* đối với mỗi bộ đã kiểm tra. Các phép so sánh trong F là $>, \geq, <, \leq, =, \neq$.

Ví dụ 2.9. Cho quan hệ R trên tập thuộc tính $U = \{a, b, c\}$ như sau:

$$R = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 1 \\ \hline 1 & 2 & 0 \\ \hline 2 & 1 & 1 \\ \hline \end{array}$$

Xét biểu thức chọn $F = (b = "1")$, thì

$$\sigma_{b="1"}(R) = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 1 \\ \hline 2 & 1 & 1 \\ \hline \end{array}$$

Tức là $\sigma_{b="1"}(R) = \{t \in R : t(b) = 1\}$.

Lưu ý, thứ tự của của phép chọn là không quan trọng. Ngoài ra, phép chọn phân phối trên các phép toán như $\cup, \cap$ và $\setminus$.

Bây giờ, xét tập thuộc tính con $X \subseteq U$. Khi đó *chiều* của R trên X , ký hiệu $\Pi_X(R)$, là một quan hệ trên X gồm tất cả các bộ $t(X)$ sao cho $t \in R$.

Một cách hình thức

$$\Pi_X(R) = \{t(X) : t \in R\}.$$

Như vậy, phép chiếu được dùng để xây dựng một quan hệ mới từ quan hệ đã cho bằng cách bỏ đi một số thuộc tính.

Xét quan hệ R ở Ví dụ 2.9 trên. Với $X = \{b, c\}$ ta có:

$$\Pi_X(R) = \begin{array}{|c|c|} \hline b & c \\ \hline 1 & 1 \\ \hline 2 & 0 \\ \hline \end{array}$$

Bây giờ, chúng ta xét hai quan hệ $R_1 \in \text{Rel}(U_1)$ và $R_2 \in \text{Rel}(U_2)$. *Phép kết nối (tự nhiên)* của R_1 và R_2 , ký hiệu $R_1 \bowtie R_2$, là một quan hệ trên tập thuộc tính $U_1 \cup U_2$ gồm các bộ $t_1 \star t_2$ sao cho $t_1 \in R_1$ và $t_2 \in R_2$.

Một cách hình thức

$$R_1 \bowtie R_2 = \{t_1 \star t_2 : t_1 \in R_1, t_2 \in R_2\}.$$

Trong định nghĩa trên, hai tập thuộc tính U_1 và U_2 không nhất thiết phân biệt nhau. Do đó rõ ràng, nếu $U_1 \cap U_2 = \emptyset$ thì $R_1 \bowtie R_2 = R_1 \times R_2$. Như vậy, có thể thấy phép toán kết nối của hai quan hệ là một sự mở rộng của phép toán tích Descartes. Có thể kiểm chứng được phép kết nối là có tính kết hợp.

Ví dụ 2.10. Xét hai quan hệ R_1 và R_2 trên tập thuộc tính tương ứng $U_1 = \{a, b, c\}$ và $U_2 = \{b, c, d\}$ như sau:

$$R_1 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 1 \\ \hline 1 & 2 & 0 \\ \hline 2 & 1 & 1 \\ \hline \end{array}, R_2 = \begin{array}{|c|c|c|} \hline b & c & d \\ \hline 2 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Ta có

$$R_1 \bowtie R_2 = \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 0 & 1 & 1 & 1 \\ \hline 2 & 1 & 1 & 1 \\ \hline \end{array}$$

Trường hợp nếu $U_1 \cap U_2 = \emptyset$ và $a \in U_1, b \in U_2$, thì *phép kết nối θ* của R_1 và R_2 theo điều kiện $a\theta b$, ký hiệu $R_1 \bowtie_{a\theta b} R_2$, là một quan hệ trên tập thuộc tính $U_1 \cup U_2$ gồm các bộ $t_1 \star t_2$ sao cho $t_1(a)$ có quan hệ θ với $t_2(b)$, với $t_1 \in R_1$ và $t_2 \in R_2$.

Một cách hình thức

$$R_1 \bowtie_{a\theta b} R_2 = \{t_1 \star t_2 : t_1 \in R_1, t_2 \in R_2, t_1(a)\theta t_2(b)\}.$$

Ở đây, θ là các phép toán so sánh số học như $=, \neq, <, \leq, >, \geq$. Trường hợp θ là phép toán " $=$ " thì phép kết nối θ trên còn được gọi là *kết nối bằng*.

Như vậy, có thể thấy phép toán kết nối θ của hai quan hệ là một trường hợp đặc biệt của phép toán tích Descartes. Đó là phép dán bộ của hai quan hệ sao cho chúng thoả mãn điều kiện θ trên chúng. Từ định nghĩa dễ dàng suy ra được:

$$R_1 \bowtie_{a\theta b} R_2 = \sigma_{t_1(a)\theta t_2(b)}(R_1 \times R_2).$$

Ví dụ 2.11. Xét hai quan hệ $R_1 \in Rel(U_1 = \{a, b, c\})$ và $R_2 \in Rel(U_2 = \{d, e\})$ như sau:

$$R_1 = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 1 & 2 \\ \hline 0 & 1 & 3 \\ \hline 0 & 2 & 3 \\ \hline \end{array}, R_2 = \begin{array}{|c|c|} \hline d & e \\ \hline 0 & 1 \\ \hline 1 & 1 \\ \hline \end{array}$$

Ta có

$$R_1 \bowtie_{b>d} R_2 = \begin{array}{|c|c|c|c|c|} \hline a & b & c & d & e \\ \hline 0 & 1 & 2 & 0 & 1 \\ \hline 0 & 1 & 3 & 0 & 1 \\ \hline 0 & 2 & 3 & 0 & 1 \\ \hline 0 & 2 & 3 & 1 & 1 \\ \hline \end{array}$$

Ý nghĩa của phép kết nối là dùng để kết hợp hai bộ có liên quan với nhau, thuộc hai quan hệ khác nhau thành một bộ mới. Phép toán này quan trọng ở chỗ cho phép chúng ta xử lý mối liên quan giữa các quan hệ trong một CSDL. Ngoài phép kết nối trên, chúng ta còn một số phép toán kết nối mở rộng khác như phép kết nối nửa, kết nối trái

và kết nối phải. Những phép kết nối mở rộng này cũng đã được cài đặt trong một số hệ quản trị CSDL như Access, Oracle, SQL Server, ...

2.2.7. Các phép toán quan hệ bổ sung

Có nhiều truy vấn CSDL không thể thực hiện được bằng các phép toán đại số quan hệ cơ bản. Hầu hết các ngôn ngữ thao tác dữ liệu của các hệ quản trị CSDL quan hệ được thương mại hóa đã có những phép toán quan hệ bổ sung để thực hiện các truy vấn như vậy.

2.2.7.1. Các hàm kết tập

Kiểu truy vấn đầu tiên không thể thực hiện được trong đại số quan hệ là kiểu: đặc tả được bằng các hàm toán học có tính kết tập trên một tập hợp các giá trị của CSDL. Chẳng hạn, chúng ta thường gặp yêu cầu “cho biết tổng lương của các nhân viên trong đơn vị” hay “cho biết trung bình cộng của lương nhân viên trong đơn vị”, ... Sau đây là các hàm toán học thường gặp, chúng được áp dụng cho các tập hợp giá trị số.

- Hàm tính tổng: *SUM*
- Hàm tính trung bình cộng: *AVERAGE*
- Hàm tính giá trị lớn nhất: *MAX*
- Hàm tính giá trị nhỏ nhất: *MIN*
- Hàm đếm các bộ giá trị: *COUNT*

2.2.7.2. Các phép gộp nhóm

Có một kiểu yêu cầu khác nữa mà chúng ta cũng hay thường gặp, tuy nhiên không thể thực hiện được trong đại số quan hệ. Đó là kiểu yêu cầu nhóm các bộ trong một quan hệ theo một giá trị của một số các thuộc tính của chúng và sau đó áp dụng các hàm kết tập trên từng nhóm một cách độc lập. Chẳng hạn, xét quan hệ *SINHVIEN* ở Ví

dụ 2.1 và thực hiện nhóm các bộ của quan hệ *SINHVIEN* theo thuộc tính *Quequan*. Khi đó, mỗi nhóm sẽ gồm các sinh viên có cùng quê. Sau đó, chúng ta có thể yêu cầu đưa ra thông tin như có bao nhiêu sinh viên cùng tên, ... Kết quả của phép gộp nhóm như vậy là một quan hệ (mới). Định nghĩa hình thức phép gộp nhóm như sau. Xét quan hệ $R \in Rel(U)$, $a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_l \in U$ và $f_1, f_2, \dots, f_l$ là các hàm kết tập. *Gộp nhóm* trên R theo các thuộc tính $a_1, a_2, \dots, a_k$ thỏa điều kiện $((f_1 b_1), (f_2 b_2), \dots, (f_l b_l))$, ký hiệu $[a_1, a_2, \dots, a_k] \mathcal{F}_{((f_1 b_1), (f_2 b_2), \dots, (f_l b_l))}(R)$, là một quan hệ xác định trên tập các thuộc tính V trong đó V gồm:

- $a_1, a_2, \dots, a_k$ và
- $b'_1, b'_2, \dots, b'_l$ với các b'_j là các thuộc tính mới sinh ra từ tác động của hàm f_j lên thuộc tính b_j ($j = 1, 2, \dots, l$).

Trường hợp nếu danh sách thuộc tính để gộp nhóm là rỗng ($k = 0$) thì các hàm kết tập f_j được áp dụng cho các giá trị của các thuộc tính của tất cả các bộ trong quan hệ. Khi đó, quan hệ kết quả chỉ gồm đúng một bộ.

Chẳng hạn, xét quan hệ (bảng) *SINHVIEN* ở Ví dụ 2.1 và thực hiện phép gộp nhóm

$$[Quequan] \mathcal{F}_{COUNT_Masv}(SINHVIEN)$$

thì quan hệ kết quả sẽ là

<i>Quequan</i>	<i>COUNT_Masv</i>
Huế	2
Đà Nẵng	1
Quảng Trị	1
Quảng Nam	1

Còn nếu thực hiện phép gộp nhóm

$$\mathcal{F}_{COUNT_Masv}(SINHVIEN)$$

thì quan hệ kết quả sẽ là

<i>COUNT_Masv</i>
5

2.2.8. Một số ví dụ về đại số quan hệ

Mục này sẽ minh họa việc sử dụng các biểu thức đại số quan hệ trong việc biểu diễn các câu hỏi. Chẳng hạn, xét một CSDL gồm các LDQH được cho bởi các thể hiện của chúng qua các bảng sau đây:

<i>NHANVIEN</i>					
<i>Manv</i>	<i>Hoten</i>	<i>Ngaysinh</i>	<i>Gioitinh</i>	<i>Madv</i>	<i>Luong</i>
1	Nguyễn Hoàng	10/01/1970	Nam	P1	10.000.000
2	Hoàng Văn Lợi	5/05/1980	Nam	P2	6.000.000
3	Phan Cẩm Ly	2/10/1980	Nữ	P3	5.000.000
4	Bùi Thị Lý	25/10/1981	Nữ	P3	5.500.000
5	Nguyễn Thị Tú	15/11/1979	Nữ	P4	8.000.000
6	Hoàng Kim Tú	12/01/1984	Nam	P2	7.000.000
7	Trần Quốc Sĩ	2/02/1975	Nam	P4	7.000.000
8	Đặng Tuấn	22/11/1980	Nam	P2	6.000.000

<i>PHONG</i>		
<i>Madv</i>	<i>Tenphong</i>	<i>Matp</i>
P1	Giám đốc	1
P2	Kỹ thuật	8
P3	Tổ chức hành chính	3
P4	Nghiên cứu và phát triển	7

<i>DIADIEMDV</i>	
<i>Madv</i>	<i>Diadiem</i>
P1	20 Trần Hưng Đạo
P2	10 Nguyễn Huệ
P3	18 Trần Hưng Đạo
P4	5 Lê Lợi

<i>DUAN</i>			
<i>Mada</i>	<i>Tenda</i>	<i>Diadiemda</i>	<i>Madv</i>
D1	Mạng A	Sở Tài chính	P2
D2	Phần mềm B	Ngân hàng Vietcombank	P1
D3	Phần mềm C	Trường PTTH Nguyễn Huệ	P4

<i>CHAMCONG</i>		
<i>Manv</i>	<i>Mada</i>	<i>Sogio</i>
2	D1	8
8	D1	6
1	D1	8
2	D2	6
6	D2	7
5	D2	10
1	D2	10
7	D3	8
3	D3	7
4	D3	10
1	D3	10

Một số chú thích về tên quan hệ: *DIADIEMDV* là địa điểm đơn vị, *DUAN* là dự án.

Một số chú thích thêm về tên thuộc tính: *Manv* là mã nhân viên, *Madv* là mã số đơn vị, *Matp* là mã trưởng phòng, *Mada* là mã số dự án, *Tenda* là tên dự án, *Diadiemda* là địa điểm dự án.

Một số câu hỏi và biểu thức đại số quan hệ tương ứng:

- Cho biết địa điểm của dự án có mã số D3:

$$\Pi_{\{Diadiemda\}}(\sigma_{Mada="D3"}(DUAN)).$$

- Cho biết họ tên và lương của những nhân viên làm việc ở phòng Kỹ thuật:

$$\Pi_{\{Hoten,Luong\}}(NHANVIEN \bowtie \sigma_{Tenphong="Kỹ thuật"}(PHONG)).$$

- Cho biết họ tên các nhân viên làm việc cho tất cả các dự án do phòng có mã số P2 quản lý:

$$KQTG := \Pi_{\{Manv,Madv\}}(CHAMCONG) \div$$

$$\Pi_{\{Mada\}}(\sigma_{Madv="P2"}(DUAN)),$$

$$\Pi_{\{Hoten\}}(KQTG \bowtie NHANVIEN).$$

- Với mỗi dự án được thực hiện tại Sở Tài chính, hãy cho biết mã số dự án đồng thời cho biết họ tên và ngày sinh của trưởng phòng quản lý dự án này:

$$KQTG := \Pi_{\{Mada, Madv\}}(\sigma_{Diadiemda="Sở Tài chính"}(DUAN)),$$

$$\Pi_{\{Mada, Hoten, Ngaysinh\}}(NHANVIEN \bowtie_{Manv=Matp} (PHONG \bowtie KQTG)).$$

2.2.9. Phép tính quan hệ

Phép tính quan hệ còn được gọi là ngôn ngữ tân từ. Tương tự như đại số quan hệ, các phép tính quan hệ được xem như là ngôn ngữ cơ sở quan trọng thao tác trên MHDL quan hệ. Đại số quan hệ cung cấp một tập các phép toán quan hệ với mục đích từ quan hệ đang có làm thế nào xây dựng ra quan hệ mà chúng ta muốn có, trong khi đó các phép tính quan hệ lại chỉ cung cấp những ký hiệu để phát biểu định nghĩa quan hệ ta muốn có theo các quan hệ đang có. Đây chính là điểm khác nhau quan trọng giữa đại số quan hệ với phép tính quan hệ. Tuy nhiên, cần nhấn mạnh rằng cả hai đều là ngôn ngữ hình thức của CSDL quan hệ. Cơ sở toán học của phép tính quan hệ là logic tân từ cấp một. Một số ngôn ngữ CSDL quan hệ đã thương mại hóa được xây dựng dựa trên phép tính quan hệ như QUEL, QBE hay SQL. Có tất cả hai cách áp dụng phép tính tân từ vào ngôn ngữ truy vấn CSDL quan hệ là ngôn ngữ tân từ biến bộ và ngôn ngữ tân từ biến miền.

So với đại số quan hệ, trong phép tính quan hệ tính “phi thủ tục” hay tính “khai báo” là cao hơn, thể hiện ở chỗ trong phép tính quan hệ chúng ta có thể viết một biểu thức khai báo những gì cần, trong khi đó đại số quan hệ phải viết một dãy các phép toán với thứ tự rõ ràng. Tuy nhiên, điều quan trọng nhất là khả năng biểu thị câu hỏi, thì cả hai là

tương đương. Do vậy, ở đây chúng ta không đi sâu vào chi tiết trình bày các phép tính quan hệ.

2.2.10. Khung nhìn

Trong kiến trúc ba mức (theo ANSI-SPARC) được trình bày ở Chương 1, chúng ta đã mô tả khung nhìn chính là cấu trúc của CSDL cho một NSD cụ thể. Trong MHDL quan hệ thuật ngữ khung nhìn có khác đi một chút.

2.2.10.1. Quan hệ cơ sở và khung nhìn

Một *quan hệ cơ sở* (là một quan hệ được đặt tên) tương ứng với một thực thể trong lược đồ logic và các bộ của nó được lưu trữ vật lý trong CSDL. Còn *khung nhìn* là kết quả thực hiện một hay nhiều thao tác (phép toán) trên các quan hệ cơ sở nhằm đưa ra một quan hệ mới. Có thể hiểu một khung nhìn là một quan hệ ảo, nghĩa là nó không thực sự tồn tại ở mức vật lý trong CSDL (như quan hệ cơ sở) mà là câu trả lời cho yêu cầu của NSD tại thời điểm nêu yêu cầu. Khung nhìn mang tính động theo nghĩa những thay đổi trong quan hệ cơ sở sẽ tác động lập tức đến thể hiện của khung nhìn. Như vậy, mức khung nhìn (mức ngoài) của một CSDL có thể bao gồm cả các quan hệ cơ sở và cả các khung nhìn được suy dẫn, kết xuất từ các quan hệ cơ sở.

2.2.10.2. Mục đích khung nhìn

Cơ chế sử dụng khung nhìn đem lại một số lợi ích sau đây.

- Cung cấp một cơ chế an ninh hữu hiệu và linh hoạt bởi sự che dấu một số phần của CSDL trước một số NSD nào đó. NSD không được thông báo về sự tồn tại một số thuộc tính hay một số bộ trong CSDL khi chúng không được xuất hiện trong khung nhìn.

- Cho phép NSD truy nhập được dữ liệu theo cách họ muốn sao cho tại cùng một thời điểm, cùng một dữ liệu có thể nhìn thấy bởi nhiều NSD khác nhau trong những khuôn dạng khác nhau.
- Cho phép đơn giản hóa những phép toán trên các quan hệ cơ sở. Chẳng hạn, nếu một khung nhìn là kết nối của hai quan hệ thì NSD có thể thực hiện một số phép toán một ngôi như chiếu, chọn trên khung nhìn.

2.2.10.3. Cập nhật qua khung nhìn

Rõ ràng, nếu quan hệ cơ sở được cập nhật thì sẽ tác động ngay đến tất cả các khung nhìn có tham chiếu đến quan hệ cơ sở này. Tuy nhiên, điều ngược lại thì cần phải có thêm một số ràng buộc chặt chẽ. Sau đây là các nguyên tắc mà hầu hết các hệ thống dùng để kiểm soát và quyết định cho phép một cập nhật CSDL xảy ra qua khung nhìn:

- Nếu khung nhìn được xác định bằng một truy vấn đơn giản chỉ liên quan đến một quan hệ cơ sở và chứa hoặc khóa chính hoặc khóa dự tuyển của quan hệ cơ sở đó thì được phép cập nhật thông qua khung nhìn này.
- Nếu khung nhìn liên quan đến nhiều quan hệ cơ sở thì không được phép cập nhật thông qua khung nhìn này.
- Nếu khung nhìn liên quan đến các hàm kết tập và các phép gộp nhóm thì không được phép cập nhật thông qua khung nhìn này.

2.2.11. Nhận xét về mô hình dữ liệu quan hệ

Có thể thấy MHDL quan hệ cho một cách biểu diễn dữ liệu đơn giản, các dữ liệu dù là thực thể hay liên kết đều được biểu diễn dưới một dạng thống nhất là quan hệ, tức là các bảng giá trị, khá dễ hiểu và quen thuộc đối với NSD không chuyên về tin học. Hơn thế nữa, MHDL quan hệ đã cung cấp các khái niệm chặt chẽ được hình thức hóa cao, cho phép áp dụng các công cụ toán học, các thuật toán tối ưu trên MHDL quan

hệ. VHDL quan hệ được trừu tượng hoá cao và chỉ dừng ở mức logic, nghĩa là độc lập với mức vật lý với sự cài đặt cũng như các thiết bị lưu trữ. Do đó, VHDL quan hệ đảm bảo được tính độc lập dữ liệu với các chương trình ứng dụng. Ngoài ra, do sự biểu diễn dữ liệu thống nhất, VHDL quan hệ có thể cho phép cung cấp các ngôn ngữ thao tác dữ liệu ở mức cao, dễ sử dụng và có thể chuẩn hoá. Tuy nhiên, VHDL quan hệ biểu diễn các liên kết chưa thật tự nhiên, còn nghèo về ngữ nghĩa.

BÀI TẬP CHƯƠNG 2

2.1. Cho tập thuộc tính $U = \{a, b, c, d, e\}$ và quan hệ $R \in Rel(U)$ như sau:

$$R = \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 2 & 0 & 1 & 1 \\ 2 & 2 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Tìm một khóa tối tiểu của quan hệ R .

2.2. Cho quan hệ $R \in Rel(U)$ và dãy bao hàm thức các tập con thuộc tính $X_1 \subseteq X_2 \subseteq \dots \subseteq X_k \subseteq U$. Chứng minh rằng

$$\Pi_{X_1}(\Pi_{X_2}(\dots \Pi_{X_k}(R) \dots)) = \Pi_{X_1}(R).$$

2.3. Chứng minh với mọi quan hệ $R \in Rel(U)$, $a, b \in U$ và $x \in dom(a)$, $y \in dom(b)$ ta luôn có

$$\sigma_{a="x"}(\sigma_{b="y"}(R)) = \sigma_{b="y"}(\sigma_{a="x"}(R)).$$

2.4. Chứng minh với mọi quan hệ $R_1, R_2 \in Rel(U)$, $a \in U$ và $x \in dom(a)$ thì

$$\sigma_{a="x"}(R_1 \circ R_2) = \sigma_{a="x"}(R_1) \circ \sigma_{a="x"}(R_2)$$

với $\circ$ là các phép toán $\cap, \cup$ hay $\setminus$.

2.5. Cho quan hệ $R \in Rel(U)$, $X \subseteq U$, $a \in X$ và $x \in dom(a)$. Chứng minh rằng

$$\Pi_X(\sigma_{a="x"}(R)) = \sigma_{a="x"}(\Pi_X(R)).$$

2.6. Cho hai quan hệ tương thích R_1 và R_2 trên tập thuộc tính $U = \{a, b, c, d, e\}$ như sau:

$$R_1 = \begin{array}{|c|c|c|c|c|} \hline a & b & c & d & e \\ \hline 0 & 1 & 2 & 0 & 1 \\ \hline 0 & 1 & 3 & 3 & 1 \\ \hline 0 & 2 & 3 & 0 & 0 \\ \hline 1 & 0 & 0 & 2 & 1 \\ \hline 0 & 2 & 1 & 1 & 0 \\ \hline \end{array}, R_2 = \begin{array}{|c|c|c|c|c|} \hline a & b & c & d & e \\ \hline 0 & 2 & 3 & 0 & 0 \\ \hline 1 & 2 & 1 & 0 & 0 \\ \hline 2 & 3 & 1 & 1 & 0 \\ \hline 0 & 2 & 1 & 1 & 0 \\ \hline \end{array}$$

Thực hiện các phép toán đại số quan hệ sau:

- $R_1 \cap R_2, R_1 \cup R_2, R_1 \setminus R_2, R_2 \setminus R_1.$
- $\Pi_X(R_1), \Pi_X(R_1 \cap R_2), \Pi_X(R_1 \cup R_2), \Pi_X(R_1 \setminus R_2)$ với $X = \{a, b, e\}.$
- $\sigma_{(a="0") \wedge (b="1")}(R_1), \sigma_{(a="0") \vee (d="1")}(R_2).$

2.7. Chứng minh trên các quan hệ tương thích chúng ta luôn có:

- Phép hợp và giao có tính giao hoán, kết hợp.
- Phép hợp (giao) phân phối đối với phép giao (hợp).

2.8. Cho các quan hệ tương thích $R_1, R_2 \in Rel(U)$. Chứng minh rằng $\forall X \subseteq U$ ta có:

- $\Pi_X(R_1 \cap R_2) = \Pi_X(R_1) \cap \Pi_X(R_2)$
- $\Pi_X(R_1 \cup R_2) = \Pi_X(R_1) \cup \Pi_X(R_2).$

2.9. Cho hai quan hệ $R_1 \in Rel(U_1), R_2 \in Rel(U_2)$ và $U_2 \subseteq U_1, U = U_1 \setminus U_2$. Chứng minh rằng $R_1 \div R_2 = \Pi_U(R_1) \setminus \Pi_U((\Pi_U(R_1) \bowtie R_2) \setminus R_1).$

2.10. Cho quan hệ $R_1 \in Rel(U), a \in U$ và $x \in dom(a)$. Xét quan hệ $R_2 \in Rel(\{a\})$ sao cho $t(a) = x$ với mọi $t \in R_2$. Chứng minh rằng $R_1 \bowtie R_2 = \sigma_{a="x"}(R_1).$

2.11. Chứng minh tính đúng đắn của Thuật toán 2.1.

2.12. Cho $\mathcal{S}$ là một hệ Sperner trên tập hữu hạn khác rỗng V . Chứng minh rằng

$$\overline{Tran(\mathcal{S})} = AntiSperner(\mathcal{S}).$$

2.13. Cho quan hệ $R \in Rel(U)$. Chứng minh rằng

$$Key(R) = Tran(\min(\overline{E(R)})).$$

Chương 3.

NGÔN NGỮ SQL

SQL hay còn gọi SEQUEL (Structured English Query Languages) là ngôn ngữ văn tin quan hệ được thiết kế và cài đặt vào năm 1970 tại phòng nghiên cứu của San Jose của hãng IBM. Đến năm 1976, ngôn ngữ SEQUEL được cải tiến thành SEQUEL2. Khoảng từ năm 1978-1979, SEQUEL2 được cải tiến và đổi tên thành SQL. Năm 1986, Viện Tiêu chuẩn Quốc gia Mỹ (ANSI) và tổ chức Tiêu chuẩn quốc tế (ISO) đã công bố một chuẩn SQL, được gọi là SQL-86. Sau đó có một chuẩn mở rộng cho SQL, gọi là SQL-89, công bố năm 1989. Các hệ CSDL hiện nay đều hỗ trợ ít nhất các đặc điểm của chuẩn này. Phiên bản hiện hành của SQL theo chuẩn ANSI/ISO là chuẩn SQL-92, trong khi việc chuẩn hóa cho SQL-3 đang được tiến hành. SQL là một ngôn ngữ phi thủ tục, chuẩn mực và điển hình. Để xây dựng được các ứng dụng phức tạp, SQL được cho phép nhúng vào các ngôn ngữ thế hệ thứ 3 như Cobol, PL1. Hiện nay, rất nhiều sản phẩm phần mềm thương mại đều có cài đặt SQL như Oracle, Ingres, DB2, ... So sánh với các ngôn ngữ thế hệ thứ 4, SQL chưa có chức năng đồ họa và phương tiện hỏi trợ giúp quyết định. Tuy nhiên, không chỉ có Query mà SQL còn thực hiện nhiều chức năng khác như bổ sung, sửa đổi, xóa dữ liệu (thông tin) và liên kết lưu trữ thông tin. Như vậy, khả năng của SQL bao gồm tạo dữ liệu, hỏi dữ liệu, cập nhật dữ liệu, sắp xếp dữ liệu và trình bày dữ liệu. Ngoài ra, SQL cũng có một số khả năng và tiện ích khác về CSDL quan hệ như tạo sinh báo cáo, chuyển đổi ngôn ngữ và chức năng trợ giúp. Chương này sẽ trình bày một cách chi tiết về ngôn ngữ SQL.

3.1. Một số quy ước ký pháp

Chuẩn ISO SQL không sử dụng các thuật ngữ hình thức như quan hệ, thuộc tính và bộ mà sử dụng các thuật ngữ tương ứng như bảng, cột và bản ghi.

Lưu ý rằng SQL không bám chặt vào định nghĩa quan hệ có tính hình thức. Chẳng hạn, SQL cho phép một bảng là kết quả của thao tác SELECT có thể chứa hai hàng giống nhau, và cho phép người dùng được sắp thứ tự các cột trong bảng. Chương này sẽ sử dụng một số quy ước về ký pháp sau đây để trình bày một cách ngắn gọn, dễ hiểu về cú pháp các câu lệnh SQL:

- Các từ khóa, các hàm được viết bằng chữ in hoa.
- Các tên do người dùng đặt được viết bằng chữ thường, trong đó tên bảng và tên ràng buộc toàn vẹn là chữ thường đậm, còn tên cột viết bằng chữ thường không đậm. Chẳng hạn:

SELECT Hoten, Luong FROM **Nhanvien**

- Các phạm trù cú pháp, tức là các thành phần ngôn ngữ mà người sử dụng phải điền cụ thể vào khi viết lệnh sẽ được viết trong cặp dấu ngoặc nhọn (< >).
- Các thành phần tùy chọn, tức là các thành phần có thể xuất hiện hay không xuất hiện trong câu lệnh, được viết trong cặp dấu ngoặc vuông ([]). Chẳng hạn:

INSERT INTO <tên bảng> [<tên cột 1>, <tên cột 2>, ...]

VALUES (<biểu thức 1>, <biểu thức 2>, ...)

- Việc lựa chọn một trong các khả năng được thể hiện bằng dấu gạch đứng (|).
- Thành phần bắt buộc phải chọn trong danh sách được viết bằng cặp dấu móc ({}).

Lưu ý, các lệnh SQL có thể được viết trên nhiều dòng và kết thúc bằng dấu chấm phẩy (;) nhưng mỗi từ khóa, tên hàm, tên thuộc tính, tên bảng phải nằm trọn vẹn trong một dòng, không được viết tách xuống dòng. Trên thực tế từ khóa, tên hàm, tên thuộc tính, tên bảng viết bằng chữ in hoa hay chữ in thường đều được coi là như nhau.

Để tiện minh họa cho các câu lệnh SQL, các ví dụ ở chương này vẫn sử dụng CSDL xét ở mục 2.2.8, và được trình bày lại dưới đây để tiện theo dõi:

Nhanvien

<i>Manv</i>	<i>Hoten</i>	<i>Ngaysinh</i>	<i>Gioitinh</i>	<i>Madv</i>	<i>Luong</i>
1	Nguyễn Hoàng	10/01/1970	Nam	P1	10.000.000
2	Hoàng Văn Lợi	5/05/1980	Nam	P2	6.000.000
3	Phan Cẩm Ly	2/10/1980	Nữ	P3	5.000.000
4	Bùi Thị Lý	25/10/1981	Nữ	P3	5.500.000
5	Nguyễn Thị Tú	15/11/1979	Nữ	P4	8.000.000
6	Hoàng Kim Tú	12/01/1984	Nam	P2	7.000.000
7	Trần Quốc Sĩ	2/02/1975	Nam	P4	7.000.000
8	Đặng Tuấn	22/11/1980	Nam	P2	6.000.000

Phong

<i>Madv</i>	<i>Tenphong</i>	<i>Matp</i>
P1	Giám đốc	1
P2	Kỹ thuật	8
P3	Tổ chức hành chính	3
P4	Nghiên cứu và phát triển	7

Diadiemdv

<i>Madv</i>	<i>Diadiem</i>
P1	20 Trần Hưng Đạo
P2	10 Nguyễn Huệ
P3	18 Trần Hưng Đạo
P4	5 Lê Lợi

Duan

<i>Mada</i>	<i>Tenda</i>	<i>Diadiemda</i>	<i>Madv</i>
D1	Mạng A	Sở Tài chính	P2
D2	Phần mềm B	Ngân hàng Vietcombank	P1
D3	Phần mềm C	Trường PTTH Nguyễn Huệ	P4

Chamcong

Manv	Mada	Sogio
2	D1	8
8	D1	6
1	D1	8
2	D2	6
6	D2	7
5	D2	10
1	D2	10
7	D3	8
3	D3	7
4	D3	10
1	D3	10

3.2. Định nghĩa dữ liệu

Mục này trình bày các câu lệnh định nghĩa dữ liệu (DDL). Các câu lệnh này được dùng để tạo ra cấu trúc của CSDL (các bảng) và xác lập các ràng buộc toàn vẹn trên đó.

3.2.1. Các kiểu miền trong SQL

SQL-92 chuẩn hỗ trợ nhiều kiểu miền được cài sẵn như sau:

- *CHAR*(n): là chuỗi ký tự có độ dài cố định, với n ($n \leq 255$) được xác định bởi NSD. Dạng tương đương đầy đủ là CHARACTER.
- *VARCHAR*(n): là chuỗi ký tự có độ dài thay đổi (từ 0 đến n), với n được xác định bởi NSD. Dạng tương đương đầy đủ là CHARACTER VARYING.
- *INT*: là số nguyên. Miền trị có kiểu này là một tập con hữu hạn các số nguyên và độ lớn của tập này tùy thuộc vào máy. Dạng tương đương đầy đủ là INTEGER.
- *SMALLINT*: là số nguyên nhỏ. Miền trị có kiểu này là một tập con của miền có kiểu *INT* và tùy thuộc vào máy.

- *NUMERIC*(p, d): là số dấu chấm thập phân cố định, với độ chính xác được xác định bởi NSD, bao gồm p chữ số và một dấu chấm thập phân, trong đó có d chữ số bên phải dấu chấm thập phân. Chẳng hạn, *NUMERIC*(4, 1) cho phép số 123.4 được lưu trữ đúng, các dạng khác chẳng hạn số 1234.5 hay số 12.56 không được lưu trữ đúng.
- *REAL*, *DOUBLE PRECISION*: tương ứng là số thập phân dấu phẩy động và số thập phân dấu phẩy động độ chính xác gấp đôi.
- *FLOAT*(n): là số thập phân dấu phẩy động với độ chính xác ít nhất n chữ số được xác định bởi NSD.
- *DATE*: là ngày-tháng-năm theo lịch (*yy-mm-dd*).
- *TIME*: là giờ trong ngày theo giờ, phút và giây.

Lưu ý, SQL cho phép các phép toán số học và các phép so sánh trên nhiều miền số. Hơn nữa, các giá trị thuộc các miền tương thích (chẳng hạn như INT và SMALLINT) cũng có thể so sánh được với nhau. Giá trị *null* là có thể thuộc mọi miền, tuy nhiên với một số thuộc tính việc sử dụng giá trị *null* sẽ không thích hợp. Do đó, SQL cho phép khai báo miền của một thuộc tính với đặc tả NOT NULL để cấm thuộc tính đó nhận giá trị *null*. Chẳng hạn, khóa chính của một quan hệ là một trường hợp đặc biệt không cho phép nhận giá trị *null*.

3.2.2. Định nghĩa bảng trong SQL

3.2.2.1. Tạo bảng

Cú pháp tổng quát của câu lệnh tạo bảng, cùng với khai báo một số ràng buộc toàn vẹn trên bảng như sau:

```
CREATE TABLE <tên bảng> (
<tên cột 1> <kiểu dữ liệu 1> (<kích thước 1>) [CONSTRAINT <tên
1>],
<tên cột 2> <kiểu dữ liệu 2> (<kích thước 2>) [CONSTRAINT <tên
2>],
```

...

<tên cột n > <kiểu dữ liệu n > (<kích thước n >) [CONSTRAINT <tên n >]

[, CONSTRAINT <tên 1>]

[, CONSTRAINT <tên 2>]

...

[, CONSTRAINT <tên n >]);

Trong đó cú pháp tổng quát của khai báo các ràng buộc toàn vẹn là:

[CONSTRAINT <tên ràng buộc toàn vẹn>] | NULL | NOT NULL

|

UNIQUE [(<tên cột i >, <tên cột j >, ...)] |

PRIMARY KEY [(<tên cột i >, <tên cột j >)]

[FOREIGN KEY [(<tên cột i >, <tên cột j >, ...)] |

REFERENCES <tên bảng> (<tên cột i >, <tên cột j >, ...) |

CHECK (<điều kiện>)

- Tác dụng: tạo các bảng theo cấu trúc được chỉ ra.

Lưu ý 3.1.

1) Tên các đối tượng như tên bảng, tên cột, tên khung nhìn, tên ràng buộc toàn vẹn phải thỏa quy định: gồm tối đa 32 ký tự chữ cái Latinh, chữ số Ả-rập và dấu gạch nối dưới (-), không chứa dấu cách hay chữ cái không thuộc bảng chữ cái Latinh như tiếng Việt chẳng hạn. Không phân biệt chữ in hoa hay chữ thường.

2) Tên bảng phải là duy nhất trong CSDL, không trùng với từ khóa nào trong ngôn ngữ quản trị CSDL.

3) Tên các cột trong một bảng phải khác nhau, thứ tự các tên cột không quan trọng. Các tên cột có thể giống nhau khi chúng thuộc các bảng khác nhau.

4) Dạng đơn giản nhất của câu lệnh tạo bảng có cú pháp là:

```
CREATE TABLE <tên bảng> (
    <tên cột 1> <kiểu dữ liệu 1> (<kích thước 1>),
    <tên cột 2 <kiểu dữ liệu 2> (<kích thước 2>),
    ...
    <tên cột n> <kiểu dữ liệu n> (<kích thước n>) );
```

Chẳng hạn, để tạo bảng **Nhanvien** ta có thể dùng câu lệnh sau:

```
CREATE TABLE Nhanvien(
    Manv          NUMBER(2),
    Hoten         VARCHAR(25),
    Ngaysinh      DATE,
    Gioitinh      VARCHAR(3),
    Madv          CHAR(2),
    Luong         NUMBER(9));
```

Bây giờ chúng ta sẽ tạo các bảng phức tạp hơn.

Ví dụ 3.1. Tạo bảng **Duan** với các cột như sau:

- Mada: khóa chính.
- Tenda: cấm giá trị *null* và có các giá trị phân biệt ở các bộ khác nhau.

- Diadiemda: có giá trị ngầm định là 'Sở Tài chính'.
- Madv.

```
CREATE TABLE Duan(
    Mada          CHAR(2) PRIMARY KEY,
    Tenda         VARCHAR(25) NOT NULL UNIQUE,
    Diadiemda     VARCHAR(40) DEFAULT 'Sở Tài chính',
    Madv          CHAR(2));
```

Ví dụ 3.2. Các câu lệnh dưới đây còn khai báo thêm ràng buộc toàn vẹn tham chiếu cho bảng **Nhanvien**:

```
CREATE TABLE Nhanvien(
```

Manv	NUMBER(2) PRIMARY KEY,
Hoten	VARCHAR(25) NOT NULL,
Ngaysinh	DATE,
Gioitinh	VARCHAR(3) DEFAULT 'Nam',
Madv	CHAR(2) REFERENCES Phong (Madv),
Luong	NUMBER(9));

Và câu lệnh để tạo ra bảng **Chamcong** với việc đặt tên cho ràng buộc toàn vẹn khóa chính là PK, có thể viết:

```
CREATE TABLE Chamcong(
    Manv          NUMBER(2) REFERENCES Nhanvien(Manv),
    Mada          CHAR(2) REFERENCES Duan(Mada),
    Sogio         NUMBER(9),
    CONSTRAINT PK PRIMARY KEY (Manv, Mada));
```

Lưu ý, đi sau từ khóa CHECK là một tân từ mà mọi bản ghi trong bảng đều phải thỏa mãn.

Ví dụ 3.3. Giá trị của cột Trinhdo của tất cả bản ghi trong bảng **Giangvien** sau phải là một trong 3 phần tử đã được liệt kê ‘Đại học’, ‘Thạc sĩ’ và ‘Tiến sĩ’:

```
CREATE TABLE Giangvien(
    Manv          NUMBER(2),
    Hoten         VARCHAR(25) NOT FULL,
    Gioitinh      VARCHAR(3) DEFAULT 'Nam',
    Madv          CHAR(2) REFERENCES Phong(Madv),
    Luong         NUMBER(9),
    Trinhdo       VARCHAR(15) NOT FULL,
    PRIMARY KEY (Manv),
    CHECK (Trinhdo IN('Đại học', 'Thạc sĩ', 'Tiến sĩ')));
```

3.2.2.2. Đặt bí danh

- Cú pháp: CREATE SYNONYM <bí danh> FOR <tên bảng>

- Tác dụng: đặt bí danh (để ngắn gọn) cho bảng.

Ví dụ 3.4. Tạo bí danh **nv** cho bảng **Nhanvien**, chúng ta dùng lệnh như sau:

```
CREATE SYNONYM nv FOR Nhanvien
```

3.2.2.3. Xóa dữ liệu

- Cú pháp: DROP TABLE <tên bảng>
- Tác dụng: xóa bảng.

Ví dụ 3.5. Để xóa bảng **Nhanvien** chúng ta dùng lệnh:

```
DROP TABLE Nhanvien
```

3.2.3. Tạo lập các chỉ mục

- Cú pháp: CREATE INDEX <tên bảng chỉ mục> ON <tên bảng> (<tên cột 1> [, <tên cột 2>, ...])
- Tác dụng: tạo ra một bảng lưu trữ vị trí các bản ghi dựa trên giá trị tăng dần của một hay một số cột nào đó.

Ví dụ 3.6. Chỉ mục cột **Hoten** trong bảng **Nhanvien** theo thứ tự giá trị tăng dần ta dùng lệnh:

```
CREATE INDEX nvht ON Nhanvien(Hoten)
```

Lưu ý, việc tạo chỉ mục không làm thay đổi thứ tự vật lý của các bản ghi trong bảng. Việc tạo chỉ mục làm tăng tốc độ tìm kiếm thông tin trong CSDL. Ngoài ra, chúng ta chỉ nên tạo chỉ mục cho các bảng có số lượng lớn bản ghi và ít được cập nhật, các cột được tạo chỉ mục là các cột chứa thông tin thường xuyên được tham chiếu tới.

Trong trường hợp này nếu muốn xóa một bảng chỉ mục ta dùng lệnh sau: DROP INDEX <tên bảng chỉ mục>

3.3. Thao tác dữ liệu

Mục này trình bày các câu lệnh thao tác dữ liệu (DML). Các câu lệnh này được sử dụng để cập nhật và truy vấn thông tin của CSDL.

3.3.1. Cập nhật cơ sở dữ liệu

SQL có tất cả ba câu lệnh có thể biến đổi CSDL là INSERT (thêm dữ liệu), DELETE (loại bỏ dữ liệu) và UPDATE (sửa đổi dữ liệu).

3.3.1.1. Thêm dữ liệu

- Cú pháp: INSERT INTO <tên bảng> [(danh sách cột)] VALUES (danh sách các giá trị)
- Tác dụng: thêm một bản ghi mới vào bảng được chỉ ra.

Lưu ý, tên bảng hoặc là tên của một bảng cơ sở hoặc là tên một khung nhìn cho phép cập nhật. Danh sách cột đưa ra tên của một hay nhiều cột. Các tên cột cách nhau bởi dấu phẩy. Nếu trong câu lệnh không xuất hiện danh sách cột, thì SQL ngầm định một danh sách tất cả các cột, và thứ tự như đã xuất hiện trong câu lệnh CREATE TABLE khi tạo ra bảng này. Nếu có cột nào đó không xuất hiện trong danh sách cột thì khi tạo ra bảng, cột này đã phải được khai báo cho phép chứa giá trị *null* hoặc đặt trước giá trị ngầm định. Ngoài ra, danh sách các giá trị phải tương thích với danh sách cột như sau: số các phần tử của hai danh sách phải bằng nhau, mỗi phần tử trong danh sách các giá trị phải thích hợp về kiểu dữ liệu đối với tên cột tương ứng trong danh sách cột.

Ví dụ 3.7. Câu lệnh sau thêm một bản ghi vào bảng **Nhanvien**:

```
INSERT INTO Nhanvien VALUES ('9', 'Nguyen Van A', '1/2/1980', 'Nam', 'P3', 2600000)
```

Để ý, câu lệnh INSERT còn có thêm một dạng nữa. Tuy nhiên, ở đây chúng ta không tìm hiểu chi tiết lệnh này.

3.3.1.2. Xóa dữ liệu

- Cú pháp: DELETE FROM <tên bảng> WHERE <điều kiện>
- Tác dụng: xóa các bản ghi thỏa điều kiện chỉ ra.

Lưu ý, điều kiện ở đây là mệnh đề logic. Trường hợp nếu câu lệnh DELETE không có WHERE thì tất cả các bản ghi trong bảng sẽ bị xóa.

Ví dụ 3.8. Câu lệnh sau xóa tất cả các bản ghi trong bảng **Nhanvien**:

```
DELETE FROM Nhanvien
```

Hay câu lệnh sau xóa nhân viên có mã số 2 trong bảng **Nhanvien**:

```
DELETE FROM Nhanvien
```

```
WHERE Manv = 2
```

3.3.1.3. Sửa đổi dữ liệu

- Cú pháp: UPDATE <tên bảng> SET <tên cột 1> = <giá trị 1> [, <tên cột 2> = <giá trị 2>, ...] [WHERE <điều kiện>]
- Tác dụng: thay giá trị của cột bởi giá trị của biểu thức tương ứng của các bản ghi thỏa điều kiện được chỉ ra.

Lưu ý, từ khóa SET dùng để khai báo một hay một số cột sẽ được sửa đổi. Trường hợp khi không có WHERE, thì mọi bản ghi trong bảng sẽ được sửa đổi trên những cột xác định bởi SET, ngược lại chỉ những bản ghi thỏa điều kiện đặt sau WHERE mới được sửa đổi.

Ví dụ 3.9. Câu lệnh sau tăng lương 10% cho mọi nhân viên trong bảng **Nhanvien**:

```
UPDATE Nhanvien
```

```
SET Luong = Luong * 1.1
```

Hay tăng lương 12% cho các nhân viên làm ở đơn vị có mã số 'P3' trong bảng **Nhanvien**:

```
UPDATE Nhanvien
```

SET Luong = Luong * 1.12

WHERE Madv = 'P3'

Ngoài ra, chúng ta có thể thay đổi cấu trúc bảng bởi lệnh sau:

- Cú pháp: ALTER TABLE <tên bảng> ADD <tên cột> <kiểu dữ liệu>
- Tác dụng: bổ sung một cột mới vào bảng.

Ví dụ 3.10. Thêm cột mới Diachi vào bảng **Nhanvien** ta dùng lệnh:

ALTER TABLE **Nhanvien** ADD Diachi VARCHAR(45)

Trường hợp nếu muốn xóa một cột của một bảng chúng ta sử dụng lệnh:

- Cú pháp: ALTER TABLE <tên bảng> DROP <tên cột>
- Tác dụng: xóa một cột của bảng.

Ví dụ 3.11. Xóa cột Diachi ra khỏi bảng **Nhanvien** ta dùng lệnh:

ALTER TABLE **Nhanvien** DROP Diachi

3.3.2. Truy vấn cơ sở dữ liệu

3.3.2.1. Cấu trúc cơ sở để truy vấn

Cấu trúc cơ sở của một biểu thức truy vấn SQL gồm ba câu SELECT, FROM và WHERE trong đó:

SELECT: ứng với phép chiếu của đại số quan hệ, được dùng để liệt kê các cột muốn có trong kết quả của câu hỏi.

FROM: ứng với phép tích Descartes của đại số quan hệ, được dùng để liệt kê các bảng được sử dụng khi tìm kết quả cho câu hỏi.

WHERE: ứng với phép chọn của đại số quan hệ, được dùng để đưa ra điều kiện chọn. Điều kiện chọn là một tân từ chứa các cột của các bảng xuất hiện trong câu FROM.

Cú pháp điển hình của một biểu thức truy vấn trong SQL như sau:

```

SELECT [DISTINCT | ALL]{[<biểu thức cột> AS [<tên mới>]]
[, ...]}
FROM <tên bảng> [<bí danh>] [, ...]
[WHERE <điều kiện>]
[GROUP BY <danh sách tên cột>]
[HAVING <điều kiện>]
[ORDER BY <danh sách tên cột>]

```

với

<biểu thức cột>: là tên của một cột hoặc một biểu thức,

<tên bảng>: là tên của một bảng trong CSDL hay một khung nhìn mà ta có quyền truy nhập vào,

<bí danh>: là tên viết tắt của tên bảng,

GROUP BY: dùng để gộp nhóm các bản ghi cùng giá trị tương ứng ở các cột xuất hiện trong <danh sách tên cột>,

HAVING: dùng để lọc các nhóm thỏa <điều kiện>,

ORDER BY: quy định thứ tự (tăng dần-ASC hay giảm dần-DESC, mặc định là tăng dần) giá trị trên một số cột nào đó trong kết quả trả ra. Lưu ý, việc sắp thứ tự có thể được thực hiện trên nhiều cột.

- Tác dụng: trả về bảng thỏa yêu cầu đặt ra.

Lưu ý 3.2.

1) Thứ tự các câu trong dạng biểu thức truy vấn ở trên không thể thay đổi.

2) Kết quả của một truy vấn SQL là một bảng, có thể chứa các bản ghi (hàng) trùng lặp. Do đó, để loại bỏ tình trạng này (chỉ giữ lại một bản ghi, nếu có nhiều bản ghi trùng lặp), chúng ta cần có thêm từ khóa DISTINCT sau SELECT. Lúc này, thời gian xử lý truy vấn sẽ nhiều hơn.

3) SELECT có thể chứa các biểu thức số học với các toán tử +, -, *, / và thao tác trên các hằng hay các giá trị cột của các bản ghi.

4) SELECT *: chọn tất cả các cột của bảng.

5) Truy vấn có thêm câu WHERE được gọi là *truy vấn có điều kiện*, ngược lại gọi là *truy vấn không có điều kiện*.

6) Điều kiện trong WHERE còn tổng quát hơn điều kiện chọn trong đại số quan hệ, chẳng hạn chúng ta có thể dùng các phép toán số học để so sánh như: WHERE $a > b + c - 10$.

Sau đây là một số ví dụ minh họa.

Ví dụ 3.12. Tìm tên các dự án và địa điểm của các dự án, ta dùng lệnh:

```
SELECT Tenda, Diadiemda
```

```
FROM Duan
```

(Biểu thức đại số quan hệ tương đương là $\Pi_{\{Tenda, Diadiemda\}}(Duan)$)

Lúc đó kết quả của truy vấn đối với bảng **Duan** là:

Tenda	Diadiemda
Mạng A	Sở Tài chính
Phần mềm B	Ngân hàng Vietcombank
Phần mềm C	Trường PTTH Nguyễn Huệ

Ví dụ 3.13. Cho biết họ tên của mỗi nhân viên và lương mới của mỗi người nếu lương họ được tăng 15%. Trong bảng kết quả, chúng ta đặt tên cho cột thứ 3, chẳng hạn tên là Luongmoi (lương mới):

```
SELECT Hoten, Luong + Luong * 0.15 AS Luongmoi
```

```
FROM Nhanvien
```

Lưu ý 3.3. Trong một câu truy vấn thể hiện một *điều kiện so sánh* có thể xuất hiện các toán tử logic như AND, OR, NOT cùng với các dấu ngoặc chỉ ra thứ tự định giá. Các luật để định giá một biểu thức điều kiện là:

- Một biểu thức được định giá từ trái sang phải.
- Các biểu thức con trong ngoặc được định giá trước.
- Các phép toán NOT được định giá trước các phép toán AND và OR.

- Các phép toán AND định giá trước các phép toán OR.

Ví dụ 3.14. Liệt kê danh sách các nhân viên có mã đơn vị là 'P2' và lương trên 6,5 triệu đồng, với mỗi nhân viên như vậy các thông tin đưa ra gồm: mã nhân viên, họ tên, mã đơn vị và lương:

```
SELECT Manv, Hoten, Madv, Luong
FROM Nhanvien
WHERE Madv = 'P2' AND Luong > 6.500.000
```

Phép toán hay được dùng là phép đối sánh mẫu với toán tử LIKE và hai ký tự đặc biệt như sau:

- Ký tự phần trăm (%) sánh hợp với mọi xâu.
- Ký tự nối dưới (–) sánh hợp với mọi ký tự.

Trong đối sánh xâu các chữ viết hoa hay viết thường là không sánh hợp. Các minh họa sau sẽ làm rõ hơn khái niệm sánh hợp:

- 'PTTH%': sánh hợp với mọi xâu bắt đầu bằng 'PTTH'.
- '%PTTH%': sánh hợp với mọi xâu chứa xâu con 'PTTH', chẳng hạn 'PTTH', 'Truong PTTH', 'PTTH Nguyen Hue'.
- '___': sánh hợp với mọi xâu có đúng ba ký tự.
- '___%': sánh hợp với mọi xâu có ít nhất ba ký tự.

SQL còn cho phép dùng toán tử NOT LIKE để tìm theo các xâu không sánh hợp.

Ví dụ 3.15. Tìm tên và mã các dự án mà địa điểm dự án chứa từ 'PTTH' bằng lệnh như sau:

```
SELECT Mada, Tenda, Diadiemda
FROM Duan
WHERE Diadiemda LIKE '%PTTH%'
```

SQL cho phép dùng các giá trị *null* để chỉ sự thiếu thông tin về giá trị của một bản ghi tại một thuộc tính. Đó có thể là các giá trị tồn tại trên thực tế nhưng chưa được biết hoặc các giá trị đó không tồn tại

hay không áp dụng được. SQL sử dụng từ khóa IS NULL để kiểm tra một giá trị có là *null* hay không với kết quả là giá trị FALSE nếu phép so sánh có chứa *null*.

Ví dụ 3.16. Cho biết tên và mã số dự án mà địa điểm dự án là *null*:

```
SELECT Mada, Tenda
FROM Duan
WHERE Diadiemda IS NULL
```

Ví dụ 3.17. Liệt kê các mã nhân viên tham gia dự án ‘D1’ theo số giờ công giảm dần:

```
SELECT Manv
FROM Chamcong
WHERE Mada = ‘D1’
ORDER BY Sogio DESC
```

3.3.2.2. Sử dụng các hàm gộp và các câu GROUP BY, HAVING

SQL có năm hàm gộp còn gọi là hàm kết tập được cài sẵn như sau:

- *COUNT*: trả về số các giá trị đếm được trong một cột.
- *SUM*: trả về tổng các giá trị trong một cột.
- *AVG*: trả về trung bình cộng của các giá trị trong một cột.
- *MIN*: trả về giá trị nhỏ nhất trong số các giá trị ở một cột.
- *MAX*: trả về giá trị lớn nhất trong số các giá trị ở một cột.

Các hàm này thực hiện trên một cột của một bảng và trả về một giá trị. Các hàm *COUNT*, *MIN*, *MAX* áp dụng cho cột mà miền giá trị là kiểu số hay không phải số, nhưng hàm *SUM* và *AVG* chỉ áp dụng được cho các cột miền trị thuộc kiểu số. Các hàm gộp, trừ hàm *COUNT(*)*, đều bỏ qua giá trị *null* trong các giá trị đầu vào của chúng. Câu lệnh sau:

```
SELECT SUM(Sogio)
```

FROM **Chamcong**

luôn cho kết quả là một giá trị khác *null*, ngay cả khi có những giá trị *null* trong cột Sogio ở bảng **Chamcong**, miễn là ở cột này có xuất hiện giá trị khác *null*. Nếu muốn loại bỏ tình trạng có bản ghi lặp (chỉ giữ lại một) trước khi các hàm gộp được áp dụng, cần phải đặt từ khóa **DISTINCT** ngay trước tên cột được áp dụng hàm gộp. Đối với các hàm *MIN* và *MAX*, từ khóa *DISTINCT* không có tác dụng.

Ví dụ 3.18. Cho biết có bao nhiêu nhân viên lương cao hơn 10 triệu đồng:

```
SELECT COUNT(*) AS Vip
FROM Nhanvien
WHERE Luong > 10.000.000
```

Kết quả là một bảng có một cột (đặt tên Vip) và một hàng (số người thỏa yêu cầu đếm).

Ví dụ 3.19. Cho biết số nhân viên làm việc ở đơn vị có mã số 'P1' và tổng lương của họ:

```
SELECT COUNT(Manv) AS Songuoi, SUM(Luong) AS
Tongluong
FROM Nhanvien
WHERE Madv = 'P1'
```

Ví dụ 3.20. Tìm lương cao nhất, thấp nhất và trung bình cộng của các nhân viên:

```
SELECT MAX(Luong) AS Max, MIN(Luong) AS Min,
AVG(Luong) AS Tb
FROM Nhanvien
```

Bảng kết quả câu truy vấn này cho CSDL trên là:

Min	Max	Tb
5.000.000	10.000.000	6.812.500

SQL không chỉ cho phép áp dụng các hàm gộp trên một tập các bản ghi, mà còn cho phép áp dụng các hàm gộp trên các nhóm bản ghi có cùng giá trị trên một hay một số cột nào đó. Để làm điều này, trong câu truy vấn phải sử dụng từ khóa GROUP BY. Khi từ khóa GROUP BY được sử dụng, mỗi giá trị của mục được liệt kê ngay sau từ khóa SELECT là một giá trị cho một nhóm. Danh sách các mục đứng ngay sau từ khóa SELECT có thể: các tên cột, các hàm gộp, các hằng hay một biểu thức chứa tổ hợp của những mục trên. Cần lưu ý rằng tất cả tên cột xuất hiện trong danh sách sau từ khóa SELECT đều phải có mặt trong câu GROUP BY, chỉ trừ những tên cột có trong khai báo của hàm gộp.

Ví dụ 3.21. Cho biết lương cao nhất và tổng lương của mỗi đơn vị:

```
SELECT Madv, MAX(Luong) AS Maxdv, SUM(Luong) AS
      Tongdv
FROM Nhanvien
GROUP BY Madv
```

Khi thực hiện truy vấn này, các bản ghi có cùng giá trị ở cột Madv sẽ tạo thành một nhóm và hàm *MAX*, *SUM* được áp dụng cho mỗi nhóm bản ghi như vậy. Bảng kết quả của truy vấn này đối với bảng **Nhanvien** sẽ là

Madv	Maxdv	Tongdv
P1	10.000.000	10.000.000
P2	7.000.000	19.000.000
P3	5.500.000	10.500.000
P4	8.000.000	15.000.000

Trường hợp nếu chúng ta không muốn áp dụng hàm gộp cho mọi nhóm theo tác động của GROUP BY, mà chỉ áp dụng cho một số nhóm thỏa một điều kiện nào đó, thì SQL cung cấp từ khóa HAVING nhằm mục đích này. Điều kiện chọn các nhóm sẽ được áp dụng hàm gộp được viết trong câu HAVING.

Ví dụ 3.22. Cho biết lương cao nhất và tổng lương của các đơn vị có tổng lương trên 10 triệu đồng:

```
SELECT Madv, MAX(Luong) AS Maxdv, SUM(Luong) AS  
Tongdv  
FROM Nhanvien  
GROUP BY Madv  
HAVING SUM(Luong) > 10.000.000
```

Cần phân biệt mục đích của câu WHERE và câu HAVING. Mặc dù, cú pháp tương tự nhưng câu WHERE lọc lấy một số bản ghi nào đó trong một bảng để đưa vào bảng kết quả, còn câu HAVING lọc lấy một số nhóm nào đó để đưa vào bảng kết quả. Theo chuẩn ISO, các tên cột dùng trong câu HAVING cũng phải có mặt trong danh sách tên cột ở câu GROUP BY hoặc trong một hàm gộp. Thực tế là điều kiện chọn (lọc) trong câu HAVING luôn chứa ít nhất một hàm gộp, nếu không thì điều kiện này có thể chuyển vào câu WHERE. Nên nhớ, hàm gộp không được dùng trong câu WHERE.

Trường hợp câu truy vấn có cả câu WHERE và HAVING (dĩ nhiên có cả câu GROUP BY), thì trước tiên điều kiện chọn ở câu WHERE được áp dụng. Các bản ghi thỏa điều kiện này được tạo thành các nhóm do câu GROUP BY. Tiếp theo câu HAVING loại đi các nhóm không thỏa điều kiện đặt sau từ HAVING. Các nhóm còn lại được câu SELECT sử dụng để tạo ra các bản ghi là kết quả của truy vấn. Bởi vậy, bảng kết quả của truy vấn ở Ví dụ 3.22 là:

Madv	Maxdv	Tongdv
P2	7.000.000	19.000.000
P3	5.500.000	10.500.000
P4	8.000.000	15.000.000

3.3.2.3. Truy vấn lồng nhau

Một *truy vấn con* là một biểu thức SELECT ... FROM ... WHERE lồng trong một truy vấn khác. Kết quả của truy vấn con được sử dụng cho câu truy vấn SELECT ... FROM ... WHERE bên ngoài để quyết định kết quả trả ra. Lưu ý, truy vấn con có thể xuất hiện trong câu WHERE hay câu HAVING của một truy vấn ngoài. Nó cũng có thể xuất hiện trong các câu lệnh INSERT, UPDATE, DELETE. Có ba loại truy vấn con:

- *Truy vấn con vô hướng*: trả ra một bảng chỉ có một cột và một hàng (bản ghi), nghĩa là trả ra một giá trị. Về nguyên tắc một truy vấn con vô hướng có thể được đặt vào chỗ cần một giá trị.
- *Truy vấn con hàng*: trả ra một bảng có nhiều cột nhưng chỉ có một hàng. Một truy vấn hàng có thể được đặt vào chỗ cần có giá trị của một hàng.
- *Truy vấn con bảng*: trả ra một bảng có một hay nhiều cột và nhiều hàng. Một truy vấn con bảng có thể được dùng ở chỗ cần một bảng.

Ví dụ 3.23. Tìm mã số và họ tên các nhân viên phòng ‘Kỹ thuật’:

```
SELECT Manv, Hoten
FROM Nhanvien
WHERE Madv =
      (SELECT Madv
       FROM Phong
       WHERE Tenphong = ‘Kỹ thuật’);
```

Câu truy vấn con trong truy vấn trên được đặt trong các dấu ngoặc tròn ‘(...)’ thuộc loại vô hướng. Kết quả của truy vấn con này là mã số của phòng ‘Kỹ thuật’, chẳng hạn ‘P2’. Khi đó câu truy vấn bên ngoài làm việc như sau:

```
SELECT Manv, Hoten
FROM Nhanvien
```

WHERE Madv = 'P2'

Lúc đó bảng kết quả sẽ là

Manv	Hoten
2	Hoàng Văn Lợi
6	Hoàng Kim Phú
8	Đặng Tuấn

Ví dụ 3.24. Tìm họ tên của các nhân viên có lương đạt trên mức lương trung bình của tất cả các nhân viên, đồng thời cho biết mức chênh lệch này:

```
SELECT Hoten, Luong-(SELECT AVG(Luong) FROM Nhanvien)  
AS Hieu
```

```
FROM Nhanvien
```

```
WHERE Luong >
```

```
(SELECT AVG(Luong)
```

```
FROM Nhanvien);
```

Lúc đó bảng kết quả sẽ là

Hoten	Hieu
Nguyễn Hoàng	3.187.500
Nguyễn Thị Tú	1.187.500
Hoàng Kim Phú	187.500
Trần Quốc Sĩ	187.000

Ví dụ trên cho thấy, có thể sử dụng truy vấn con với một hàm gộp. Tuy nhiên, truy vấn con cần phải tuân thủ một số điều kiện sau:

- Câu ORDER BY không được sử dụng trong một truy vấn con, mặc dù nó có thể được dùng trong truy vấn ngoài cùng.
- Danh sách các mục được liệt kê bởi truy vấn con SELECT phải chứa tên của một cột hoặc một biểu thức, trừ phi câu truy vấn con dùng từ khóa EXIST.

- Theo ngầm định các tên cột trong truy vấn con tham chiếu đến tên bảng trong câu FROM của truy vấn con này hoặc có thể tham chiếu đến bảng trong câu FROM của truy vấn ngoài bằng việc xác định tên cột.
- Khi một câu truy vấn con là một trong hai toán hạng của một biểu thức so sánh thì truy vấn con này phải xuất hiện ở vế phải của so sánh.

Ví dụ 3.25. Cho biết thông tin về những nhân viên làm việc cho đơn vị có địa điểm là '10 Nguyễn Huệ':

```
SELECT *
FROM Nhanvien
WHERE Madv IN
    (SELECT Madv
     FROM Diadiemdv
     WHERE Diadiem = '10 Nguyễn Huệ');
```

Các từ khóa ALL và SOME (hay ANY) trong SQL luôn đi kèm một câu hỏi con mà câu hỏi con này trả ra kết quả chỉ là một cột. Ngoài ra trong SQL, phép so sánh “lớn hơn mọi” có thể biểu diễn bởi câu lệnh “> ALL”.

Ví dụ 3.26. Tìm họ tên và lương của những người có lương cao hơn lương của mọi nhân viên thuộc đơn vị mã P4:

```
SELECT Hoten
FROM Nhanvien
WHERE Luong > ALL
    (SELECT Luong
     FROM Nhanvien
     WHERE Madv = 'P4');
```

Bảng kết quả sẽ là:

Hoten	Luong
Nguyễn Hoàng	10.000.000

Ví dụ 3.27. Cho biết thông tin về những nhân viên có lương cao hơn ít nhất một nhân viên làm ở phòng 'Nghiên cứu và phát triển':

```

SELECT Hoten
FROM Nhanvien
WHERE Luong > SOME
      (SELECT Luong
       FROM Nhanvien
       WHERE Madv IN
            (SELECT Madv
             FROM Phong
             WHERE Tenphong = 'Nghiên cứu và phát
triển'))
);

```

SQL cũng cho phép các so sánh như: < ALL, <= ALL, >= ALL, <> ALL, < SOME, <= SOME, >= SOME, = SOME, <> SOME. SQL không cho phép dùng hàm hợp của các hàm gộp, chẳng hạn như *MAX(AVG(...))*.

Ví dụ 3.28. Cho biết tên phòng có trung bình lương cao nhất:

```

SELECT Tenphong
FROM Phong
WHERE Madv IN
      (SELECT Madv
       FROM Nhanvien
       GROUP BY Madv
       HAVING AVG(Luong) >= ALL
            (SELECT AVG(Luong)

```

FROM **Nhanvien**
GROUP BY Madv)

);

Câu hỏi con bên trong nhất sẽ cho kết quả là một bảng chỉ có một cột (Madv) và có bao nhiêu mã đơn vị thì cột này có bấy nhiêu giá trị. Mỗi giá trị là trung bình cộng lương của một phòng (tương ứng với một mã đơn vị). Câu hỏi con ở giữa được thực hiện như sau: đầu tiên các nhóm được tạo ra, mỗi nhóm gồm những bản ghi của bảng **Nhanvien** có cùng Madv, và mỗi nhóm được tính trung bình cộng theo lương. Kết quả của câu hỏi con này là mã đơn vị tương ứng của nhóm có trung bình cộng lương không thấp hơn mọi giá trị tính được của câu hỏi con bên trong nhất. Câu hỏi ngoài cùng sẽ chọn trong bảng **Phong** những bản ghi có mã đơn vị là một phần tử thuộc kết quả của câu hỏi con giữa, kết quả đó là một tập hợp gồm duy nhất một giá trị Madv. Và chỉ có một bản ghi được chọn vì Madv là khóa tối thiểu của bảng **Phong**, không phải giá trị mọi thuộc tính mà chỉ giá trị trên thuộc tính Tenphong của bản ghi này được đưa ra kết quả.

Hai từ khóa EXIST và NOT EXIST trong SQL cũng chỉ được sử dụng với các câu hỏi con. Các câu hỏi con này sẽ đưa ra kết quả là một trong hai giá trị TRUE và FALSE. Đó là EXIST(<câu hỏi con>) có giá trị TRUE nếu kết quả của <câu hỏi con> khác rỗng, và có giá trị FALSE trong trường hợp ngược lại. Câu hỏi con đi kèm với EXIST hay NOT EXIST dùng để kiểm tra một tập rỗng hay không nên bảng kết quả của câu hỏi con này có thể có hơn một cột.

Ví dụ 3.29. Cho biết họ tên tất cả nhân viên của những đơn vị (phòng) ở địa điểm '5 Lê Lợi':

```
SELECT Manv, Hoten  
FROM Nhanvien  
WHERE EXIST  
(SELECT *
```

```
FROM Diadiemdv
WHERE Nhanvien.Madv = Diadiemdv.Madv AND
      Diadiem = '5 Lê Lợi');
```

Lưu ý, chúng ta cũng có thể diễn giải câu hỏi trên theo một cách khác tương đương như sau: tìm các bản ghi trong bảng **Nhanvien** sao cho tồn tại một bản ghi trong bảng **Diadiemdv** chứa giá trị '5 Lê Lợi' (trên thuộc tính Diadiem) cũng có Madv như vậy. Lúc đó bảng kết quả sẽ là:

Manv	Hoten
5	Nguyễn Thị Tú
7	Trần Quốc Sĩ

Lưu ý, trong câu lệnh SQL ở trên, nếu chúng ta bỏ qua điều kiện **Nhanvien**.Madv = **Diadiemdv**.Madv thì sẽ dẫn đến kết quả là chiếu của các bản ghi trong bảng **Nhanvien** lên hai thuộc tính Manv và Hoten.

3.3.2.4. Dùng các biến bản ghi

Đôi khi chúng ta cần liên hệ đến hai hay nhiều hơn hai bản ghi trong cùng một bảng. Để thực hiện được điều này, chúng ta cần định nghĩa nhiều biến bản ghi cho bảng đó ở câu FROM và dùng những biến bản ghi này làm bí danh của bảng.

Ví dụ 3.30. Xét Ví dụ 3.29, chúng ta có thể viết gọn lại:

```
SELECT Manv, Hoten
FROM Nhanvien Nv
WHERE EXIST
      (SELECT *
       FROM Diadiemdv Dd
       WHERE Nv.Madv = Dd.Madv AND
            Diadiem = '5 Lê Lợi');
```

Ví dụ 3.31. Tìm thông tin về họ tên và lương của trưởng phòng 'Nghiên cứu và phát triển':

```
SELECT Nv.Hoten, Nv.Luong
FROM Nhanvien Nv, Phong P
WHERE Nv.Manv = P.Matp AND
      P.Tenphong = 'Nghiên cứu và phát triển'
```

Các biến bản ghi rất hữu ích khi so sánh hai bản ghi trong cùng một bảng.

Ví dụ 3.32. Tìm nhân viên có lương cao hơn ít nhất lương của một nam nhân viên:

```
SELECT DISTINCT a.Hoten, a.Luong
FROM Nhanvien a, Nhanvien b
WHERE a.Luong > b.Luong AND b.Gioitinh = 'Nam'
```

Lưu ý, theo cách viết của SQL, một tên đi theo sau bởi một tên khác không có dấu phân cách cho biết tên thứ hai là một bí danh của tên thứ nhất.

3.3.2.5. Các phép toán tập hợp

SQL-92 hỗ trợ các phép toán UNION, INTERSECT và EXCEPT tương ứng với các phép toán hợp, giao và hiệu của đại số quan hệ trên các bảng (quan hệ) tương thích.

Chẳng hạn, chúng ta xét CSDL như trên và giả sử có thêm bảng **Hoc** thể hiện thông tin về các nhân viên tham gia các lớp đào tạo nâng cao trình độ chuyên môn như sau:

Hoten	Ngaysinh	Gioitinh	Lop	Tienhoc
Nguyễn Hoàng	10/01/1970	Nam	Quản lý dự án	800.000
Phan Cẩm Ly	2/10/1980	Nữ	Anh văn giao tiếp	900.000
Nguyễn Thị Tú	15/11/1979	Nữ	Quản lý dự án	800.000

Ví dụ 3.33. Cho biết họ tên và ngày sinh của các nhân viên vừa lương thấp dưới 6 triệu đồng, vừa phải đóng tiền học phí.

Chúng ta có thể dùng phép toán INTERSECT để biểu diễn câu hỏi này như sau:

```
(SELECT Hoten, Ngaysinh
FROM Nhanvien
WHERE Luong < 6.000.000)
INTERSECT
(SELECT Hoten, Ngaysinh
FROM Hoc
WHERE Tienhoc > 0)
```

Khi đó bảng kết quả sẽ là:

Hoten	Ngaysinh
Phan Cẩm Ly	2/10/1980

Lưu ý, khác với SELECT, trong kết quả của phép toán UNION, INTERSECT và EXCEPT sẽ không xuất hiện các bản ghi trùng lặp. Còn khi dùng từ khóa UNION ALL, INTERSECT ALL hay EXCEPT ALL thay cho UNION hay INTERSECT hay EXCEPT thì các bản ghi trùng lặp không bị loại khỏi bảng kết quả.

3.3.2.6. Các bảng dẫn xuất và khung nhìn

SQL-92 cho phép dùng câu hỏi con trong câu FROM, trong đó bảng kết quả của câu hỏi con phải được đặt tên và các cột được phép đổi tên. Một kết quả như vậy được gọi là bảng (quan hệ) được dẫn xuất.

Ví dụ 3.34. Tìm lương trung bình của các phòng có lương trung bình lớn hơn 7 triệu đồng:

```
SELECT Maphong, Luongtb
FROM (SELECT Madv, AVG(Luong)
```

FROM **Nhanvien**

GROUP BY Madv)

AS **Ketqua**(Maphong, Luongtb)

WHERE Luongtb > 7.000.000)

Câu FROM cho kết quả là một bảng **Ketqua** có hai cột, cột thứ nhất với tên Maphong chứa Madv của mỗi nhóm và cột thứ hai với tên Luongtb chứa các giá trị trung bình của mỗi nhóm. Ở đây, chúng ta không cần dùng câu HAVING bởi vì câu WHERE dùng trực tiếp các cột trong bảng **Ketqua** để biểu diễn điều kiện chọn.

Chúng ta có thể cấu trúc lại các câu hỏi phức tạp giúp chúng trở nên dễ viết và dễ hiểu hơn bằng cách định nghĩa các khung nhìn.

- Cú pháp: CREATE VIEW <tên khung nhìn> AS <biểu thức truy vấn>
- Tác dụng: định nghĩa một khung nhìn.

Ví dụ 3.35. Định nghĩa khung nhìn có tên **Trocap** (trợ cấp) chứa họ tên, ngày sinh và giới tính của những nhân viên có lương dưới 8,1 triệu đồng và tham dự một khóa đào tạo:

CREATE VIEW **Trocap** AS

(SELECT Hoten, Ngaysinh, Gioitinh

FROM **Nhanvien**

WHERE Luong <8.100.000)

INTERSECT

(SELECT Hoten, Ngaysinh, Gioitinh

FROM **Hoc**

WHERE Tienhoc > 0)

Khi đó khung nhìn **Trocap** sẽ là:

Hoten	Ngaysinh	Gioitinh
Phan Cẩm Ly	2/10/1980	Nữ
Nguyễn Thị Tú	15/11/1979	Nữ

Sử dụng khung nhìn **Trocap** tạo được ở trên để tìm những nữ nhân viên lương thấp dưới 8,1 triệu đồng và phải đóng học phí, ta dùng lệnh:

```
SELECT Hoten
FROM Trocap
WHERE Gioitinh = 'Nữ'
```

3.4. Cấp phát và thu hồi quyền truy cập cơ sở dữ liệu

SQL-92 định nghĩa sáu dạng đặc quyền như sau:

- SELECT: quyền đưa ra những câu hỏi từ quan hệ.
- INSERT: quyền thêm bản ghi vào quan hệ.
- DELETE: quyền xóa quan hệ.
- UPDATE: quyền cập nhật dữ liệu quan hệ.
- REFERENCES: quyền tham khảo (chiếu) đến một quan hệ trong một ràng buộc về tính toàn vẹn.
- USAGE: quyền về cách sử dụng.

Trong đó, bốn loại đặc quyền đầu áp dụng cho các quan hệ là bảng cơ sở hoặc khung nhìn. Lưu ý, một mô đun chứa một câu lệnh SQL không thể thực hiện được nếu không có đặc quyền thích hợp cho câu lệnh đó. Chẳng hạn, câu lệnh SELECT...FROM...WHERE yêu cầu đặc quyền SELECT trên mỗi bảng mà nó truy nhập. Ngoài ra, có hai khía cạnh về cấp phát đặc quyền là làm thế nào để một NSD có được đặc quyền, và các đặc quyền sẽ được chuyển từ NSD này đến NSD khác ra sao. Người quản trị CSDL sẽ là người thực hiện lệnh khai báo NSD (hay nhóm NSD) CSDL và cấp các đặc quyền cho họ trên từng bảng của CSDL.

- Cú pháp: CREATE USER <tên NSD> IDENTIFIED BY <mật khẩu>
- Tác dụng: khai báo NSD.

Ví dụ 3.36. Khai báo một NSD mới có tên là Hieutruong với mật khẩu là sole13579:

```
CREATE USER Hieutruong IDENTIFIED BY sole13579
```

Câu lệnh cấp phát đặc quyền thường có dạng tổng quát như sau:

```
GRANT <danh sách đặc quyền> ON <phần tử CSDL> TO <danh sách NSD>
```

Lưu ý rằng phần tử CSDL là một quan hệ (bảng cơ sở hoặc khung nhìn).

Ví dụ 3.37. Cấp phát đặc quyền cho NSD Hieutruong:

```
GRANT SELECT, INSERT (Hoten) ON Nhanvien TO Hieutruong
```

Để thực hiện lệnh cấp phát này một cách hợp pháp, NSD thực hiện nó phải sở hữu những đặc quyền cấp phát. Một đặc quyền đã được cấp cũng có thể bị thu hồi bất kỳ lúc nào. Câu lệnh thu hồi đặc quyền có dạng như sau:

```
REVOKE <danh sách đặc quyền> ON <phần tử CSDL> FROM <danh sách NSD>
```

Ví dụ 3.38. Thu hồi những đặc quyền đã cấp phát cho Hieutruong ở ví dụ trên:

```
REVOKE SELECT, INSERT ON Nhanvien FROM Hieutruong
```

3.5. SQL nhúng

Qua các phần đã trình bày ở trên chúng ta có thể thấy SQL là một ngôn ngữ CSDL mạnh, cho phép diễn đạt khá dễ dàng các yêu cầu định nghĩa dữ liệu và thao tác dữ liệu. Tuy nhiên, trong nhiều trường hợp việc truy cập tới CSDL từ một ngôn ngữ lập trình bậc cao là cần thiết. Có thể kể ra hai lý do chính như sau: thứ nhất không phải mọi câu hỏi đều có thể biểu thị được bằng SQL, thứ hai SQL không thực hiện các

hành động phi khai báo như in một báo cáo hay tương tác với NSD hay gửi kết quả của một câu hỏi tới giao diện của NSD.

Các ứng dụng thường có nhiều thành phần trong đó việc hỏi hay truy cập dữ liệu chỉ là một thành phần, các thành phần khác được viết trong ngôn ngữ bậc cao hơn. Với một ứng dụng tích hợp, các chương trình được viết bằng ngôn ngữ lập trình phải có khả năng truy cập CSDL. Ngôn ngữ lập trình trong đó có nhúng các câu hỏi SQL được gọi là *ngôn ngữ chủ*, còn các cấu trúc của SQL được cho phép trong ngôn ngữ chủ làm thành *SQL nhúng*. Các chương trình được viết trong ngôn ngữ chủ có thể dùng cú pháp của SQL nhúng để truy cập và cập nhật dữ liệu trong CSDL. Trước khi dịch chương trình, các câu hỏi của SQL nhúng được thay thế bằng các khai báo và các lời gọi thủ tục trong ngôn ngữ chủ (được thực hiện bởi bộ tiền xử lý). Để bộ tiền xử lý nhận dạng được các lệnh của SQL nhúng, chương trình cần có câu lệnh dạng:

```
EXEC SQL <câu lệnh SQL nhúng> ENDEXEC
```

Lưu ý, trong Pascal hay C thì ENDEXEC được thay bằng dấu chấm phẩy (;). Ngoài ra, để sử dụng các biến đặc biệt cho việc trao đổi giữa chương trình và CSDL, chúng ta dùng từ khóa SQL INCLUDE trong chương trình. Các biến trong ngôn ngữ chủ có thể được dùng trong các câu lệnh của SQL nhúng, nhưng để phân biệt với các biến của SQL chúng được viết sau dấu hai chấm (:). Các cài đặt SQL-92 được hỗ trợ tại ít nhất 7 ngôn ngữ chủ sau: ADA, C, Cobol, Fortran, M, Pascal và PL/1.

BÀI TẬP CHƯƠNG 3

3.1. Cho hai tập thuộc tính $U_1 = \{a, b, c\}$, $U_2 = \{d, e, f\}$ và hai quan hệ tương ứng $R_1 \in Rel(U_1)$, $R_2 \in Rel(U_2)$. Hãy viết các biểu thức SQL tương ứng với mỗi một câu hỏi như sau:

- a) $R_1 \times R_2$
- b) $\Pi_{\{a,c\}}(R_1)$
- c) $\sigma_{e=1}(R_2)$
- d) $\Pi_{\{a,e\}}(\sigma_{b=f}(R_1 \times R_2))$.

3.2. Cho một tập thuộc tính $U = \{a, b, c\}$ và hai quan hệ $R_1, R_2 \in Rel(U)$. Hãy viết các biểu thức SQL tương ứng với mỗi một câu hỏi như sau:

- a) $R_1 \cup R_2$
- b) $R_1 \cap R_2$
- c) $R_1 \setminus R_2$
- d) $\Pi_{\{a,b\}}(R_1 \bowtie \Pi_{\{b,c\}}(R_2))$.

3.3. Xét một CSDL nhân viên gồm các LĐQH được cho bởi các thể hiện của chúng (ở đây giả sử không quan tâm đến các giá trị) như sau:

NHANVIEN(Hotennv, Duongpho, Thanhpho)

LAMVIEC(Hotennv, Tencongty, Luong)

CONGTY(Tencongty, Thanhpho)

QUANLY(Hotennv, Hotengiamdoc).

Hãy viết một biểu thức SQL cho mỗi câu hỏi sau:

a) Tìm họ tên và thành phố sinh sống của tất cả nhân viên làm việc cho công ty AMA.

b) Tìm họ tên, tên đường phố và thành phố sinh sống của tất cả các nhân viên làm việc cho công ty AMA có lương lớn hơn 3.000.000 đ.

c) Tìm tất cả nhân viên trong CSDL sống trong cùng thành phố của công ty nơi họ làm việc.

d) Tìm tất cả nhân viên trong CSDL sống trong cùng thành phố và trong cùng phố với giám đốc của họ.

e) Tìm tất cả nhân viên trong CSDL không làm việc cho AMA.

f) Tìm tất cả nhân viên có lương cao hơn mọi nhân viên của công ty AMA.

g) Giả sử các công ty được đặt tại nhiều thành phố. Tìm tất cả các công ty đặt tại mọi thành phố nơi có mặt công ty AMA.

h) Tìm tất cả nhân viên có lương cao hơn lương trung bình của mọi nhân viên của công ty của họ.

i) Tìm công ty có nhiều nhân viên nhất.

j) Tìm công ty có quỹ tiền lương nhỏ nhất.

k) Tìm các công ty mà nhân viên đều có lương trung bình cao hơn lương trung bình tại công ty AMA.

3.4. Xét CSDL nhân viên ở Câu 3.3. Dùng SQL định nghĩa một khung nhìn gồm họ tên giám đốc và lương trung bình của mọi nhân viên làm việc cho giám đốc đó. Giải thích tại sao hệ CSDL không được phép thực hiện các cập nhật được biểu thị theo khung nhìn đó.

Chương 4.

THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ

4.1. Dư thừa dữ liệu và các dị thường cập nhật

Mục đích chính của thiết kế CSDL quan hệ là nhóm các thuộc tính vào các bảng (quan hệ) sao cho giảm được nhiều nhất sự dư thừa dữ liệu và dẫn đến giảm được không gian lưu trữ cần thiết cho các quan hệ cơ sở. Ngoài ra, khi dữ liệu dư thừa một số vấn đề khác có thể nảy sinh như dị thường thêm bộ, dị thường xóa bộ và dị thường sửa bộ. Các dị thường này còn được gọi chung lại là dị thường cập nhật. Để minh họa chúng ta xét ví dụ sau:

Ví dụ 4.1. Xét các quan hệ *NCC* (người cung cấp) trên tập thuộc tính $\{Hoten, Diachi\}$ và quan hệ *CCH* (cung cấp hàng) trên tập thuộc tính $\{Hoten, Mathang, Gia\}$. Giả sử chúng ta đã kết nối hai quan hệ này thành quan hệ *NCCH* (người cung cấp hàng) với tập thuộc tính $\{Hoten, Diachi, Mathang, Gia\}$. Có thể thấy, quan hệ *NCCH* chứa tất cả các thông tin về người cung cấp hàng (cho siêu thị). Rõ ràng quan hệ này có dữ liệu dư thừa, đó là địa chỉ (duy nhất) của người cung cấp hàng lặp lại mỗi lần cho mỗi mặt hàng được cung cấp. Lúc này, các dị thường cập nhật có thể xuất hiện là:

- Dị thường khi sửa bộ: chúng ta có thể cập nhật địa chỉ mới của một người cung cấp hàng trong một bộ nhưng vẫn để lại địa chỉ cũ trong một bộ khác (do hậu quả của dư thừa). Khi đó, người cung cấp hàng không có địa chỉ duy nhất như chúng ta đã nghĩ (dữ liệu không nhất quán).

- Dị thường khi thêm bộ: chúng ta không thể biết được địa chỉ của một người cung cấp hàng nếu hiện tại họ không cung cấp ít nhất một mặt hàng. Có thể đặt những giá trị *null* trong các thuộc tính *Mathang*

và Gia của một bộ cho một người cung cấp hàng nào đó, nhưng khi thêm một mặt hàng cho người cung cấp hàng đó, chúng ta có nhớ xóa đi bộ mang giá trị $null$ hay không. Mặt khác, nếu giả thiết $\{Hoten, Mathang\}$ là một khóa tối thiểu của quan hệ $NCCH$, thì lúc đó chúng ta không thể tìm ra các bộ nhờ chỉ mục sơ cấp được nếu có những giá trị $null$ trong thuộc tính $Mathang$ của khóa tối thiểu.

- Dị thường khi xóa bộ: nếu giả sử chúng ta xóa tất cả các mặt hàng được cung cấp bởi một người cung cấp hàng thì vô ý làm mất dấu vết để tìm ra địa chỉ của người cung cấp hàng này.

Các dị thường cập nhật trên sẽ không tồn tại nữa nếu chúng ta tách quan hệ $NCCH$ thành hai quan hệ NCC và CCH . Khi đó quan hệ NCC cung cấp địa chỉ của mỗi người cung cấp hàng đúng một lần, do vậy không có dư thừa dữ liệu. Ngoài ra, chúng ta cũng có thể nhập địa chỉ của người cung cấp hàng dù hiện tại họ không cung cấp một mặt hàng nào.

Tuy vậy, vẫn còn một số vấn đề cần quan tâm khi thực hiện việc phân tách như trên, chẳng hạn để (truy vấn) tìm địa chỉ của tất cả những người cung cấp hàng có cung cấp một mặt hàng nào đó, thì đầu tiên chúng ta phải thực hiện một phép kết nối và sau đó thực hiện phép chọn, phép chiếu để trả lời truy vấn. Trong khi đối với quan hệ $NCCH$, chúng ta chỉ cần thực hiện một phép chọn và một phép chiếu đơn giản với thời gian thực hiện nhanh hơn. Vậy nên chúng ta cần xem sự thay thế ở trên lúc nào là có lợi. Ngoài ra, liệu còn có những vấn đề khác ngoài các vấn đề đề cập ở trên hay không? Chúng ta sẽ tìm một sự thay thế tốt đối với một LĐQH có các quan hệ thể hiện tòi như thế nào? Nội dung chương này sẽ cho phép trả lời những câu hỏi như vậy. Một trong những cách tiếp cận đối với vấn đề thiết kế CSDL quan hệ là thiết kế các LĐQH ở một dạng chuẩn thích hợp. Trọng tâm của việc thiết kế các LĐQH là các phụ thuộc dữ liệu, tức là các ràng buộc có thể giữa các tập thuộc tính của các LĐQH. Các phụ thuộc dữ liệu này chính là nguyên nhân gây nên sự dư thừa dữ liệu và các dị thường cập nhật.

4.2. Phụ thuộc hàm

Khái niệm phụ thuộc hàm (trên quan hệ) được giới thiệu bởi E. F. Codd vào những năm 1970, là một loại phụ thuộc dữ liệu xảy ra tự nhiên nhất giữa các tập thuộc tính và có tầm quan trọng hết sức lớn đối với việc thiết kế CSDL quan hệ. Mặc dù hiện nay có nhiều loại phụ thuộc dữ liệu được giới thiệu (như phụ thuộc mạnh, phụ thuộc yếu, phụ thuộc đối ngẫu, phụ thuộc đa trị, phụ thuộc Boole dương, ...), xong về cơ bản các hệ quản trị CSDL lớn đều sử dụng phụ thuộc hàm.

4.2.1. Định nghĩa

Cho một tập hữu hạn khác rỗng các thuộc tính $U = \{a_1, a_2, \dots, a_n\}$ và một quan hệ $R = \{t_1, t_2, \dots, t_m\} \in Rel(U)$. Một *phụ thuộc hàm* (PTH) trên U là một mệnh đề có dạng $X \rightarrow Y$ trong đó $X, Y \subseteq U$ và đọc là “ X xác định hàm Y ” hay “ Y phụ thuộc hàm vào X ”. Tập X gọi là vế trái, tập Y gọi là vế phải của phụ thuộc hàm $X \rightarrow Y$. PTH $X \rightarrow Y$ được gọi là *đúng* trên quan hệ R nếu

$$\forall t_i, t_j \in R : t_i(X) = t_j(X) \Rightarrow t_i(Y) = t_j(Y).$$

Khi PTH $X \rightarrow Y$ đúng trên quan hệ R thì ta còn nói quan hệ R *thỏa PTH* $X \rightarrow Y$ và ký hiệu là $R(X \rightarrow Y)$. Ký hiệu $FD(R)$ là tập tất cả các PTH đúng trên quan hệ R . Nghĩa là:

$$FD(R) = \{X \rightarrow Y : R(X \rightarrow Y)\}.$$

Trường hợp nếu quan hệ R *không thỏa PTH* $X \rightarrow Y$ thì ta viết $R(\neg X \rightarrow Y)$. Có thể thấy PTH (trên quan hệ) là sự phụ thuộc (theo nghĩa hàm) của một số thuộc tính vào một số thuộc tính khác.

Ví dụ 4.2. Xét quan hệ $R \in Rel(U)$, với $U = \{a, b, c\}$, như sau:

$$R = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline 2 & 0 & 0 \\ \hline \end{array}$$

Ta có các PTH đúng trên R là $\{a\} \rightarrow \{b\}$, $\{a\} \rightarrow \{c\}$, $\{a\} \rightarrow \{b, c\}$, $\{c\} \rightarrow \{b\}$, $\{a, b\} \rightarrow \{c\}$ và $\{a, c\} \rightarrow \{b\}$. Các PTH không đúng trên R là $\{b\} \rightarrow \{a\}$, $\{b\} \rightarrow \{c\}$, $\{c\} \rightarrow \{a\}$, $\{c\} \rightarrow \{a, b\}$ và $\{b, c\} \rightarrow \{a\}$.

Lưu ý, trên tập thuộc tính U có thể có nhiều quan hệ R khác nhau nhưng tập $FD(R)$ lại như nhau. Chẳng hạn, hai quan hệ khác nhau $R_1, R_2 \in Rel(U)$ với $U = \{a, b\}$ có $FD(R_1) = FD(R_2)$:

$$R_1 = \begin{array}{|c|c|} \hline a & b \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline \end{array}, \quad R_2 = \begin{array}{|c|c|} \hline a & b \\ \hline 0 & 1 \\ \hline 1 & 1 \\ \hline \end{array}$$

Để đơn giản, trong lý thuyết CSDL người ta thường viết gọn lại về trái và về phải của PTH dưới dạng xâu ký tự. Chẳng hạn, PTH $\{a, b, c\} \rightarrow \{d\}$ sẽ được viết lại là $abc \rightarrow d$. Ngoài ra, trong lý thuyết CSDL chúng ta cũng sẽ không quan tâm đến một số dạng PTH tầm thường sau:

- $X \rightarrow \emptyset$: PTH này đúng với bất kỳ quan hệ $R \in Rel(U)$ nào với $X \subseteq U$.
- $\emptyset \rightarrow Y$: PTH này đúng với quan hệ $R \in Rel(U)$ có tất cả các bộ có giá trị bằng nhau trên $Y \subseteq U$. Trong trường hợp này, với mọi $\emptyset \neq X \subseteq U$ ta đều có $R(X \rightarrow Y)$.

Cho tập PTH F trên tập thuộc tính U . Quan hệ $R \in Rel(U)$ được gọi là *thỏa tập PTH F* , ký hiệu $R(F)$, nếu với mọi $X \rightarrow Y \in F$ thì $R(X \rightarrow Y)$. Ngược lại, nếu tồn tại $X \rightarrow Y \in F$ sao cho $R(\neg X \rightarrow Y)$ thì ta nói R *không thỏa tập PTH F* và viết $R(\neg F)$. Tập tất cả các quan hệ R thỏa F ký hiệu là $Sat_U(F)$ hoặc $Sat(F)$ nếu không sợ nhầm lẫn.

Như vậy

$$Sat(F) = \{R : R(F)\} = \bigcap_{X \rightarrow Y \in F} Sat(\{X \rightarrow Y\}).$$

Tập $FD(R)$ và $Sat(F)$ có một số tính chất cơ bản sau

Mệnh đề 4.1. Cho hai quan hệ $R_1, R_2 \in Rel(U)$.

- 1) Nếu $R_1 \subseteq R_2$ thì $FD(R_2) \subseteq FD(R_1)$.
- 2) $FD(R_1 \cup R_2) \subseteq FD(R_1) \cap FD(R_2)$.

Chứng minh. 1) Giả sử PTH $X \rightarrow Y \in FD(R_2)$ và lấy tùy ý hai bộ $t_i, t_j \in R_1$ sao cho $t_i(X) = t_j(X)$. Vì $R_1 \subseteq R_2$ nên $t_i, t_j \in R_2$ và do đó $t_i(Y) = t_j(Y)$. Theo định nghĩa suy ra $X \rightarrow Y \in FD(R_1)$.

2) Suy ra từ 1). □

Mệnh đề 4.2. Cho hai tập PTH F và G trên U .

- 1) Nếu $F \subseteq G$ thì $Sat(G) \subseteq Sat(F)$.
- 2) $Sat(F \cup G) = Sat(F) \cap Sat(G)$.

Chứng minh. 1) Lấy bất kỳ quan hệ $R \in Sat(G)$. Vì $F \subseteq G$ nên suy ra $R(F)$, hay $R \in Sat(F)$.

2) Suy ra từ định nghĩa của Sat và 1). □

Bây giờ một LDQH được hiểu là một cặp $S = (U, F)$, trong đó U là tập các thuộc tính và F là tập các PTH trên U . Khi về trái, về phải của các PTH trong F đều có đúng một thuộc tính thì S còn được gọi là *LDQH đơn*. Đây là trường hợp đặc biệt thú vị của LDQH, nhiều bài toán quan trọng trong lý thuyết CSDL có độ phức tạp hàm mũ trên LDQH bất kỳ nhưng trên LDQH đơn thì chỉ có độ phức tạp đa thức [28].

4.2.2. Suy diễn theo quan hệ

Cho F là một tập các PTH trên tập thuộc tính U và $X \rightarrow Y$ là một PTH bất kỳ. Ta nói F suy diễn theo quan hệ $X \rightarrow Y$, ký hiệu $F \models X \rightarrow Y$, nếu với mọi quan hệ $R \in Rel(U)$ sao cho $R(F)$ thì $R(X \rightarrow Y)$.

Như vậy, $F \models X \rightarrow Y$ khi và chỉ khi $Sat(F) \subseteq Sat(\{X \rightarrow Y\})$. Trường hợp nếu F không suy diễn theo quan hệ được PTH $X \rightarrow Y$ thì ta viết $F \not\models X \rightarrow Y$. Đặt

$$F^* = \{X \rightarrow Y : F \models X \rightarrow Y\}.$$

Rõ ràng $X \rightarrow Y \notin F^*$ khi và chỉ khi $F \not\models X \rightarrow Y$.

Ví dụ 4.3. Xét LDQH $S = (U, F)$ với $U = \{a, b, c\}$ và $F = \{a \rightarrow b, b \rightarrow c\}$.

Ta có $F \models a \rightarrow c$. Thật vậy, lấy bất kỳ quan hệ $R \in Rel(U)$ sao cho $R(F)$. Sau đó, chọn tùy ý hai bộ $t_i, t_j \in R$ sao cho $t_i(\{a\}) = t_j(\{a\})$. Từ đây và giả thiết $R(a \rightarrow b)$, suy ra $t_i(\{b\}) = t_j(\{b\})$. Tương tự, từ đây và giả thiết $R(b \rightarrow c)$, suy ra tiếp $t_i(\{c\}) = t_j(\{c\})$. Theo định nghĩa, điều này có nghĩa $R(a \rightarrow c)$ hay $F \models a \rightarrow c$.

Có thể hiểu một cách nôm na suy diễn theo quan hệ là “ở đâu F thỏa thì ở đó $X \rightarrow Y$ cũng thỏa”. Điều này cho thấy, chúng ta khó có khả năng xây dựng một thuật toán hữu hiệu để tính đúng tập F^* . Để giải quyết vấn đề này một cách hiệu quả, năm 1974 W. W. Armstrong [1] tiên đề hóa khái niệm PTH bằng cách xây dựng một hệ qui tắc suy diễn cho phép tính đúng tập F^* .

4.2.3. Hệ tiên đề Armstrong

Xét tập PTH F trên tập thuộc tính U . Bao đóng của F , ký hiệu F^+ , là tập các PTH trên U nhỏ nhất chứa F thỏa các tính chất sau: với mọi $X, Y, Z \subseteq U$

- F1) *Tính phản xạ*: Nếu $Y \subseteq X$ thì $X \rightarrow Y \in F^+$
- F2) *Tính gia tăng*: Nếu $X \rightarrow Y \in F^+$ thì $X \cup Z \rightarrow Y \cup Z \in F^+$
- F3) *Tính bắc cầu*: Nếu $X \rightarrow Y \in F^+$ và $Y \rightarrow Z \in F^+$ thì $X \rightarrow Z \in F^+$.

Các tính chất (F1)-(F3) còn được gọi là *tập qui tắc suy diễn Armstrong* hay *hệ tiên đề Armstrong*. Vì U là tập hữu hạn nên bao đóng F^+ cũng hữu hạn. Trường hợp nếu $Y = X$ thì tính chất (F1) còn được gọi là *tính phản xạ chặt*. Các PTH $X \rightarrow Y$ được suy ra từ tính phản xạ còn được gọi là *PTH tầm thường*, tức là các PTH mà vế trái bao hàm vế phải. Các PTH như thế này đúng trong mọi quan hệ, chúng nói lên rằng việc sử dụng qui tắc này chỉ phụ thuộc vào tập thuộc tính U , không phụ thuộc vào tập PTH F .

Ví dụ 4.4. Cho tập PTH $F = \{a \rightarrow b, b \rightarrow c\}$ trên tập thuộc tính $U = \{a, b, c\}$. Vận dụng các qui tắc suy diễn Armstrong từ F ta thu được $a \rightarrow U \in F^+$. Thật vậy, sử dụng tính gia tăng, từ $a \rightarrow b \in F^+$ suy ra $a \rightarrow ab \in F^+$, từ $b \rightarrow c \in F^+$ suy ra $b \rightarrow bc \in F^+$ và do đó $ab \rightarrow U \in F^+$. Từ $a \rightarrow ab \in F^+$ và $ab \rightarrow U \in F^+$, áp dụng tính bắc cầu ta thu được $a \rightarrow U \in F^+$.

Tập PTH F được gọi là *suy diễn theo tiên đề* PTH $X \rightarrow Y$, ký hiệu $F \models_A X \rightarrow Y$, nếu $X \rightarrow Y \in F^+$. Nói một cách khác, PTH $X \rightarrow Y$ được suy diễn theo tiên đề từ F nếu chúng ta xuất phát từ F rồi áp dụng các qui tắc suy diễn từ (F1) đến (F3) sau một số hữu hạn lần thì thu được $X \rightarrow Y$. Khi F không suy diễn theo tiên đề được $X \rightarrow Y$ thì ta viết $F \not\models_A X \rightarrow Y$.

Chẳng hạn, xét Ví dụ 4.4 trên ta có $F \models_A a \rightarrow c$ và $F \models_A a \rightarrow bc$ nhưng $F \not\models_A c \rightarrow b$.

Bổ đề 4.1 (Tính đúng của hệ tiên đề). *Hệ tiên đề Armstrong là đúng, nghĩa là $F^+ \subseteq F^*$.*

Chứng minh. Bổ đề khẳng định nếu PTH bất kỳ $Z \rightarrow W$ được suy diễn

theo tiên đề từ tập PTH F thì $Z \rightarrow W$ sẽ đúng trong mọi quan hệ thỏa F . Như vậy, chúng ta sẽ xét 3 trường hợp sau:

1) $X \rightarrow Y$ được suy diễn từ (F1): trường hợp này đúng với bất kỳ quan hệ nào, vì không thể tồn tại hai bộ có giá trị bằng nhau trên X nhưng lại khác nhau trên tập con Y của nó.

2) $X \cup Z \rightarrow Y \cup Z$ được suy diễn từ (F2): lấy bất kỳ quan hệ $R \in Rel(U)$ sao cho $R(X \rightarrow Y)$ và hai bộ tùy ý $t_i, t_j \in R$ sao cho $t_i(X \cup Z) = t_j(X \cup Z)$. Lúc này $t_i(X) = t_j(X)$ và $t_i(Z) = t_j(Z)$. Từ $t_i(X) = t_j(X)$ và $R(X \rightarrow Y)$ theo định nghĩa suy ra $t_i(Y) = t_j(Y)$, và do đó $t_i(Y \cup Z) = t_j(Y \cup Z)$. Như vậy $R(X \cup Z \rightarrow Y \cup Z)$.

3) $X \rightarrow Z$ được suy diễn từ (F3): xét bất kỳ quan hệ $R \in Rel(U)$ sao cho $R(\{X \rightarrow Y, Y \rightarrow Z\})$ và hai bộ tùy ý $t_i, t_j \in R$ sao cho $t_i(X) = t_j(X)$. Từ $t_i(X) = t_j(X)$ và $R(X \rightarrow Y)$ suy ra $t_i(Y) = t_j(Y)$. Từ $t_i(Y) = t_j(Y)$ và $R(Y \rightarrow Z)$ suy ra $t_i(Z) = t_j(Z)$. Tóm lại, chúng ta có $R(X \rightarrow Z)$. $\square$

Như vậy, Bổ đề 4.1 khẳng định vận dụng hệ tiên đề Armstrong chỉ có thể cho ra các PTH được suy diễn theo quan hệ từ F . Tức là suy diễn theo tiên đề là suy diễn theo quan hệ. Từ hệ tiên đề Armstrong trên, chúng ta suy ra được một số qui tắc suy diễn khác. Các qui tắc này cũng đóng vai trò quan trọng trong suy diễn PTH.

Mệnh đề 4.3 (Một số qui tắc suy diễn khác). *Với mọi $X, Y, Z, W \subseteq U$ ta có*

F4) Tính cộng tính phải:

Nếu $X \rightarrow Y \in F^+$ và $X \rightarrow Z \in F^+$ thì $X \rightarrow Y \cup Z \in F^+$.

F5) Tính thu hẹp phải (hay tách):

Nếu $X \rightarrow Y \in F^+$ thì $X \rightarrow Z \in F^+$ với mọi $Z \subseteq Y$.

F6) Tính giả bắc cầu:

Nếu $X \rightarrow Y \in F^+$ và $Y \cup W \rightarrow Z \in F^+$ thì $X \cup W \rightarrow Z \in F^+$.

F7) Tính cộng tính đầy đủ:

Nếu $X \rightarrow Y \in F^+$ và $W \rightarrow Z \in F^+$ thì $X \cup W \rightarrow Y \cup Z \in F^+$.

F8) Tính mở rộng trái và thu hẹp phải:

Nếu $X \rightarrow Y \in F^+$ thì $X \cup W \rightarrow Y \setminus Z \in F^+$.

Chứng minh. F4) Vận dụng tính gia tăng, từ $X \rightarrow Y \in F^+$ suy ra $X \rightarrow X \cup Y \in F^+$ và từ $X \rightarrow Z \in F^+$ suy ra $X \cup Y \rightarrow Y \cup Z \in F^+$. Sau đó, từ hai PTH thu được $X \rightarrow X \cup Y \in F^+$ và $X \cup Y \rightarrow Y \cup Z \in F^+$, áp dụng tính bắc cầu ta có $X \rightarrow Y \cup Z \in F^+$.

F5) Theo tính phản xạ, với mọi $Z \subseteq Y$ ta có $Y \rightarrow Z \in F^+$. Từ đây và giả thiết $X \rightarrow Y \in F^+$, vận dụng tính bắc cầu ta thu được $X \rightarrow Z \in F^+$.

F6) Từ giả thiết $X \rightarrow Y \in F^+$, vận dụng tính gia tăng suy ra $X \cup W \rightarrow Y \cup W \in F^+$. Từ đây và giả thiết $Y \cup W \rightarrow Z \in F^+$, áp dụng tính bắc cầu ta nhận được $X \cup W \rightarrow Z \in F^+$.

F7) Vận dụng tính gia tăng, từ $X \rightarrow Y \in F^+$ và $W \rightarrow Z \in F^+$ suy ra $X \cup W \rightarrow Y \cup W \in F^+$ và $Y \cup W \rightarrow Y \cup Z \in F^+$. Vận dụng tính bắc cầu, từ hai PTH thu được này ta có $X \cup W \rightarrow Y \cup Z \in F^+$.

F8) Theo tính phản xạ ta có $X \cup W \rightarrow X \in F^+$. Từ đây và giả thiết $X \rightarrow Y \in F^+$, vận dụng tính bắc cầu suy ra $X \cup W \rightarrow Y \in F^+$. Từ PTH thu được này, áp dụng tính thu hẹp phải ta có ngay $X \cup W \rightarrow Y \setminus Z \in F^+$. $\square$

Từ tính chất cộng tính phải và tính chất thu hẹp phải, chúng ta có ngay hệ quả sau.

Hệ quả 4.1. $X \rightarrow a_1 a_2 \dots a_n \in F^+$ khi và chỉ khi $X \rightarrow a_i \in F^+$ với mọi $i = 1, 2, \dots, n$.

Bao đóng tập PTH có một số tính chất cơ bản sau.

Mệnh đề 4.4. Cho F và G là hai tập PTH trên tập thuộc tính U . Khi đó

1) Tính phản xạ: $F \subseteq F^+$.

2) Tính đơn điệu: nếu $F \subseteq G$ thì $F^+ \subseteq G^+$.

3) Tính lũy đẳng: $(F^+)^+ = F^+$.

Chứng minh. 1) Hiển nhiên theo định nghĩa bao đóng F^+ .

2) Bởi tính phản xạ ta có $G \subseteq G^+$. Suy ra $F \subseteq G^+$. Từ đây và theo định nghĩa bao đóng F^+ , ta thu được $F^+ \subseteq G^+$.

3) Theo tính phản xạ và tính đơn điệu ta có $F^+ \subseteq (F^+)^+$. Hơn nữa $F^+ \subseteq F^+$, nên từ định nghĩa bao đóng $(F^+)^+$ suy ra $(F^+)^+ \subseteq F^+$. $\square$

Sau đây là hai bài toán quan trọng trong lý thuyết thiết kế CSDL:

Bài toán 4.1 (Bài toán thành viên). Cho tập PTH F trên tập thuộc tính U và một PTH $X \rightarrow Y$. Xác định xem $X \rightarrow Y \in F^+$ hay không?

Bài toán 4.2 (Bài toán suy diễn phụ thuộc). Cho tập thuộc tính U và một quan hệ $R \in Rel(U)$. Tìm một tập các PTH F trên U sao cho $F^+ = FD(R)$.

Để ý rằng $(FD(R))^+ = FD(R)$. Do đó, điều kiện $F^+ = FD(R)$ tương đương với điều kiện $F^+ = (FD(R))^+$. Nếu thỏa điều kiện này thì trong mục sau tập F sẽ được gọi là một *phủ* của $FD(R)$. Vì vậy, bài toán suy diễn phụ thuộc còn được hiểu là đi tìm một phủ F của $FD(R)$. Hai bài toán này sẽ được giải quyết ở trong mục 4.2.5 sau.

4.2.4. Bao đóng của thuộc tính

Xét LĐQH $S = (U, F)$ và tập con thuộc tính $X \subseteq U$. Vận dụng tính cộng tính phải ta luôn tìm được PTH $X \rightarrow Y \in F^+$ sao cho Y là tập lớn nhất theo nghĩa, với mọi PTH $X \rightarrow Z \in F^+$ thì $Z \subseteq Y$. Tập Y như vậy được gọi là *bao đóng* của X (ứng với S hay F), ký hiệu X_F^+ hay X^+ nếu không sợ nhầm lẫn.

Như vậy

$$X^+ = \{a \in U : X \rightarrow a \in F^+\}.$$

Trường hợp $X^+ = X$, thì X được gọi là *tập đóng* (hay *điểm bất động*). Tập tất cả các tập đóng ký hiệu là $Closed(S)$ hay $Closed(F)$. Rõ ràng $U \in Closed(F)$. Hai tập thuộc tính X và Y được gọi là *tương đương* trong S nếu $X^+ = Y^+$. Đặc biệt, khi $X = \{a\}, Y = \{b\}$ và $X^+ = Y^+$ thì ta nói hai thuộc tính a và b là tương đương với nhau.

Ví dụ 4.5. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c\}$ và $F = \{a \rightarrow b, b \rightarrow c\}$. Theo định nghĩa, ta có $\{a\}^+ = \{a, b, c\}$, $\{b, c\}^+ = \{b, c\}$.

Bao đóng của thuộc tính có các tính chất cơ bản sau.

Mệnh đề 4.5. Cho LĐQH $S = (U, F)$ và $X, Y \subseteq U$. Khi đó

- 1) Tính phản xạ: $X \subseteq X^+$.
- 2) Tính đơn điệu: nếu $X \subseteq Y$ thì $X^+ \subseteq Y^+$.
- 3) Tính lũy đẳng: $(X^+)^+ = X^+$.
- 4) $(X \cup Y)^+ \supseteq X^+ \cup Y^+$ và $(X \cap Y)^+ \subseteq X^+ \cap Y^+$.
- 5) $(X^+ \cup Y)^+ = (X \cup Y^+)^+ = (X \cup Y)^+$.
- 6) $X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X^+$.
- 7) $X^+ \rightarrow X \in F^+$ và $X \rightarrow X^+ \in F^+$.
- 8) $X^+ = Y^+ \Leftrightarrow X \rightarrow Y \in F^+$ và $Y \rightarrow X \in F^+$.

Chứng minh. 1) Hiển nhiên theo định nghĩa bao đóng X^+ .

7) Suy ra từ tính chất 1) và định nghĩa bao đóng của tập thuộc tính.

2) Vì $X \subseteq Y$ nên theo tính phản xạ $Y \rightarrow X \in F^+$. Từ đây và tính chất 7), vận dụng tính bắc cầu ta thu được $Y \rightarrow X^+ \in F^+$. Biết rằng, Y^+ là tập lớn nhất sao cho $Y \rightarrow Y^+ \in F^+$, nên suy ra $X^+ \subseteq Y^+$.

3) Theo tính chất 7) ta có $X \rightarrow X^+ \in F^+$ và $X^+ \rightarrow (X^+)^+ \in F^+$. Từ đây và tính bắc cầu suy ra $X \rightarrow (X^+)^+ \in F^+$. Hơn nữa, biết X^+ là tập lớn nhất sao cho $X \rightarrow X^+ \in F^+$. Điều này có nghĩa $(X^+)^+ \subseteq X^+$. Ngoài ra, theo tính phản xạ và tính đơn điệu chúng ta có $X^+ \subseteq (X^+)^+$.

4) Suy ra ngay từ tính chất 1) và 2).

6) Theo định nghĩa bao đóng X^+ và tính chất 7), rõ ràng nếu $X \rightarrow Y \in F^+$ thì $Y \subseteq X^+$.

Ngược lại, nếu $Y \subseteq X^+$ thì suy ra $X \rightarrow a \in F^+$ với mọi $a \in Y$. Lúc này theo Hệ quả 4.1 ta có ngay $X \rightarrow Y \in F^+$.

5) Chỉ cần chứng minh đẳng thức $(X^+ \cup Y)^+ = (X \cup Y)^+$, sau đó hoán đổi vai trò của X và Y thì chúng ta thu được tính chất 5). Thật vậy theo tính chất 1) và 2), ta suy ra ngay $(X \cup Y)^+ \subseteq (X^+ \cup Y)^+$.

Theo tính chất 7) và tính giao tăng suy ra $X \cup Y \rightarrow X^+ \cup Y \in F^+$. Cũng theo tính chất 7) thì $X^+ \cup Y \rightarrow (X^+ \cup Y)^+ \in F^+$. Do đó, theo tính bắc cầu suy ra $X \cup Y \rightarrow (X^+ \cup Y)^+ \in F^+$. Từ đây và theo định nghĩa bao đóng của $(X \cup Y)^+$, chúng ta thu được $(X^+ \cup Y)^+ \subseteq (X \cup Y)^+$.

8) Theo tính bắc cầu và tính chất 7), nếu $X^+ = Y^+$ thì ta có ngay $X \rightarrow Y \in F^+$ và $Y \rightarrow X \in F^+$.

Ngược lại, theo tính bắc cầu và tính chất 7), nếu $X \rightarrow Y \in F^+$ và $Y \rightarrow X \in F^+$ thì suy ra $Y \rightarrow X^+ \in F^+$. Lúc này, theo định nghĩa bao đóng Y^+ , ta có ngay $X^+ \subseteq Y^+$. Vì vai trò của X và Y là như nhau, nên ta cũng thu được $Y^+ \subseteq X^+$. $\square$

Mệnh đề 4.6. Nếu X, Y là các tập đóng thì

$$(X \cap Y)^+ = X^+ \cap Y^+.$$

Chứng minh. Theo tính phản xạ của bao đóng của tập thuộc tính, nếu X và Y là các tập đóng thì

$$(X \cap Y)^+ \supseteq X \cap Y = X^+ \cap Y^+.$$

Ngoài ra, theo Mệnh đề 4.5:

$$(X \cap Y)^+ \subseteq X^+ \cap Y^+.$$

Do đó $(X \cap Y)^+ = X^+ \cap Y^+$. $\square$

Như vậy, nếu X và Y là các tập đóng thì $X \cap Y$ cũng là các tập đóng. Tức là, tập tất cả các tập đóng là đóng đối với phép toán giao. Ngoài ra, dễ thấy bao đóng X^+ là tập đóng nhỏ nhất chứa X .

Định lý 4.7 (Tính đúng đắn và đầy đủ). *Hệ tiên đề Armstrong là*

- 1) *Đúng: nghĩa là $F^+ \subseteq F^*$*
- 2) *Đầy đủ: nghĩa là $F^+ = F^*$.*

Chứng minh. Tính đúng của 1) đã được khẳng định ở Bổ đề 4.1. Bây giờ chúng ta chỉ còn chứng minh tính đủ của hệ tiên đề. Tính đủ được chứng minh theo lược đồ sau: nếu có một PTH $X \rightarrow Y$ trên U sao cho $X \rightarrow Y \notin F^+$ thì phải tồn tại một quan hệ $R \in Rel(U)$ sao cho $R(F)$ (thậm chí $R(F^+)$) nhưng $R(\neg X \rightarrow Y)$.

Thật vậy, xét một quan hệ $R = \{t_1, t_2\} \in Rel(U)$, với $U = \{a_1, a_2, \dots, a_n\}$ như sau:

$$t_1 = (0, 0, \dots, 0)$$

và

$$t_2(a_i) = \begin{cases} 0 & \text{nếu } a_i \in X^+ \\ 1 & \text{ngược lại} \end{cases}$$

với mọi $i = 1, 2, \dots, n$.

Theo cách xây dựng quan hệ R , nếu tồn tại một PTH $Z \rightarrow W \in F$ sao cho $R(\neg Z \rightarrow W)$ thì $Z \subseteq X^+$ và $W \not\subseteq X^+$. Theo Mệnh đề 4.5, với $Z \subseteq X^+$ thì $X \rightarrow Z \in F^+$, và do đó theo tính bắc cầu ta có $X \rightarrow W \in F^+$ hay $W \subseteq X^+$. Điều này là mâu thuẫn. Suy ra $R(F)$.

Như vậy, theo lược đồ chúng ta chỉ còn chứng minh thêm quan hệ $R(\neg X \rightarrow Y)$. Chứng minh bằng phản chứng. Giả sử $R(X \rightarrow Y)$. Bởi tính phản xạ của bao đóng X^+ , nên từ quan hệ R xây dựng ở trên chúng ta phải có $Y \subseteq X^+$. Theo Mệnh đề 4.5 suy ra $X \rightarrow Y \in F^+$. Điều này trái với giả thiết ban đầu $X \rightarrow Y \notin F^+$. Vậy ta phải có $R(\neg X \rightarrow Y)$. $\square$

Định lý 4.7 khẳng định rằng suy diễn theo quan hệ và suy diễn theo tiên đề là một, tức là

$$F \models X \rightarrow Y \Leftrightarrow F \models_A X \rightarrow Y.$$

Như vậy, từ nay về sau chúng ta sẽ không phân biệt giữa suy diễn theo quan hệ và suy diễn theo tiên đề khi thảo luận về PTH. Đôi lúc để ngắn gọn ta chỉ cần gọi là suy diễn.

Xét LDQH $S = (U, F)$. Quan hệ $R \in Rel(U)$ thỏa mãn $FD(R) = F^+$ được gọi là *quan hệ Armstrong* hay *thể hiện LDQH S*. Rõ ràng một LDQH có thể có nhiều thể hiện. Đây là khái niệm được đưa ra bởi Fagin R. (1984), có vai trò rất quan trọng trong quá trình nghiên cứu về cấu trúc logic của MHDL quan hệ. Chứng minh của Định lý 4.7 hướng dẫn cho chúng ta thấy một quan hệ Armstrong của S luôn tồn tại. Sau đây là một điều kiện cần và đủ để R là một quan hệ Armstrong của S [28] (1987).

Định lý 4.8. *Cho quan hệ $R = \{t_1, t_2, \dots, t_m\} \in Rel(U)$ và LDQH $S = (U, F)$. Điều kiện cần và đủ để R là quan hệ Armstrong của S là*

$$X^+ = \begin{cases} \bigcap_{X \subseteq E_{ij}} E_{ij} & \text{nếu } \exists E_{ij} \in E(R) : X \subseteq E_{ij} \\ U & \text{ngược lại} \end{cases}$$

với mọi $X \subseteq U$.

Phần chứng minh của Định lý 4.8 được dành làm bài tập (Tuy vậy người đọc có thể tham khảo qua phần chứng minh của Định lý 4.10 ở mục 4.2.6 phía sau). Bài toán xây dựng quan hệ Armstrong của LDQH S rất khó, có độ phức tạp hàm mũ theo kích thước của S [9] (1990):

Bài toán 4.3 (Xây dựng quan hệ Armstrong). Cho LDQH $S = (U, F)$. Xây dựng quan hệ $R \in Rel(U)$ sao cho R là quan hệ Armstrong của S .

Bài toán này đóng một vai trò cực kỳ quan trọng trong lý thuyết

thiết kế CSDL. Một chứng minh về độ phức tạp sử dụng hoàn toàn công cụ siêu đồ thị cho bài toán này có thể xem trong [16] (2006).

4.2.5. Một số thuật toán cơ bản

Sau đây là thuật toán tìm bao đóng của thuộc tính trên LDQH.

Thuật toán 4.1 (Thuật toán tìm bao đóng thuộc tính)

Vào: LDQH $S = (U, F)$ và $X \subseteq U$.

Ra: Bao đóng X^+ .

Phương pháp:

Xây dựng dãy tập thuộc tính bao hàm nhau $X_0 \subseteq X_1 \subseteq \dots \subseteq X_k$ theo qui tắc:

Bước 1. Đặt $X_0 := X$.

Bước 2. Tính lặp đi lặp lại ($i = 0, 1, \dots$):

$$X_{i+1} := X_i \cup Z_i$$

trong đó $Z_i = \{a \in (U \setminus X_i) : \exists Z \rightarrow W \in F, a \in W, Z \subseteq X_i\}$ và X_i được giả sử đã tính được. Chừng nào tồn tại một số nguyên không âm nhỏ nhất k sao cho $X_{k+1} = X_k$ thì thuật toán dừng.

Bước 3. Đặt $X^+ := X_k$. Kết luận X^+ là bao đóng cần tìm.

Dễ kiểm chứng được độ phức tạp thời gian của Thuật toán 4.1 là $O(|U||F|^2)$. Nếu tổ chức dữ liệu tốt, độ phức tạp thời gian của Thuật toán 4.1 có thể hạ xuống chỉ là $O(|U||F|)$.

Ví dụ 4.6. Xét LDQH $S = (U, F)$ với tập thuộc tính $U = \{a, b, c\}$ và tập các PTH trên U là $F = \{a \rightarrow b, b \rightarrow c\}$. Xét $X = \{a\}$, ta có dãy tập thuộc tính bao hàm X là

$$X_0 = \{a\},$$

$$X_1 = \{a, b\},$$

$$X_2 = \{a, b, c\},$$

$$X_3 = X_2.$$

$$\text{Do đó } \{a\}^+ = U.$$

$$\text{Thực hiện tương tự ta có } \{c\}^+ = \{c\}, \{b, c\}^+ = \{b, c\}.$$

Định lý 4.9. Thuật toán 4.1 tính đúng bao đóng X^+ , nghĩa là tồn tại số nguyên không âm nhỏ nhất k sao cho $X_k = X_{k+1} = X_{k+2} = \dots$ và $X^+ = X_k$.

Chứng minh. Vì U hữu hạn và $X_0 \subseteq X_1 \subseteq \dots \subseteq U$ nên sau một số hữu hạn bước, rõ ràng thuật toán phải tồn tại một số nguyên không âm nhỏ nhất k sao cho $X_k = X_{k+1} = X_{k+2} = \dots$

Bây giờ ta chỉ còn chứng minh $X^+ = X_k$. Giả sử $U = \{a_1, a_2, \dots, a_n\}$ và $a \in X^+$. Từ X_k , ta xét quan hệ $R = \{t_1, t_2\} \in \text{Rel}(U)$ sau đây:

$$t_1 = (0, 0, \dots, 0)$$

và

$$t_2(a_i) = \begin{cases} 0 & \text{nếu } a_i \in X_k \\ 1 & \text{ngược lại} \end{cases}$$

với mọi $i = 1, 2, \dots, n$.

Với quan hệ R như vậy, nếu tồn tại $Z \rightarrow W \in F$ sao cho $R(\neg Z \rightarrow W)$ thì $Z \subseteq X_k$ và $W \not\subseteq X_k$. Do đó, theo thuật toán suy ra $X_{k+1} \neq X_k$. Điều này mâu thuẫn điều kiện kết thúc của thuật toán. Vậy $R(F)$.

Từ $R(F)$ và $X \rightarrow a \in F^+$, suy ra $R(X \rightarrow a)$ và do đó $a \in X_k$. Nên $X^+ \subseteq X_k$. Ngoài ra, dễ thấy $X \rightarrow X_1 \in F^+, X_1 \rightarrow X_2 \in F^+, \dots, X_{k-1} \rightarrow X_k \in F^+$. Do đó, theo tính bắc cầu thì $X \rightarrow X_k \in F^+$ hay $X_k \subseteq X^+$. $\square$

Trên cơ sở tính chất (6) trong Mệnh đề 4.5 và Thuật toán 4.1, thuật toán giải bài toán thành viên được xây dựng như sau.

Thuật toán 4.2 (Thuật toán thành viên)

Vào: Tập PTH F trên tập thuộc tính U và PTH $X \rightarrow Y$.

Ra: Cho biết $X \rightarrow Y$ có thuộc F^+ hay không?

Phương pháp:

Bước 1. Tính bao đóng X^+ .

Bước 2. Nếu $Y \subseteq X^+$ thì kết luận $X \rightarrow Y \in F^+$. Ngược lại kết luận $X \rightarrow Y \notin F^+$.

Rõ ràng, độ phức tạp thời gian của Thuật toán 4.2 chính là độ phức tạp thời gian của Thuật toán 4.1, tức là $O(|U||F|^2)$. Cuối cùng trong mục này là thuật toán giải bài toán suy diễn phụ thuộc.

Thuật toán 4.3 (Thuật toán suy diễn phụ thuộc)

Vào: Quan hệ $R \in Rel(U)$.

Ra: Tập PTH F sao cho $F^+ = FD(R)$.

Phương pháp:

Bước 1. $F := \emptyset$.

Bước 2. (*Lặp*) Với mỗi tập con thuộc tính $X \subseteq U$, với mỗi thuộc tính $a \in U \setminus X$: nếu $R(X \rightarrow a)$ thì

$$F := F \cup \{X \rightarrow a\}.$$

Rõ ràng, tập F thu được bởi Thuật toán 4.3 chỉ gồm các PTH đúng trên quan hệ R . Vì các PTH $X \rightarrow Y \in F^+$ khi và chỉ khi $X \rightarrow a \in F^+$ với mọi $a \in Y$ và $Y \subseteq U$. Do đó, tập PTH F thu được thỏa mãn $F^+ = FD(R)$. Dễ dàng thấy độ phức tạp thời gian của Thuật toán 4.3 là hàm mũ theo số thuộc tính trong U . Do đó, khi số thuộc tính lớn thì thuật toán này không hiệu quả. Chính vì vậy, thuật toán này còn được hiểu là thuật toán “ngây thơ”, với mục đích nhằm chứng minh sự tồn tại của tập $FD(R)$.

Chẳng hạn, xét quan hệ R ở Ví dụ 4.2 trên. Vận dụng Thuật toán 4.3 chúng ta thu được tập PTH $F = \{a \rightarrow b, a \rightarrow c, c \rightarrow b, ab \rightarrow c, ac \rightarrow b\}$. Dễ dàng kiểm chứng được $F^+ = FD(R)$.

Hệ quả 4.2. Với mỗi quan hệ $R \in Rel(U)$, luôn tìm được một LDQH $S = (U, F)$ sao cho R là quan hệ Armstrong của S .

4.2.6. Bao đóng của thuộc tính trên quan hệ và thuật toán

Trên LDQH nhiều bài toán có độ phức tạp hàm mũ trở lên, nhưng trên thể hiện của nó là các quan hệ thì lại có phức tạp chỉ là đa thức. Một trong các công cụ giúp các bài toán đó trở nên dễ hơn là khái niệm

bao đóng thuộc tính trên quan hệ. Khái niệm này được định nghĩa như sau. Xét tập thuộc tính U , quan hệ $R \in Rel(U)$ và tập con thuộc tính $X \subseteq U$. Bao đóng của X trên R , ký hiệu X_R^+ , là tập các thuộc tính $a \in U$ sao cho PTH $X \rightarrow a$ đúng trên R . Tức là

$$X_R^+ = \{a \in U : R(X \rightarrow a)\}.$$

Lưu ý rằng các tính chất của bao đóng X^+ vẫn còn đúng đối với bao đóng X_R^+ .

Ví dụ 4.7. Xét quan hệ $R \in Rel(U)$, với $U = \{a, b, c\}$, như sau:

$$R = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline 2 & 0 & 0 \\ \hline \end{array}$$

$$\text{Ta có } \{c\}_R^+ = \{b, c\}, \{b\}_R^+ = \{b\}, \{b, c\}_R^+ = \{b, c\}, \{a\}_R^+ = U.$$

Bao đóng của tập thuộc tính trên quan hệ được biểu diễn qua hệ bảng nhau [28] (1987) như sau.

Định lý 4.10. Cho tập thuộc tính U , quan hệ $R \in Rel(U)$ và $X \subseteq U$. Khi đó

$$X_R^+ = \begin{cases} \bigcap_{X \subseteq E_{ij}} E_{ij} & \text{nếu } \exists E_{ij} \in E(R) : X \subseteq E_{ij} \\ U & \text{ngược lại.} \end{cases}$$

Chứng minh. Xét họ $E(X)$ và tập E tương ứng như sau:

$$E(X) = \{E_{ij} \in E(R) : X \subseteq E_{ij}\},$$

$$E = \bigcap E(X).$$

Giả sử X là một tập thuộc tính sao cho $E(X) = \emptyset$. Suy ra $t_i(X) \neq t_j(X)$ với mọi $t_i, t_j \in R$. Do đó, theo định nghĩa PTH ta có $R(X \rightarrow U)$,

hay bao đóng $X_R^+ = U$. Trường hợp $X = \emptyset$ thì định lý là hiển nhiên theo định nghĩa X_R^+ và $E(R)$, tức là $X_R^+ = E$. Cuối cùng, ta xét $X \neq \emptyset$ và $E(X) \neq \emptyset$. Dễ thấy ngay $X \subseteq E$. Trường hợp nếu $E(X) = E(R)$, thì ta có ngay $R(X \rightarrow E)$. Còn nếu $E(X) \neq E(R)$ thì với mọi $E_{ij} \in E(X)$ ta có

$$t_i(X) = t_j(X) \Rightarrow t_i(E) = t_j(E)$$

và với mọi $E_{ij} \notin E(X)$ thì phải tồn tại một thuộc tính $a \in X$ sao cho $t_i(a) \neq t_j(a)$. Điều này có nghĩa $R(X \rightarrow E)$. Tóm lại, trong cả hai trường hợp chúng ta đều có $R(X \rightarrow E)$, và do đó theo định nghĩa bao đóng thì $E \subseteq X_R^+$. Để ý, $R \in \text{Rel}(U)$ nên $E \subset U$. Vì $X \subseteq E \subseteq X_R^+$, suy ra $R(E \rightarrow X_R^+)$. Bây giờ chúng ta chỉ còn chứng minh thêm nếu có một thuộc tính $a \notin E$ thì $R(\neg E \rightarrow E \cup \{a\})$. Thật vậy, nếu $a \notin E$ thì suy ra phải tồn tại một $E_{ij} \in E(X)$ sao cho $a \notin E_{ij}$. Lúc đó, tồn tại một cặp $t_i, t_j \in R$ sao cho $t_i(E) = t_j(E)$ nhưng $t_i(a) \neq t_j(a)$. Do đó $t_i(E \cup \{a\}) \neq t_j(E \cup \{a\})$. Như vậy, theo định nghĩa bao đóng cuối cùng ta có $X_R^+ = E$. $\square$

Thuật toán 4.4 (Thuật toán tìm bao đóng thuộc tính trên quan hệ)

Vào: Tập thuộc tính U , quan hệ $R = \{t_1, t_2, \dots, t_m\} \in \text{Rel}(U)$ và $X \subseteq U$.

Ra: Bao đóng X_R^+ .

Phương pháp:

Bước 1. Từ R xây dựng $E(R)$.

Bước 2. Từ hệ $E(R)$ tính X_R^+ theo Định lý 4.10:

$$X_R^+ = \begin{cases} \bigcap_{X \subseteq E_{ij}} E_{ij} & \text{nếu } \exists E_{ij} \in E(R) : X \subseteq E_{ij} \\ U & \text{ngược lại.} \end{cases}$$

Bước 3. Kết luận X_R^+ là bao đóng cần tìm.

Tính đúng của Thuật toán 4.4 suy ra từ Định lý 4.10. Dễ thấy độ phức tạp thời gian của Thuật toán 4.4 là đa thức theo kích thước của quan hệ R .

Ví dụ 4.8. Xét quan hệ $R \in \text{Rel}(U)$, với $U = \{a, b, c\}$, như sau:

$$R = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline 2 & 0 & 0 \\ \hline \end{array}$$

Ta có $E_{12} = \{b, c\}$, $E_{13} = \{b\}$, $E_{23} = \{b\}$. Suy ra $E(R) = \{\{b\}, \{b, c\}\}$.

Khi đó

$$\begin{aligned} \{c\}_R^+ &= \bigcap \{\{b, c\}\} = \{b, c\} \\ \{b\}_R^+ &= \bigcap \{\{b\}, \{b, c\}\} = \{b\} \\ \{a\}_R^+ &= U. \end{aligned}$$

4.3. Phủ của phụ thuộc hàm

Mục này trình bày các phương pháp biểu diễn tập PTH tương đương tốt hơn, hay còn nói “gọn hơn” theo nghĩa số PTH ít hơn hoặc số các thuộc tính tham gia vào trong tập PTH ít hơn. Các tập PTH tương đương như vậy được gọi là phủ. Chẳng hạn, với tập PTH $F = \{a \rightarrow b, b \rightarrow c, a \rightarrow c, ab \rightarrow c, a \rightarrow bc\}$, chúng ta có thể thay thế bằng tập PTH $G = \{a \rightarrow b, b \rightarrow c\}$, bởi tất cả các PTH của F và những PTH có thể suy diễn từ F đều có thể suy diễn được từ các PTH của G . Việc tìm các biểu diễn tốt hơn của F sẽ giúp cho quá trình tính toán trên F nhanh hơn. Ngoài ra, về mặt không gian lưu trữ các PTH cũng ít hơn.

4.3.1. Định nghĩa

Hai tập PTH F và G trên tập thuộc tính U được gọi là *tương đương*, ký hiệu $F \equiv G$, nếu $F^+ = G^+$. Trường hợp ngược lại, $F^+ \neq G^+$, ta nói F *không tương đương* với G và ký hiệu $F \not\equiv G$. Khi $F \equiv G$ người ta nói G là một *phủ* của F .

Rõ ràng nếu G là một phủ của F thì F cũng là một phủ của G . Như đã đề cập việc gọi G là một phủ của F người ta ngụ ý rằng G là

tốt hơn F theo một nghĩa nào đó. Để tìm hiểu về sự tương đương của hai tập PTH, người ta mở rộng khái niệm suy diễn như sau.

Tập PTH F được gọi là *suy diễn* tập PTH G , ký hiệu $F \models G$, nếu với mọi PTH $X \rightarrow Y \in G$ thì $F \models X \rightarrow Y$. Trường hợp ngược lại, nếu tồn tại một PTH $X \rightarrow Y \in G$ sao cho $F \not\models X \rightarrow Y$ thì ta nói F *không suy diễn* G và ký hiệu $F \not\models G$.

Dễ thấy $F \models G$ nếu và chỉ nếu $G \subseteq F^+$.

Mệnh đề 4.11. $F \equiv G$ khi và chỉ khi $F \models G$ và $G \models F$.

Chứng minh. Theo tính đơn điệu và tính lũy đẳng của bao đóng PTH ta có:

$$\begin{aligned} F \models G &\Leftrightarrow G \subseteq F^+ \\ &\Leftrightarrow G^+ \subseteq F^+. \end{aligned}$$

Như vậy, $F \models G$ và $G \models F$ nếu và chỉ nếu $G^+ \subseteq F^+$ và $F^+ \subseteq G^+$, hay $F \equiv G$. $\square$

Trên cơ sở Mệnh đề 4.11, thuật toán kiểm tra sự tương đương của hai tập PTH được xây dựng như sau.

Thuật toán 4.5 (Thuật toán kiểm tra tính tương đương của PTH)

Vào: Hai tập PTH F và G trên tập thuộc tính U .

Ra: Cho biết $F \equiv G$ hay không?

Phương pháp:

Bước 1. Với mọi $X \rightarrow Y \in G$ sao cho $X \rightarrow Y \in F^+$ thì $F \models G$. Ngược lại nếu tồn tại $X \rightarrow Y \in G$ sao cho $X \rightarrow Y \notin F^+$ thì $F \not\models G$.

Bước 2. Nếu $F \models G$ và $G \models F$ thì kết luận $F \equiv G$. Ngược lại kết luận $F \not\equiv G$.

Dễ kiểm chứng được độ phức tạp thời gian của thuật toán trên là đa thức theo $|U|, |F|$ và $|G|$.

Ví dụ 4.9. Xét tập thuộc tính $U = \{a, b, c\}$ và các tập PTH $F = \{a \rightarrow b, b \rightarrow c, a \rightarrow c\}$, $G = \{a \rightarrow b, b \rightarrow c\}$, $H = \{a \rightarrow b, a \rightarrow c\}$ trên U .

Ta có $F \models G$ và $G \models F$. Do đó $F \equiv G$. Tuy nhiên $G \models H$ nhưng $H \not\models G$, suy ra $G \not\equiv H$.

4.3.2. Phủ không dư

Với một tập PTH F có thể có nhiều PTH dư trong F , tức chúng ta có thể suy diễn được các PTH này thông qua các PTH còn lại của F . Khái niệm này được định nghĩa hình thức như sau. Tập PTH F được gọi *không dư* nếu không tồn tại tập con thật sự F' của F sao cho $F' \equiv F$. Trường hợp nếu tồn tại một tập F' như thế thì F được gọi là *dư*. Tập PTH G được gọi là *phủ không dư* của F nếu G là một phủ của F và G không dư.

Ví dụ 4.10. Xét tập thuộc tính $U = \{a, b, c\}$ và các tập PTH $F = \{a \rightarrow b, b \rightarrow c, a \rightarrow c\}$, $G = \{a \rightarrow b, b \rightarrow c\}$ trên U .

Ta có tập PTH F là dư vì $G \equiv F$ và $G \subset F$. Tuy nhiên, tập PTH G không dư và do đó G là một phủ không dư của F .

Mệnh đề 4.12. Tập PTH F là dư khi và chỉ khi tồn tại một PTH $X \rightarrow Y \in F$ sao cho $F \setminus \{X \rightarrow Y\} \models X \rightarrow Y$.

Chứng minh. Giả sử F dư. Suy ra, tồn tại tập PTH F' sao cho $F' \subset F$ và $F' \equiv F$. Không mất tính tổng quát gọi $F' = F \setminus \{X \rightarrow Y\}$. Khi đó $(F')^+ = F^+$, và do đó $F' \models X \rightarrow Y$.

Ngược lại, nếu tồn tại một PTH $X \rightarrow Y \in F$ sao cho $F \setminus \{X \rightarrow Y\} \models X \rightarrow Y$, thì đặt $F' = F \setminus \{X \rightarrow Y\}$. Rõ ràng $F' \subset F$ và $(F')^+ = F^+$. Do đó, F là dư. $\square$

Như vậy, tập PTH F là không dư nếu và chỉ nếu

$$\nexists X \rightarrow Y \in F : F \setminus \{X \rightarrow Y\} \models X \rightarrow Y.$$

PTH $X \rightarrow Y$ thỏa mãn Mệnh đề 4.12, tức là loại nó đi và các PTH còn lại của F vẫn duy diễn được nó, được gọi là *PTH dư* trong F .

Vì

$$\begin{aligned}
F \setminus \{X \rightarrow Y\} \models X \rightarrow Y &\Leftrightarrow F \setminus \{X \rightarrow Y\} \models_A X \rightarrow Y \\
&\Leftrightarrow X \rightarrow Y \in (F \setminus \{X \rightarrow Y\})^+ \\
&\Leftrightarrow Y \subseteq X_{F \setminus \{X \rightarrow Y\}}^+.
\end{aligned}$$

Do đó, định nghĩa PTH dư ở trên còn có thể phát biểu tương đương lại như sau: PTH $X \rightarrow Y \in F$ được gọi là dư trong F nếu và chỉ nếu $Y \subseteq X_{F \setminus \{X \rightarrow Y\}}^+$.

Chẳng hạn, xét Ví dụ 4.10, ta thấy với PTH $a \rightarrow c \in F$ thì $F \setminus \{a \rightarrow c\} \models a \rightarrow c$ nên tập PTH F là dư. Tuy nhiên, không tồn tại PTH dư nào trong G nên tập PTH G là không dư.

Rõ ràng, với một tập PTH F luôn tồn tại một tập con $G \subseteq F$ sao cho G là một phủ không dư của F , đó là nếu F không dư thì $G = F$, còn nếu F dư thì suy ra tồn tại một PTH $X \rightarrow Y \in F$ sao cho $X \rightarrow Y$ là dư, khi đó đặt $F_1 = F \setminus \{X \rightarrow Y\}$. Chú ý, $F_1 \equiv F$. Lặp lại quá trình kiểm tra như trên đối với F_1 . Kết quả, ta thu được một phủ không dư của F . Cách làm như thế này là cơ sở cho thuật toán sau.

Thuật toán 4.6 (Thuật toán tìm một phủ không dư)

Vào: Tập PTH F trên tập thuộc tính U .

Ra: Một phủ không dư G của F .

Phương pháp:

Bước 1. $G := F$.

Bước 2. Với mỗi PTH $X \rightarrow Y \in F$, nếu $G \setminus \{X \rightarrow Y\} \models X \rightarrow Y$ thì $G := G \setminus \{X \rightarrow Y\}$.

Bước 3. Kết luận G là một phủ không dư cần tìm.

Có thể kiểm chứng được độ phức tạp thời gian của Thuật toán 4.6 là $O(|U||F|^3)$.

Ví dụ 4.11. Xét tập thuộc tính $U = \{a, b, c\}$ và tập PTH $F = \{a \rightarrow b, b \rightarrow a, b \rightarrow c, a \rightarrow c\}$ trên U . Ta có một phủ không dư của F là $G_1 = \{a \rightarrow b, b \rightarrow a, a \rightarrow c\}$. Ngoài ra, nếu tập F biểu diễn theo thứ

tự $\{a \rightarrow b, a \rightarrow c, b \rightarrow a, b \rightarrow c\}$ thì một phủ không dư khác của F là $G_2 = \{a \rightarrow b, b \rightarrow a, b \rightarrow c\}$.

Như vậy, phủ không dư là không duy nhất. Ngoài ra phủ không dư của một tập PTH F không nhất thiết chứa trong F . Chẳng hạn, $G_3 = \{a \rightarrow b, b \rightarrow a, ab \rightarrow c\}$ là một phủ không dư của tập PTH F ở ví dụ trên, tuy nhiên $G_3 \not\subseteq F$.

4.3.3. *Phủ rút gọn*

Với một tập PTH F không dư thì việc loại bỏ bất kỳ một PTH nào của F đều cho ra một tập PTH không tương đương với F . Tuy vậy, chúng ta còn có thể rút gọn được kích thước của F , đó là loại bỏ một số thuộc tính trong các PTH của F mà vẫn cho ra một tập PTH tương đương với F .

Xét PTH $X \rightarrow Y \in F$. Thuộc tính $a \in U$ được gọi là *dư* trong $X \rightarrow Y$ (ứng với F) nếu xảy ra một trong hai điều kiện sau:

- i) $X = Z \cup \{a\}, X \neq Z$ và $F \setminus \{X \rightarrow Y\} \cup \{Z \rightarrow Y\} \equiv F$.
- ii) $Y = W \cup \{a\}, Y \neq W$ và $F \setminus \{X \rightarrow Y\} \cup \{X \rightarrow W\} \equiv F$.

Trong trường hợp (i) thuộc tính a được gọi là *dư trái*, trong trường hợp (ii) thuộc tính a được gọi là *dư phải*.

PTH $X \rightarrow Y$ được gọi là *rút gọn trái* (tương ứng *rút gọn phải*) nếu X (tương ứng Y) không chứa thuộc tính dư trái (tương ứng dư phải). Trường hợp nếu PTH $X \rightarrow Y$ rút gọn trái, rút gọn phải và $Y \neq \emptyset$ thì $X \rightarrow Y$ được gọi là *rút gọn*.

Tập PTH F được gọi là *rút gọn trái* (tương ứng *rút gọn phải* hay *rút gọn*) nếu mọi PTH trong F là rút gọn trái (tương ứng rút gọn phải hay rút gọn). Tập PTH G được gọi là *phủ rút gọn trái* (tương ứng *phủ rút gọn phải* hay *phủ rút gọn*) của tập PTH F nếu G là một phủ của F và G rút gọn trái (tương ứng rút gọn phải hay rút gọn).

Ví dụ 4.12. Xét tập PTH $F = \{a \rightarrow bc, b \rightarrow c, ab \rightarrow d\}$ trên tập thuộc tính $U = \{a, b, c, d\}$. Ta có thuộc tính c là dư phải trong PTH $a \rightarrow bc$

và thuộc tính b là dư trái trong PTH $ab \rightarrow d$. Suy ra, F không rút gọn trái, không rút gọn phải và do đó không rút gọn.

Khi đó, dễ thấy tập PTH $G_1 = \{a \rightarrow bc, b \rightarrow c, a \rightarrow d\}$ là rút gọn trái nhưng không rút gọn phải. Tập PTH $G_2 = \{a \rightarrow b, b \rightarrow c, ab \rightarrow d\}$ là rút gọn phải nhưng không rút gọn trái. Tập PTH $G_3 = \{a \rightarrow b, b \rightarrow c, a \rightarrow d\}$ rút gọn trái, rút gọn phải và do đó là rút gọn. Ngoài ra, G_1, G_2 và G_3 là các phủ của F . Do đó, G_1 là một phủ rút gọn trái của F , G_2 là một phủ rút gọn phải của F và G_3 là một phủ rút gọn của F .

Bây giờ chúng ta nghiên cứu thuật toán tìm một phủ rút gọn. Đầu tiên ta có lưu ý sau.

Nhận xét 4.1.

1) Tương tự như vậy, trong quá trình rút gọn thuộc tính dư phải a ở PTH $X \rightarrow Y \in F$, ta thấy với $G := F \setminus \{X \rightarrow Y\} \cup \{X \rightarrow Y \setminus \{a\}\}$ thì

$$\begin{aligned} G \equiv F &\Leftrightarrow G \models F \\ &\Leftrightarrow G \models X \rightarrow Y \\ &\Leftrightarrow G \models X \rightarrow a \\ &\Leftrightarrow X \rightarrow a \in G^+ \\ &\Leftrightarrow a \in X_G^+. \end{aligned}$$

Trong trường hợp này thuộc tính a là dư phải trong PTH $X \rightarrow Y$ nếu và chỉ nếu $X \rightarrow a \in G^+$ hay $a \in X_G^+$.

1) Trong quá trình rút gọn thuộc tính dư trái a ở PTH $X \rightarrow Y \in F$ ta thấy với $G := F \setminus \{X \rightarrow Y\} \cup \{X \setminus \{a\} \rightarrow Y\}$ thì

$$\begin{aligned} G \equiv F &\Leftrightarrow F \models X \setminus \{a\} \rightarrow Y \\ &\Leftrightarrow X \setminus \{a\} \rightarrow Y \in F^+ \\ &\Leftrightarrow Y \subseteq (X \setminus \{a\})^+. \end{aligned}$$

Do đó, trong trường hợp này thuộc tính a là dư trái trong PTH

$X \rightarrow Y$ nếu và chỉ nếu $X \setminus \{a\} \rightarrow Y \in F^+$, hay $Y \subseteq (X \setminus \{a\})^+$.

Để ý giả sử $X \setminus \{a\} \rightarrow a \in F^+$ với $a \in X$. Khi đó nếu $X \rightarrow Y \in F$ thì rõ ràng ta có $X \setminus \{a\} \rightarrow Y \in F^+$. Tức là thuộc tính a là dư trái trong PTH $X \rightarrow Y \in F$.

3) PTH $X \rightarrow Y \in F$ là dư khi và chỉ khi mọi thuộc tính trong Y là dư phải. Khi đó, loại bỏ Y sẽ cho ra PTH $X \rightarrow \emptyset$.

Nhận xét 4.1 là cơ sở cho thuật toán tìm một phủ rút gọn trái, thuật toán tìm một phủ rút gọn phải và thuật toán tìm một phủ rút gọn sau đây

Thuật toán 4.7 (Thuật toán tìm một phủ rút gọn trái)

Vào: Tập PTH F trên tập thuộc tính U .

Ra: Một phủ rút gọn trái G của F .

Phương pháp:

Bước 1. $G := F$.

Bước 2. Với mỗi PTH $X \rightarrow Y \in F$ và với mỗi $a \in X$, nếu $Y \subseteq (X \setminus \{a\})^+$ thì loại bỏ $a \in X$ trong $X \rightarrow Y \in G$.

Bước 3. Kết luận G là một phủ rút gọn trái cần tìm.

Theo Lưu ý 4.1, Thuật toán 4.7 cho ra một phủ rút gọn trái G của tập PTH F . Ngoài ra, dễ kiểm chứng được độ phức tạp thời gian của Thuật toán 4.7 là $O(|U|^2|F|^3)$.

Thuật toán sau cũng cho ra một phủ rút gọn phải G của tập PTH F với độ phức tạp thời gian của thuật toán là $O(|U|^2|F|^3)$.

Thuật toán 4.8 (Thuật toán tìm một phủ rút gọn phải)

Vào: Tập PTH F trên tập thuộc tính U .

Ra: Một phủ rút gọn phải G của F .

Phương pháp:

Bước 1. $G := F$.

Bước 2. Với mỗi PTH $X \rightarrow Y \in F$ và với mỗi $a \in Y$, nếu $a \in X_{G \setminus \{X \rightarrow Y\} \cup \{X \rightarrow Y \setminus \{a\}\}}^+$ thì loại bỏ $a \in Y$ trong $X \rightarrow Y \in G$.

Bước 3. Kết luận G là một phủ rút gọn phải cần tìm.

Lúc này, chúng ta có ngay thuật toán tìm một phủ rút gọn với độ

phức tạp thời gian là $O(|U|^2|F|^3)$.

Thuật toán 4.9 (Thuật toán tìm một phủ rút gọn)

Vào: Tập PTH F trên tập thuộc tính U .

Ra: Một phủ rút gọn G của F .

Phương pháp:

Bước 1. Tìm phủ rút gọn trái của F bằng Thuật toán 4.7. Giả sử phủ đó là G_1 .

Bước 2. Tìm phủ rút gọn phải của G_1 bằng Thuật toán 4.8. Giả sử phủ đó là G_2 .

Bước 3. Với mỗi PTH $X \rightarrow Y \in G_2$, nếu $Y = \emptyset$ thì $G_2 := G_2 \setminus \{X \rightarrow Y\}$.

Bước 4. Đặt $G := G_2$. Kết luận G là một phủ rút gọn cần tìm.

Ví dụ 4.13. Xét tập PTH $F = \{a \rightarrow c, ab \rightarrow de, ab \rightarrow cdi, ac \rightarrow j\}$ trên tập thuộc tính $U = \{a, b, c, d, e, i, j\}$.

Ta có thuộc tính c là dư trái trong PTH $ac \rightarrow j$, các thuộc tính d và e tương ứng là dư phải trong PTH $ab \rightarrow de$ và PTH $ab \rightarrow cdi$. Suy ra một phủ rút gọn của F là

$$G = \{a \rightarrow c, ab \rightarrow e, ab \rightarrow di, a \rightarrow j\}.$$

Lưu ý 4.1. Thứ tự rút gọn tập PTH là quan trọng, đó là rút gọn phải trước sau đó đến rút gọn trái có thể sẽ dẫn đến một tập PTH không rút gọn (không rút gọn phải). Chẳng hạn, với tập PTH $F = \{a \rightarrow b, c \rightarrow b, ab \rightarrow c\}$. Rõ ràng, F là rút gọn phải. Tuy nhiên, bây giờ nếu ta rút gọn trái F thì sẽ cho ra tập PTH $G = \{a \rightarrow b, c \rightarrow b, a \rightarrow c\}$. Lúc này dễ thấy thuộc tính b là dư phải trong PTH $a \rightarrow b \in G$, suy ra không phải rút gọn phải và do đó không phải rút gọn. Như vậy, thứ tự rút gọn tập PTH là rút gọn trái trước khi rút gọn phải.

Phủ không dư có thể không rút gọn và ngược lại phủ rút gọn có thể là phủ dư. Chẳng hạn, tập PTH F ở Ví dụ 4.12 là không dư nhưng không rút gọn. Còn chẳng hạn, với tập PTH $F = \{a \rightarrow be, a \rightarrow c, be \rightarrow c\}$, rõ ràng F là rút gọn nhưng dư.

4.3.4. Phủ tối thiểu

Mục này giới thiệu một phủ tốt hơn nữa thỏa hai đặc tính là không dư và rút gọn. Phủ như thế này được gọi là phủ tối thiểu, phủ này đóng vai trò rất quan trọng trong quá trình chuẩn hóa. Đầu tiên chúng ta có kết quả sau, kết quả này suy ra ngay từ tính thu hẹp phải và tính cộng tính phải.

Mệnh đề 4.13. *Bất kỳ tập PTH F nào đều tồn tại một tập PTH G tương đương với F , sao cho vế phải của các PTH trong G có đúng một thuộc tính.*

Các PTH rút gọn trái trong F có dạng $X \rightarrow a$ còn được gọi là PTH tối thiểu. Với kết quả trên, phủ tối thiểu được định nghĩa như sau. Tập PTH F được gọi là tối thiểu nếu nó thỏa các điều kiện:

- i) Vế phải của mọi PTH trong F chỉ gồm đúng một thuộc tính.
- ii) F rút gọn trái.
- iii) F không dư.

Tập PTH G được gọi là một phủ tối thiểu¹ của F nếu G là một phủ của F và G tối thiểu.

Như vậy, phủ tối thiểu là một phủ không dư và cũng là một phủ rút gọn.

Mệnh đề 4.14. *Mỗi tập PTH F đều có ít nhất một phủ tối thiểu.*

Chứng minh. Theo Mệnh đề 4.13, tồn tại tập PTH G tương đương F thỏa điều kiện (i). Sau đó loại bỏ các thuộc tính dư trái nếu có trong G để thu được tập G thỏa (ii). Cuối cùng loại bỏ các PTH dư trong G nếu có để nhận được G thỏa (iii). Tập PTH G sau cùng rõ ràng là một phủ tối thiểu cần tìm của F . $\square$

Chứng minh của Mệnh đề 4.14 cũng là cơ sở để chúng ta có ngay thuật toán tìm một phủ tối thiểu như sau.

¹Một số tác giả còn gọi là tập bất khả qui của F .

Thuật toán 4.10 (Thuật toán tìm một phủ tối thiểu)

Vào: Tập PTH F trên tập thuộc tính U .

Ra: Một phủ tối thiểu G của F .

Phương pháp:

Bước 1. $G := \emptyset$.

Bước 2. Với mỗi PTH $X \rightarrow Y \in F$, và với mỗi thuộc tính $a \in Y \setminus X$, nếu $X \rightarrow a \notin G$ thì $G := G \cup \{X \rightarrow a\}$.

Bước 3. Tìm phủ rút gọn trái của G bằng Thuật toán 4.7. Giả sử phủ đó là G_1 .

Bước 4. Tìm một phủ không dư của G_1 bằng Thuật toán 4.6. Giả sử phủ đó là G_2 .

Bước 5. Đặt $G := G_2$. Kết luận G là một phủ tối thiểu cần tìm.

Dễ kiểm chứng được độ phức tạp thời gian của Thuật toán 4.10 là đa thức theo $|U|$ và $|F|$. Ngoài ra, rõ ràng nếu chúng ta bắt đầu với một phủ không dư thì quá trình loại bỏ các thuộc tính dư trái có thể sẽ cho ra một tập PTH dư. Do đó, thứ tự thực hiện của hai bước rút gọn trái và tìm một phủ không dư trong Thuật toán 4.10 là không được thay đổi.

Ví dụ 4.14. Xét tập PTH $F = \{a \rightarrow bce, ab \rightarrow de, bi \rightarrow j\}$ trên tập thuộc tính $U = \{a, b, c, d, e, i, j\}$.

Vận dụng Thuật toán 4.10, quá trình tìm một phủ tối thiểu được thực hiện qua từng bước như sau:

- Tách: tách các PTH trong F thành các PTH có đúng một thuộc tính ở vế phải:

$$G_1 = \{a \rightarrow b, a \rightarrow c, a \rightarrow e, ab \rightarrow d, ab \rightarrow e, bi \rightarrow j\}.$$

- Rút gọn trái: thuộc tính b là dư trái trong hai PTH $ab \rightarrow d$ và $ab \rightarrow e$ của G_1 , rút gọn ta được:

$$G_2 = \{a \rightarrow b, a \rightarrow c, a \rightarrow e, a \rightarrow d, bi \rightarrow j\}.$$

- Loại bỏ PTH dư: không có PTH dư trong G_2 .

Suy ra, một phủ tối tiểu của F là $G_2 = \{a \rightarrow b, a \rightarrow c, a \rightarrow e, a \rightarrow d, bi \rightarrow j\}$.

Lưu ý 4.2. Phủ tối tiểu là không duy nhất. Chẳng hạn với tập PTH $F = \{a \rightarrow b, b \rightarrow a, a \rightarrow c, c \rightarrow a, b \rightarrow c\}$ chúng ta có hai phủ tối tiểu là

$$G_1 = \{a \rightarrow b, c \rightarrow a, b \rightarrow c\} \text{ và } G_2 = \{a \rightarrow b, b \rightarrow a, a \rightarrow c, c \rightarrow a\}.$$

Ví dụ này cũng cho thấy, phủ rút gọn cũng là không duy nhất.

4.4. Khóa của lược đồ quan hệ

Khóa của một LĐQH $S = (U, F)$ là một tập các thuộc tính K trong U sao cho K xác định hàm được tất cả các thuộc tính U . Nghĩa là, nếu chúng ta biết được thông tin (giá trị) về K thì sẽ biết được thông tin về U một cách duy nhất. Rõ ràng, điều này cũng đã nói lên được ý nghĩa của khóa K trong LĐQH.

4.4.1. Định nghĩa

Xét LĐQH $S = (U, F)$. Một *khóa* của S là một tập con $K \subseteq U$ sao cho $K \rightarrow U \in F^+$. Trường hợp nếu không tồn tại $K' \subset K$ sao cho $K' \rightarrow U \in F^+$ thì K được gọi là *khóa tối tiểu* (hay *cực tiểu*) của S .

Như vậy, dễ thấy U là một khóa của LĐQH S và bất kỳ LĐQH nào cũng có ít nhất một khóa tối tiểu. Ký hiệu $Key(S)$ là tập tất cả các khóa tối tiểu của LĐQH S . Bài toán tìm $Key(S)$ là bài toán có độ phức tạp hàm mũ theo số thuộc tính của U . Do đó, trong trường hợp CSDL có rất nhiều thuộc tính thì đây là một bài toán cực kỳ khó. Trong thực tế, khi một LĐQH có nhiều hơn một khóa tối tiểu, thì người ta thường chỉ định một trong các khóa tối tiểu làm khóa chính. Khóa chính này có thể phục vụ như là khóa tập tin khi quan hệ trên LĐQH này được cài đặt. Dễ thấy $Key(S)$ là một hệ Sperner, tức là không có hai khóa tối tiểu bất kỳ nào của LĐQH S chứa nhau. Khi đó, ta có thể suy ra ngay đánh giá về số khóa tối tiểu của LĐQH S .

Hệ quả 4.3.

$$|Key(S)| \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

Thuộc tính $a \in U$ được gọi là *thuộc tính cơ bản* của LĐQH S nếu a thuộc một khóa tối tiểu nào đó của S . Trong trường hợp ngược lại, a được gọi là *thuộc tính không cơ bản*. Ký hiệu U_N là tập tất cả thuộc tính không cơ bản. Rõ ràng $U_N \in Closed(S)$ và $U_N = U \setminus \bigcup Key(S)$. Các thuộc tính không cơ bản đóng một vai trò rất quan trọng trong quá trình chuẩn hóa. Bài toán xác định xem một thuộc tính a có phải là thuộc tính cơ bản của một LĐQH S hay không là bài toán có độ phức tạp NP-đầy đủ.

Ví dụ 4.15. Cho LĐQH $S = (U, F)$ với tập thuộc tính $U = \{c, s, z\}$ (ở đây ý nghĩa của các thuộc tính c, s, z tương ứng là thành phố, địa chỉ đường và mã bưu điện) và tập PTH $F = \{cs \rightarrow z, z \rightarrow c\}$.

Ta có $cs \rightarrow U \in F^+$ và $sz \rightarrow U \in F^+$. Tuy nhiên, $c \rightarrow U \notin F^+$, $s \rightarrow U \notin F^+$ và $z \rightarrow U \notin F^+$. Suy ra $\{c, s\}$ và $\{s, z\}$ là các khóa tối tiểu của S . Do đó, các thuộc tính c, s, z đều là các thuộc tính cơ bản và như vậy $U_N = \emptyset$.

4.4.2. Một số tính chất cơ bản

Từ đây về sau chúng ta luôn giả thiết các LĐQH $S = (U, F)$ với $U = \{a_1, a_2, \dots, a_n\}$ và

$$F = \{X_i \rightarrow Y_i | X_i, Y_i \subseteq U, X_i \cap Y_i = \emptyset, i = 1, 2, \dots, m\}.$$

$$\text{Ký hiệu } U_L = \bigcup_{i=1}^m X_i, U_R = \bigcup_{i=1}^m Y_i.$$

Lưu ý, ở trên chúng ta giả thiết vế trái và vế phải của các PTH là không có giao. Bởi phần giao nếu có, có thể bỏ đi ở vế phải và các PTH thu được vẫn tương đương PTH ban đầu. Các PTH như thế này còn được xem là PTH thu gọn tự nhiên. Chẳng hạn, PTH $abc \rightarrow bd$ rõ ràng tương đương với PTH thu gọn $abc \rightarrow d$.

Bổ đề 4.2. K là một khóa của LĐQH S khi và chỉ khi $K^+ = U$.

Chứng minh. Nếu K là một khóa của S thì $K \rightarrow U \in F^+$ và do đó $U \subseteq K^+$. Hơn nữa theo thuật toán tính bao đóng ta có $K^+ \subseteq U$. Vậy $K^+ = U$.

Ngược lại, nếu $K^+ = U$ thì $K \rightarrow U \in F^+$. Tức K là một khóa của S . $\square$

Bổ đề 4.3. Nếu $X \rightarrow Y \in F^+$ và $a \notin U_L$ thì $X \setminus \{a\} \rightarrow Y \setminus \{a\} \in F^+$.

Chứng minh. Vì $X \rightarrow Y \in F^+$ nên $Y \subseteq X^+$ và do đó $Y \setminus \{a\} \subseteq X^+ \setminus \{a\}$.

Theo thuật toán tính bao đóng:

- Nếu $a \notin X$ thì $(X \setminus \{a\})^+ = X^+$. Suy ra

$$Y \setminus \{a\} \subseteq X^+ \setminus \{a\} \subseteq X^+ = (X \setminus \{a\})^+.$$

Như vậy, $X \setminus \{a\} \rightarrow Y \setminus \{a\} \in F^+$.

- Còn nếu $a \in X$ và theo giả thiết $a \notin U_L$ thì suy ra $(X \setminus \{a\})^+ = X^+ \setminus \{a\}$ và do đó

$$Y \setminus \{a\} \subseteq X^+ \setminus \{a\} = (X \setminus \{a\})^+.$$

Cũng như vậy, $X \setminus \{a\} \rightarrow Y \setminus \{a\} \in F^+$.

Tóm lại, chúng ta luôn có $X \setminus \{a\} \rightarrow Y \setminus \{a\} \in F^+$. $\square$

Bổ đề 4.4. Nếu $a \in K$ và $K \setminus \{a\} \rightarrow a \in F^+$ thì K không phải là một khóa tối thiểu của LĐQH S .

Chứng minh. Theo tính phản xạ $K \setminus \{a\} \rightarrow K \setminus \{a\} \in F^+$ và giả thiết $K \setminus \{a\} \rightarrow a \in F^+$, vận dụng tính cộng tính phải suy ra $K \setminus \{a\} \rightarrow K \in F^+$. Vì $K \setminus \{a\} \subset K$ do đó nếu $K \in \text{Key}(S)$ thì $K \setminus \{a\} \rightarrow U \in F^+$. Điều này mâu thuẫn với tính chất tối thiểu của K . Vậy K không phải là một khóa tối thiểu của LĐQH S . $\square$

Định lý 4.15. Nếu K là một khóa tối tiểu của $LDQH S$ thì

$$(U \setminus U_R) \subseteq K \subseteq (U \setminus U_R) \cup (U_L \cap U_R).$$

Chứng minh. Theo Bổ đề 4.2 và thuật toán tính bao đóng, ta có

$$U = K^+ \subseteq K \cup U_R \subseteq U.$$

Suy ra $K \cup U_R = U$, và do đó $U \setminus U_R \subseteq K$.

Hơn nữa, ta có $K \subseteq U = (U \setminus U_R) \cup (U_L \cap U_R) \cup (U_R \setminus U_L)$. Do đó, phần chứng minh còn lại của định lý là quy về chứng minh $K \cap (U_R \setminus U_L) = \emptyset$. Thật vậy, giả sử tồn tại thuộc tính $a \in K, a \in U_R$ và $a \notin U_L$. Theo Bổ đề 4.3, từ $K \rightarrow U \in F^+$ và $a \notin U_L$, suy ra

$$K \setminus \{a\} \rightarrow U \setminus \{a\} \in F^+ \quad (4.1)$$

Vì $a \notin U_L$, nên $U_L \subseteq U \setminus \{a\}$, và do đó

$$U \setminus \{a\} \rightarrow U_L \in F^+ \quad (4.2)$$

Hơn nữa, ta có

$$U_L \rightarrow U_R \in F^+ \quad (4.3)$$

Cuối cùng vì $a \in U_R$, nên

$$U_R \rightarrow a \in F^+ \quad (4.4)$$

Vận dụng tính bắc cầu cho (4.1), (4.2), (4.3) và (4.4) ta thu được $K \setminus \{a\} \rightarrow a \in F^+$. Theo Bổ đề 4.4, suy ra K không phải khóa tối tiểu. Điều này là mâu thuẫn với giả thiết.

Như vậy, $K \cap (U_R \setminus U_L) = \emptyset$. □

Từ Định lý 4.15 ta thấy ngay $(U \setminus U_R) \cup (U_L \cap U_R)$ là một khóa, ngoài ra các khóa tối tiểu chỉ khác nhau trên tập giao $U_L \cap U_R$. Nói

cách khác, nếu K_1 và K_2 là hai khóa tối tiểu thì $K_1 \setminus K_2 \subset U_L \cap U_R$ và $K_2 \setminus K_1 \subset U_L \cap U_R$. Cũng từ Định lý 4.15, chúng ta thu được ngay một số kết quả nhận biết một LĐQH khi nào có một khóa tối tiểu duy nhất và cách xác định khóa tối tiểu đó.

Hệ quả 4.4.

- 1) Nếu $U_L \cap U_R = \emptyset$ thì $U \setminus U_R$ là khóa tối tiểu duy nhất của LĐQH S .
- 2) Nếu $(U \setminus U_R)^+ = U$ thì $U \setminus U_R$ là khóa tối tiểu duy nhất của LĐQH S .

Lưu ý 4.3. Định lý 4.15 vẫn còn đúng cho khoảng sau:

$$U \setminus U_R \subseteq K \subseteq (U \setminus U_R) \cup ((U_L \cap U_R) \setminus a(U_L, U_R))$$

với $a(U_L, U_R) = (U_L \cap U_R) \cap (U_L \setminus U_R)^+$. Lúc này ta cũng có thể viết K lại:

$$U \setminus U_R \subseteq K \subseteq (U \setminus U_R) \cup ((U_L \cap U_R) \setminus (U_L \setminus U_R)^+).$$

Chúng minh kết quả này được dành làm bài tập.

Mệnh đề 4.16. Nếu K là một khóa tối tiểu của LĐQH S thì $X^+ \cap K = X$ với mọi $X \subseteq K$.

Chúng minh. Giả sử tồn tại thuộc tính $a \in X^+$ và $a \in K \setminus X$. Suy ra, $X \rightarrow a \in F^+$ và $X \subseteq K \setminus \{a\}$, hay $K \setminus \{a\} \rightarrow X \in F^+$. Vận dụng tính bắc cầu, ta có $K \setminus \{a\} \rightarrow a \in F^+$. Theo Bổ đề 4.4, suy ra K không phải khóa tối tiểu. Điều này là mâu thuẫn với giả thiết. $\square$

Bây giờ xét U_I là tập giao của các khóa tối tiểu:

$$U_I = \bigcap Key(S).$$

Từ Định lý 4.15, chúng ta suy ra ngay $U \setminus U_R \subseteq U_I$.

Bổ đề 4.5. $U_I = U \setminus U_R$.

Chứng minh. Ta có $U_I \cap U_R = \emptyset$. Thật vậy, vì U_I là giao của các khóa tối tiểu nên bây giờ chúng ta chỉ cần chứng minh với mọi $a \in U_R$, tồn tại một khóa tối tiểu K sao cho $a \notin K$ là xong. Lấy bất kỳ $a \in U_R$. Suy ra, tồn tại một PTH $X_i \rightarrow Y_i \in F$ sao cho $a \in Y_i$ và $a \notin X_i$. Vận dụng tính gia tăng cho PTH $X_i \rightarrow Y_i \in F$, ta thu được $X_i \rightarrow X_i \cup Y_i \in F^+$ và do đó cũng thu được $X_i \cup (U \setminus (X_i \cup Y_i)) \rightarrow U \in F^+$. Điều này có nghĩa $X_i \cup (U \setminus (X_i \cup Y_i))$ là khóa và $a \notin X_i \cup (U \setminus (X_i \cup Y_i))$. Bởi vì khóa là chứa khóa tối tiểu. Do đó, tồn tại một khóa tối tiểu $K \subseteq X_i \cup (U \setminus (X_i \cup Y_i))$ sao cho $a \notin K$.

Như vậy, suy ra $U_I \subseteq U \setminus U_R$. Hơn nữa, theo nhận xét trên $U \setminus U_R \subseteq U_I$. Vậy $U_I = U \setminus U_R$. $\square$

Định lý 4.17. *LDQH S có một khóa tối tiểu duy nhất khi và chỉ khi là $U_I^+ = U$.*

Chứng minh. Rõ ràng, nếu S có một khóa tối tiểu duy nhất K , thì suy ra $U_I^+ = K^+ = U$.

Ngược lại, nếu $U_I^+ = U$ thì theo Hệ quả 4.4 và Mệnh đề 4.5 chúng ta có $U \setminus U_R$ là khóa tối tiểu duy nhất của S . $\square$

Định lý 4.18. *Giả sử K là một khóa của LDQH S . Khi đó K là khóa tối tiểu duy nhất của S khi và chỉ khi $K \cap U_R = \emptyset$.*

Chứng minh. Theo Bổ đề 4.5, nếu K là khóa tối tiểu duy nhất của S thì $U \setminus U_R = K$. Suy ra, $K \cap U_R = \emptyset$.

Ngược lại, giả sử $K \cap U_R = \emptyset$. Suy ra, $K \subseteq U \setminus U_R$. Hơn nữa, theo Bổ đề 4.15 ta có $U \setminus U_R \subseteq K$. Do đó, $K = U \setminus U_R$. Điều này có nghĩa K phải là khóa tối tiểu duy nhất. $\square$

4.4.3. Thuật toán tìm khóa

Trên cơ sở Định lý 4.15, chúng ta có thuật toán tìm một khóa tối tiểu của một LDQH với độ phức tạp thời gian đa thức như sau.

Thuật toán 4.11 (Thuật toán tìm một khóa tối tiểu)

Vào: LĐQH $S = (U, F)$.

Ra: Một khóa tối tiểu K .

Phương pháp:

Bước 1. Nếu $(U \setminus U_R)^+ = U$ thì kết luận $K := U \setminus U_R$ là khóa tối tiểu duy nhất cần tìm và thuật toán dừng. Còn không thì sang Bước 2.

Bước 2. Tính $U_L \cap U_R$. Giả sử $U_L \cap U_R = \{a_{t_1}, a_{t_2}, \dots, a_{t_l}\}$. Xây dựng dãy tập thuộc tính $K_0, K_1, \dots, K_l$ theo qui tắc sau:

$$K_0 := (U \setminus U_R) \cup (U_L \cap U_R)$$

$$K_i := \begin{cases} K_{i-1} \setminus \{a_{t_i}\} & \text{nếu } (K_{i-1} \setminus \{a_{t_i}\})^+ = U \\ K_{i-1} & \text{ngược lại} \end{cases}$$

với $i = 1, 2, \dots, l$.

Bước 3. Đặt $K := K_l$. Kết luận K là một khóa tối tiểu cần tìm.

Dễ thấy, độ phức tạp thời gian của Thuật toán 4.11 là đa thức theo $|U|$ và $|F|$. Chúng ta biết rằng khóa tối tiểu chứa trong khóa, do đó chúng ta có thể tìm một khóa tối tiểu bắt đầu từ một khóa, chẳng hạn khóa là tập $U = \{a_1, a_2, \dots, a_n\}$. Khi đó, Bước 2 của thuật toán trên cũng có thể viết lại như sau: xây dựng dãy tập thuộc tính $K_0, K_1, \dots, K_n$ theo qui tắc:

$$K_0 := U$$

$$K_i := \begin{cases} K_{i-1} \setminus \{a_i\} & \text{nếu } (K_{i-1} \setminus \{a_i\})^+ = U \\ K_{i-1} & \text{ngược lại} \end{cases}$$

với $i = 1, 2, \dots, n$. Tuy nhiên, dễ thấy phương pháp này có độ phức tạp thời gian kém hơn Thuật toán 4.11.

Ví dụ 4.16. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c, d\}$ và $F = \{ab \rightarrow d, c \rightarrow b\}$.

Ta có $U_R = \{b, d\}$. Suy ra $U \setminus U_R = \{a, c\}$ và $(U \setminus U_R)^+ = U$. Do đó, S có một khóa tối tiểu duy nhất là $\{a, c\}$.

Ví dụ 4.17. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c, d, e, g\}$ và $F =$

$\{b \rightarrow c, c \rightarrow b, a \rightarrow gd\}$.

Ta có $U_R = \{b, c, d, g\}$, nên $U \setminus U_R = \{a, e\}$. Và do đó $(U \setminus U_R)^+ = \{a, e, d, g\} \neq U$. Khi đó, tính $U_L \cap U_R = \{b, c\}$ và xây dựng dãy các tập thuộc tính sau:

$$K_0 = \{a, e, b, c\}$$

$$K_1 = \{a, e, c\}$$

$$K_2 = K_1.$$

Vậy, một khóa tối tiểu của S là $\{a, c, e\}$.

Có thể thấy, nếu chúng ta thay đổi thứ tự các thuộc tính trong tập $U_L \cap U_R$ thì sẽ tìm thấy thêm các khóa tối tiểu khác nếu có. Chẳng hạn trong ví dụ trên, nếu tập $U_L \cap U_R = \{c, b\}$ thì thực hiện tương tự chúng ta tìm thêm được một khóa tối tiểu nữa là $\{a, e, b\}$. Từ đây, rõ ràng nếu hoán vị tập $U_L \cap U_R$ chúng ta sẽ tìm được tất cả các khóa tối tiểu của một LĐQH $S = (U, F)$. Tuy nhiên, phương pháp như thế này có độ phức tạp thời gian rất lớn, giai thừa theo số thuộc tính trong U .

Sau đây là kết quả nhận biết một tập khác rỗng $K(S)$ gồm một số khóa tối tiểu cho trước có phải là tập $Key(S)$ hay không. Kết quả này là cơ sở để tìm tất cả các khóa tối tiểu của LĐQH S .

Định lý 4.19. *Giả sử $K(S)$ là một tập khác rỗng gồm một số khóa tối tiểu của LĐQH S . Điều kiện cần và đủ để tồn tại thêm một khóa tối tiểu $K \in Key(S) \setminus K(S)$ là*

$$(\exists X_i \rightarrow Y_i \in F)(\exists K_j \in K(S)) : (\forall K_p \in K(S))(X_i \cup (K_j \setminus Y_i) \not\subseteq K_p).$$

Chứng minh. Gọi K là một khóa tối tiểu mới của S chưa có trong $K(S)$. Từ K chúng ta xây dựng một tập K' như sau: $K' \supseteq K$ và K' là tập cực đại với tính chất không chứa bất kỳ một khóa tối tiểu nào của $K(S)$. Rõ ràng, K' là một khóa và $K' \subset U$. Suy ra tồn tại một PTH $X_i \rightarrow Y_i \in F$ sao cho $X_i \subseteq K'$ và $Y_i \not\subseteq K'$. Từ $Y_i \not\subseteq K'$ suy ra $K' \cup Y_i \supseteq K_j$ với $K_j \in K(S)$, hay $K_j \setminus Y_i \subseteq K'$. Và do đó $X_i \cup (K_j \setminus Y_i) \subseteq K'$. Điều này có nghĩa $X_i \cup (K_j \setminus Y_i) \not\subseteq K_p$ với mọi $K_p \in K(S)$.

Ngược lại, nếu tồn tại PTH $X_i \rightarrow Y_i \in F$ thì theo tính gia tăng, $X_i \cup (K_j \setminus Y_i) \rightarrow K_j \in F^+$ với $K_j \in K(S)$. Điều này có nghĩa $X_i \cup (K_j \setminus Y_i)$ là một khóa và do đó nó phải chứa ít nhất một khóa tối tiểu. Nhưng theo giả thiết $X_i \cup (K_j \setminus Y_i)$ không chứa bất kỳ một khóa tối tiểu trong $K(S)$, suy ra $X_i \cup (K_j \setminus Y_i)$ phải chứa một khóa tối tiểu mới chưa có trong $K(S)$. $\square$

Thuật toán 4.12 (Thuật toán tìm tất cả khóa tối tiểu)

Vào: LĐQH $S = (U, F)$.

Ra: Tập tất cả các khóa tối tiểu $Key(S)$.

Phương pháp:

Bước 1. Sử dụng Thuật toán 4.11, tìm một khóa tối tiểu K của S , và đặt $Key(S) := \{K\}$.

Bước 2. Với mỗi PTH $X_i \rightarrow Y_i \in F$, với mỗi khóa tối tiểu $K_j \in Key(S)$, nếu

$$X_i \cup (K_j \setminus Y_i) \not\subseteq K_p, \forall K_p \in Key(S) \quad (4.5)$$

thì dùng Thuật toán 4.11 tìm một khóa tối tiểu mới K' từ khóa $X_i \cup (K_j \setminus Y_i)$.

Bổ sung $Key(S) := Key(S) \cup \{K'\}$.

Lặp lại quá trình trên cho đến khi điều kiện (4.5) không thỏa mãn.

Bước 3. Kết luận, $Key(S)$ là tập tất cả các khóa tối tiểu cần tìm.

Dễ kiểm chứng được Thuật toán 4.12 có độ phức tạp thời gian là hàm mũ theo số thuộc tính của U .

Ví dụ 4.18. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c, d, e, g\}$ và $F = \{b \rightarrow c, c \rightarrow b, a \rightarrow gd\}$. Biết rằng S không phải có một khóa tối tiểu duy nhất và $\{a, c, e\}$ là một khóa tối tiểu. Đặt $Key(S) = \{\{a, c, e\}\}$. Vì tồn tại PTH $b \rightarrow c \in F$ sao cho $\{b\} \cup (\{a, c, e\} \setminus \{c\}) = \{a, b, e\} \not\subseteq K_p$ với mọi $K_p \in Key(S)$ nên S có thêm một khóa tối tiểu mới chứa trong khóa $\{a, b, e\}$ chưa có trong $Key(S)$. Khóa tối tiểu mới này được tìm thấy như sau: xây dựng dãy các tập thuộc tính K_0, K_1, K_2, K_3 :

$$K_0 = \{a, b, e\}$$

$$K_1 = K_0$$

$$K_2 = K_1$$

$$K_3 = K_2.$$

$$\text{Suy ra } Key(S) = \{\{a, c, e\}, \{a, b, e\}\}.$$

Lúc này, không tồn tại PTH $X_i \rightarrow Y_i \in F, K_j \in Key(S)$ sao cho $X_i \cup (K_j \setminus Y_i) \not\subseteq K_p$ với mọi $K_p \in Key(S)$.

Vậy, tập tất cả khóa tối thiểu của S là

$$Key(S) = \{\{a, c, e\}, \{a, b, e\}\}.$$

4.5. Phản khóa của lược đồ quan hệ

4.5.1. Định nghĩa

Xét LĐQH $S = (U, F)$. Một *phản khóa* của S là một tập con $K^{-1} \subseteq U$ thỏa các điều kiện sau:

- i) $(K^{-1})^+ \neq U$,
- ii) $\forall a \in U \setminus K^{-1}$ ta có $(K^{-1} \cup \{a\})^+ = U$.

Như vậy, phản khóa là một tập con thật sự của U và cực đại với tính chất không chứa bất kỳ một khóa nào của S . Tương tự như khóa tối thiểu, mỗi LĐQH có ít nhất là một phản khóa, và phản khóa có thể là tập $\emptyset$. Ký hiệu tập tất cả các phản khóa của LĐQH S là $Antikkey(S)$. Dễ thấy $Antikkey(S)$ cũng là một hệ Sperner. Do đó, không có hai phản khóa nào là chứa nhau. Lúc này, họ $Antikkey(S)$ có thể biểu diễn qua $Key(S)$ như sau [16]:

$$Antikkey(S) = \{K^{-1} \subset U | (K \in Key(S) \Rightarrow K \not\subseteq K^{-1}) \text{ và } (K^{-1} \subset X \Rightarrow \exists K \in Key(S), K \subseteq X)\}$$

và

$$Antikkey(S) = \overline{Tran(Key(S))}.$$

Vậy nên, nếu cho trước tập $Key(S)$ thì trên cơ sở kết quả trên

và thuật toán tìm tập tất cả các đại diện chúng ta có ngay thuật toán tìm $Antikkey(S)$: đầu tiên tính $Tran(Key(S))$, sau đó tính phần bù của $Tran(Key(S))$. Kết quả thu được chính là $Antikkey(S)$.

Ví dụ 4.19. Xét LDQH $S = (U, F)$ trong Ví dụ 4.15. Ta đã biết $Key(S) = \{\{c, s\}, \{s, z\}\}$. Vận dụng định nghĩa phản khóa hoặc các đặc trưng của họ $Antikkey(S)$ ngay trên, chúng ta dễ dàng thu được $Antikkey(S) = \{\{s\}, \{c, z\}\}$.

4.5.2. Một số đặc trưng cơ bản

Mệnh đề 4.20. $Antikkey(S) = \{\emptyset\}$ khi và chỉ khi $Key(S) = \{\{a\} : a \in U\}$.

Chứng minh. Giả sử $Antikkey(S) = \{\emptyset\}$. Tức là, với mỗi phản khóa K^{-1} ta có $K^{-1} = \emptyset$. Theo định nghĩa phản khóa, suy ra với mọi thuộc tính $a \in U$, thì

$$\{a\}^+ = (\{a\} \cup \emptyset)^+ = (K^{-1} \cup \{a\})^+ = U$$

và

$$(\{a\} \setminus \{a\})^+ = \emptyset^+ = (K^{-1})^+ \neq U.$$

Như vậy, với mỗi thuộc tính $a \in U$ chúng ta có $\{a\}$ là một khóa tối tiểu của LDQH S .

Ngược lại, giả sử $Key(S) = \{\{a\} : a \in U\}$, tức là $K = \{a\}$ với mọi $K \in Key(S)$ và $a \in U$. Theo định nghĩa khóa tối tiểu, suy ra với mọi $a \in U$:

$$\emptyset^+ = (\{a\} \setminus \{a\})^+ \neq U$$

và

$$(\emptyset \cup \{a\})^+ = \{a\}^+ = U.$$

Vậy, tập $\emptyset$ là phản khóa duy nhất của LDQH S . □

Mệnh đề 4.21. Với mọi LĐQH S ta có

$$\bigcup Key(S) = U \setminus \bigcap Antikey(S).$$

Chứng minh. Giả sử a là một thuộc tính cơ bản bất kỳ của S . Suy ra, tồn tại một khóa tối tiểu K sao cho $a \in K$. Lúc này, ta xây dựng một tập M cực đại chứa tập $K \setminus \{a\}$ sao cho $M^+ \neq U$. Tập M như thế rõ ràng là một phản khóa của S và không chứa a . Từ đây, suy ra rằng nếu a xuất hiện trong một khóa tối tiểu của S thì a không thể xuất hiện trong bất kỳ phản khóa nào đó của S . Tức là, $a \in U \setminus \bigcap Antikey(S)$.

Ngược lại, giả sử a là một thuộc tính không cơ bản của S và a không xuất hiện trong bất kỳ $K^{-1} \in Antikey(S)$. Suy ra, $K^{-1} \cup \{a\}$ không thể là khóa của S . Điều này mâu thuẫn với tính chất của phản khóa. Vậy, $a \in \bigcap Antikey(S)$. $\square$

Hệ quả 4.5. Với mọi LĐQH S ta có

$$\bigcap Antikey(S) = U_N.$$

Chúng ta có tiếp đặc trưng cơ bản khác của họ $Antikey(S)$.

Bổ đề 4.6. Nếu $Y \supset X$, với $X \in \max(Closed(S) \setminus \{U\})$, thì Y là một khóa của LĐQH S .

Chứng minh. Chứng minh phản chứng. Giả sử $Y^+ \neq U$. Suy ra $Y^+ \in (Closed(S) \setminus \{U\})$. Hơn nữa $Y^+ \supset X$. Điều này mâu thuẫn với giả thiết X là một phần tử tối đại trong họ $Closed(S) \setminus \{U\}$. Như vậy, Y phải là một khóa của LĐQH S . $\square$

Định lý 4.22. Với mọi LĐQH S ta có

$$Antikey(S) = \max(Closed(S) \setminus \{U\}).$$

Chứng minh. Giả sử K^{-1} là một phản khóa của S và $K^{-1} \subset (K^{-1})^+$.

Theo định nghĩa của phản khóa, ta có $K^{-1} \neq U$ và

$$(K^{-1})^+ = ((K^{-1})^+)^+ = U.$$

Suy ra, K^{-1} là một khóa của S . Điều này mâu thuẫn với giả thiết $K^{-1} \in \text{Antikey}(S)$. Do đó, ta phải có $(K^{-1})^+ = K^{-1}$. Suy ra $K^{-1} \in \text{Closed}(S) \setminus \{U\}$. Ngoài ra, nếu có bất kỳ một $X \in \text{Closed}(S) \setminus \{U\}$ sao cho $X \supset K^{-1}$ thì $X^+ = U \neq X$. Điều này mâu thuẫn với X là một tập đóng. Vậy $K^{-1} \in \max(\text{Closed}(S) \setminus \{U\})$.

Ngược lại, nếu $X \in \max(\text{Closed}(S) \setminus \{U\})$ và có một khóa tối tiểu K sao cho $K \subset X$ thì $X^+ = U$, tức là $X = U$. Điều này mâu thuẫn với giả thiết $X \neq U$. Như vậy, không tồn tại một khóa tối tiểu nào của S chứa trong X . Ngoài ra, rõ ràng nếu có một tập thuộc tính Y sao cho $Y \supset X$ thì theo Bổ đề 4.6 ta có $Y^+ = U$. Vậy, $X \in \text{Antikey}(S)$. $\square$

4.6. Chuẩn hóa lược đồ quan hệ

LĐQH với tập PTH thỏa mãn một số điều kiện nào đó được gọi là dạng chuẩn. Chúng ta có tổng cộng bốn dạng chuẩn cho PTH theo thứ tự 1NF, 2NF, 3NF và BCNF. Các dạng chuẩn này được E. F. Codd đề xuất từ năm 1971 cho đến năm 1974, với mục đích loại bỏ các dư thừa dữ liệu trong CSDL. Mức độ loại bỏ thứ tự tăng dần từ 1NF, 2NF, 3NF cho đến BCNF. Khi một LĐQH không ở dạng chuẩn hoặc ở dạng chuẩn thấp (chẳng hạn như 1NF, 2NF), thì chúng ta sẽ biến đổi (tách) nó về các dạng chuẩn cao hơn (chẳng hạn 3NF hay BCNF). Quá trình biến đổi các LĐQH về các dạng chuẩn như vậy được gọi là chuẩn hóa LĐQH. Như vậy có thể hiểu, thuật ngữ chuẩn hóa là mô tả quá trình từng bước và có thể đảo ngược, trong đó một LĐQH có thể thay bằng một tập các LĐQH con có dạng đơn giản hơn và cấu trúc cân đối hơn. Việc chuẩn hoá LĐQH đóng một vai trò cực kỳ quan trọng trong quá trình thiết kế CSDL quan hệ. Nhờ có chuẩn hoá, chúng ta tránh được dư thừa dữ liệu, dị thường cập nhật và do đó tăng tốc độ của các phép toán xử lý

trên quan hệ, nhưng vẫn không làm mất đi bất kỳ dữ liệu nào.

4.6.1. Phép tách lược đồ quan hệ

4.6.1.1. Định nghĩa

Xét LĐQH $S = (U, F)$. Một *phép tách* của U hay S là một họ các tập con của U , $\rho = \{X_1, X_2, \dots, X_k\}$, sao cho $\bigcup_{i=1}^k X_i = U$. Các thành phần X_i trong ρ không nhất thiết phải phân biệt nhau. Người ta còn gọi phép tách như thế này là *phép tách dọc*. Như vậy, một phân hoạch của U cũng chính là một phép tách. Tuy nhiên, điều ngược lại thì không đúng.

Với tập con thuộc tính $Z \subseteq U$, *chiếu* của F lên Z , ký hiệu $\Pi_Z(F)$, là tập các PTH xác định bởi:

$$\Pi_Z(F) = \{X \rightarrow Y : X \rightarrow Y \in F^+, X \cup Y \subseteq Z\}.$$

Lúc này, mỗi LĐQH $S_i = (X_i, \Pi_{X_i}(F))$ còn được gọi là *LĐQH con* của S . Lưu ý, với mỗi phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ ta luôn có:

$$F \models \bigcup_{i=1}^k \Pi_{X_i}(F).$$

4.6.1.2. Phép tách không mất thông tin

Cho $\rho = \{X_1, X_2, \dots, X_k\}$ là một phép tách của LĐQH S . Phép tách ρ được gọi là *(kết nối) không mất thông tin* ứng với F nếu

$$\forall R \in \text{Sat}(F) : \Pi_{X_1}(R) \bowtie \Pi_{X_2}(R) \bowtie \dots \bowtie \Pi_{X_k}(R) = R.$$

Trong trường hợp ngược lại, phép tách ρ được gọi là *(kết nối) mất thông tin*.

Tính chất không mất thông tin của phép tách là rất cần thiết, nó đảm bảo quan hệ bị tách có thể khôi phục lại qua phép kết nối từ các

quan hệ chiều của nó. Ánh xạ m_ρ xác định bởi $m_\rho(R) = \Pi_{X_1}(R) \bowtie \Pi_{X_2}(R) \bowtie \cdots \bowtie \Pi_{X_k}(R)$ được gọi là *ánh xạ chiếu-kết nối*. Như vậy, ánh xạ $m_\rho(R)$ là kết nối (tự nhiên) các chiều của quan hệ R lên các thành phần X_i của phép tách ρ . Khi đó, điều kiện để phép tách ρ không mất thông tin ứng với F sẽ là:

$$\forall R \in \text{Sat}(F) : m_\rho(R) = R.$$

Mệnh đề 4.23. *Giả sử $\rho = \{X_1, X_2, \dots, X_k\}$ là một phép tách của LĐQH S và các quan hệ $R_1, R_2 \in \text{Sat}(F)$. Khi đó*

- 1) $R_1 \subseteq m_\rho(R_1)$.
- 2) Nếu $R_1 \subseteq R_2$ thì $m_\rho(R_1) \subseteq m_\rho(R_2)$.
- 3) $m_\rho(m_\rho(R_1)) = m_\rho(R_1)$.

Chứng minh. 1) Hiển nhiên. Vì với mọi $t \in R_1$ thì $t(X_i) \in \Pi_{X_i}(R_1)$, với $i = 1, 2, \dots, k$. Khi đó, theo định nghĩa của phép kết nối ta có $t \in m_\rho(R_1)$.

2) Nếu $R_1 \subseteq R_2$ thì $\Pi_{X_i}(R_1) \subseteq \Pi_{X_i}(R_2)$, với mọi $i = 1, 2, \dots, k$. Và do đó, $m_\rho(R_1) \subseteq m_\rho(R_2)$.

3) Đặt $R = m_\rho(R_1)$. Dễ kiểm chứng được $\Pi_{X_i}(R_1) = \Pi_{X_i}(R)$, với mọi $i = 1, 2, \dots, k$. Khi đó

$$\begin{aligned} m_\rho(m_\rho(R_1)) &= m_\rho(R) \\ &= \Pi_{X_1}(R) \bowtie \Pi_{X_2}(R) \bowtie \cdots \bowtie \Pi_{X_k}(R) \\ &= \Pi_{X_1}(R_1) \bowtie \Pi_{X_2}(R_1) \bowtie \cdots \bowtie \Pi_{X_k}(R_1) \\ &= m_\rho(R_1). \end{aligned}$$

□

Sau đây là thuật toán kiểm tra tính không mất thông tin của một phép tách với độ phức tạp thời gian đa thức dựa vào các biến đổi trên bảng.

Thuật toán 4.13 (Kiểm tra tính không mất thông tin của một phép tách)

Vào: LĐQH $S = (U, F)$ với $U = \{a_1, a_2, \dots, a_n\}$ và một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của S .

Ra: Cho biết ρ có mất thông tin hay không.

Phương pháp:

Bước 1. (*Lập bảng ban đầu*) Lập ma trận cỡ $k \times n$ với các phần tử (i, j) được xác định theo công thức sau:

$$(i, j) = \begin{cases} x_j & \text{nếu } a_j \in X_i \\ y_{ij} & \text{ngược lại} \end{cases}$$

trong đó hàng i tương ứng với thành phần X_i ($1 \leq i \leq k$) và cột j tương ứng với thuộc tính a_j ($1 \leq j \leq n$).

Bước 2. (*Biến đổi bảng*) Lần lượt xét các PTH $X \rightarrow Y \in F$. Với mỗi PTH như vậy tìm xem trong bảng những hàng nào có giá trị giống nhau trên X nhưng khác nhau trên Y . Nếu có những hàng như thế hãy làm cho chúng bằng nhau trên Y . Khi làm bằng, nếu một trong hai giá trị là x_j thì cho giá trị kia trở thành x_j , còn nếu hai giá trị là y_{ij} và y_{kj} thì cho một trong hai giá trị bằng giá trị kia. Quá trình biến đổi cứ tiếp tục như thế, kể cả việc lặp lại PTH đã được áp dụng, cho đến khi không thể biến đổi bảng được nữa.

Bước 3. (*Kết luận*) Nếu trong bảng cuối cùng có chứa một hàng gồm $x_1, x_2, \dots, x_n$ thì phép tách ρ là không mất thông tin, ngược lại ρ là phép tách mất thông tin.

Có thể thấy, Thuật toán 4.13 là rất hiệu quả. Tính đúng của Thuật toán 4.13 được dành làm bài tập (có thể tham khảo chứng minh trong [11]).

Ví dụ 4.20. Cho LĐQH $S = (U, F)$ với tập thuộc tính $U = \{a, b, c, d, e\}$ và tập PTH

$$F = \{a \rightarrow c, b \rightarrow c, c \rightarrow d, de \rightarrow c, ce \rightarrow a\}.$$

Xét phép tách $\rho = \{\{a, d\}, \{a, b\}, \{b, e\}, \{c, d, e\}, \{a, e\}\}$ của S .

Khi đó, ta có bảng ban đầu là

	a	b	c	d	e
$\{a, d\}$	x_1	y_{12}	y_{13}	x_4	y_{15}
$\{a, b\}$	x_1	x_2	y_{23}	y_{24}	y_{25}
$\{b, e\}$	y_{31}	x_2	y_{33}	y_{34}	x_5
$\{c, d, e\}$	y_{41}	y_{42}	x_3	x_4	x_5
$\{a, e\}$	x_1	y_{52}	y_{53}	y_{54}	x_5

Lần lượt xét các PTH và thực hiện các biến đổi:

$a \rightarrow c$: thay y_{23} và y_{53} bằng y_{13}

$b \rightarrow c$: thay y_{33} bằng y_{13}

$c \rightarrow d$: thay y_{24}, y_{34}, y_{54} bằng x_4

$de \rightarrow c$: thay y_{13} bằng x_3

$ce \rightarrow a$: thay y_{31}, y_{41} bằng x_1

Khi đó, ta có bảng

	a	b	c	d	e
$\{a, d\}$	x_1	y_{12}	x_3	x_4	y_{15}
$\{a, b\}$	x_1	x_2	x_3	x_4	y_{25}
$\{b, e\}$	x_1	x_2	x_3	x_4	x_5
$\{c, d, e\}$	x_1	y_{42}	x_3	x_4	x_5
$\{a, e\}$	x_1	y_{52}	x_3	x_4	x_5

Hàng thứ 3 trong bảng là $x_1, x_2, \dots, x_5$. Do vậy, phép tách ρ là không mất thông tin.

Trong thuật toán trên, chúng ta lưu ý khi cho hai ký hiệu bằng nhau thì tất cả các xuất hiện của chúng trong cột cũng phải cho bằng như thế. Chúng ta sẽ thực hiện không “đầy đủ” nếu chỉ cho bằng nhau những ký hiệu nằm trong phạm vi của PTH $X \rightarrow Y$.

Thuật toán 4.13 áp dụng được cho các phép tách có số thành phần bất kỳ. Tuy nhiên, trường hợp nếu phép tách chỉ có hai thành phần thì việc kiểm tra trở nên đơn giản hơn qua điều kiện đủ sau.

Định lý 4.24. *Giả sử $\rho = \{X_1, X_2\}$ là một phép tách của LĐQH S .*

Khi đó, phép tách ρ không mất thông tin ứng với F nếu

$$X_1 \cap X_2 \rightarrow X_1 \setminus X_2 \in F$$

hoặc

$$X_1 \cap X_2 \rightarrow X_2 \setminus X_1 \in F.$$

Chứng minh. Vận dụng Thuật toán 4.13 cho phép tách $\rho = \{X_1, X_2\}$, chúng ta thu được bảng ban đầu như sau:

	$X_1 \cap X_2$	$X_1 \setminus X_2$	$X_2 \setminus X_1$
X_1	$x_i x_{i+1} \dots x_p$	$x_j x_{j+1} \dots x_q$	$y_{1j} y_{1(j+1)} \dots y_{1q}$
X_2	$x_i x_{i+1} \dots x_p$	$y_{2j} y_{2(j+1)} \dots y_{2q}$	$x_j x_{j+1} \dots x_q$

Ở đây $1 \leq i, j, p, q \leq n$. Rõ ràng, nếu $X_1 \cap X_2 \rightarrow X_1 \setminus X_2 \in F$ hoặc $X_1 \cap X_2 \rightarrow X_2 \setminus X_1 \in F$, thì theo Thuật toán 4.13 chúng ta có thể biến đổi bảng trên để thu được bảng cuối cùng có một hàng là $x_1, x_2, \dots, x_n$. Điều này có nghĩa, phép tách ρ là không mất thông tin. $\square$

Hệ quả 4.6. Cho LĐQH S và một PTH $X \rightarrow Y \in F$. Khi đó, phép tách $\rho = \{\{X \cup Y\}, \{X \cup Z\}\}$, với $Z = U \setminus (X \cup Y)$, là không mất thông tin ứng với F .

Ví dụ 4.21. Cho LĐQH $S = (U, F)$ với $U = \{a, b, c\}$ và $F = \{a \rightarrow b, b \rightarrow ac\}$. Xét phép tách $\rho = \{\{a, b\}, \{a, c\}\}$ của S . Vận dụng Định lý 4.24, chúng ta suy ra ngay ρ là một phép tách không mất thông tin ứng với F .

4.6.1.3. Phép tách bảo toàn phụ thuộc hàm

Như chúng ta đã biết ở trên, một phép tách cần phải có đặc tính không mất thông tin vì nó cho phép khôi phục lại một quan hệ từ các chiếu của nó qua các phép kết nối. Bên cạnh đó, một đặc tính quan trọng khác của phép tách cũng cần chúng ta xem xét đến là suy diễn lại được

tập PTH F ban đầu từ các chiều của F lên các thành phần của phép tách.

Một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của LĐQH $S = (U, F)$ được gọi là *bảo toàn tập PTH F* nếu nó thỏa mãn điều kiện:

$$\bigcup_{i=1}^k \Pi_{X_i}(F) \models F.$$

Lưu ý, hai tính chất không mất thông tin và bảo toàn tập PTH F là độc lập với nhau. Tức là, một phép tách có thể không mất thông tin ứng với F nhưng không bảo toàn được F , và ngược lại một phép tách có thể bảo toàn F nhưng mất thông tin.

Ví dụ 4.22. Cho LĐQH $S = (U, F)$ với $U = \{a, b, c, d\}$, $F = \{a \rightarrow b, c \rightarrow d\}$ và một phép tách $\rho = \{\{a, b\}, \{c, d\}\}$.

Ta có $\Pi_{\{a,b\}} = \{a \rightarrow b\}$ và $\Pi_{\{c,d\}} = \{c \rightarrow d\}$ (để ý các PTH tầm thường chúng ta không quan tâm). Suy ra, phép tách ρ bảo toàn F . Tuy nhiên, dễ thấy phép tách ρ là mất thông tin ứng với F .

Ví dụ 4.23. Cho LĐQH $S = (U, F)$ với $U = \{a, b, c\}$, $F = \{a \rightarrow b, b \rightarrow c\}$ và một phép tách $\rho = \{\{a, b\}, \{a, c\}\}$.

Dễ thấy, phép tách ρ là không mất thông tin ứng với F . Tuy nhiên, ta có $\Pi_{\{a,b\}} = \{a \rightarrow b\}$ và $\Pi_{\{a,c\}} = \{a \rightarrow c\}$. Do đó, phép tách ρ là không bảo toàn F .

Bằng cách ký hiệu $G = \bigcup_{i=1}^k \Pi_{X_i}(F)$, thì lúc đó phép tách ρ bảo toàn F nếu $G \models F$. Để kiểm tra điều kiện này, chúng ta lần lượt xét các PTH $X \rightarrow Y \in F$ và xác định xem $Y \subseteq X_G^+$ hay không? Trong đó, để tính bao đóng X^+ mà không cần G là tính lặp đi lặp lại bao đóng $X_{\Pi_{X_i}(F)}^+$. Sau đây là thuật toán kiểm tra tính bảo toàn tập PTH của một phép tách với độ phức tạp thời gian đa thức theo kích thước của S .

Thuật toán 4.14 (Kiểm tra tính bảo toàn tập PTH của một phép tách)

Vào: LĐQH $S = (U, F)$ và một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của S .

Ra: Cho biết ρ có bảo toàn F hay không.

Phương pháp:

Bước 1. Ký hiệu

$$G = \bigcup_{i=1}^k \Pi_{X_i}(F).$$

Bước 2.

Xét PTH $X \rightarrow Y \in F$. Đặt $Z := X$.

Với mọi $i = 1, 2, \dots, k$, tính $Z := Z \cup ((Z \cap X_i)_F^+ \cap X_i)$. Đặt $X_G^+ := Z$.

Nếu $Y \subseteq X_G^+$ thì $G \models X \rightarrow Y$.

Bước 3. Nếu $G \models X \rightarrow Y$ với mọi $X \rightarrow Y \in F$, thì kết luận phép tách ρ bảo toàn F . Ngược lại, kết luận phép tách ρ không bảo toàn F .

Ví dụ 4.24. Xét LĐQH $S = (U, F)$ với tập thuộc tính $U = \{a, b, c, d\}$ và tập PTH $F = \{a \rightarrow b, b \rightarrow c, c \rightarrow d, d \rightarrow a\}$. Xét phép tách của S là $\rho = \{\{a, b\}, \{b, c\}, \{c, d\}\}$.

Ký hiệu $G = \Pi_{\{a,b\}}(F) \cup \Pi_{\{b,c\}}(F) \cup \Pi_{\{c,d\}}(F)$.

Xét $Z := \{d\}$. Các thành phần $\{a, b\}$ và $\{b, c\}$ không làm thay đổi Z . Thành phần $\{c, d\}$ làm thay đổi Z thành U . Suy ra $\{d\}_G^+ = U$. Do đó $G \models d \rightarrow a$.

Thực hiện tương tự với Z là $\{a\}$, $\{b\}$ và $\{c\}$ thì tương ứng ta thu được $G \models a \rightarrow b$, $G \models b \rightarrow c$ và $G \models c \rightarrow d$.

Như vậy, với mọi PTH trong F đã cho thì G đều suy diễn được. Tức là, phép tách ρ đã xét bảo toàn F .

4.6.2. Các dạng chuẩn của lược đồ quan hệ

4.6.2.1. Dạng chuẩn một

Một giá trị x của một thuộc tính trong LĐQH S được gọi là *giá trị nguyên tố* đối với một ứng dụng P nào đó (của S) nếu ứng dụng P

không giải quyết một phần của x .

Ví dụ 4.25. Xét quan hệ $R \in Rel(U)$, là thể hiện của một LĐQH $S = (U, F)$ nào đó chứa thông tin về họ tên và ngày sinh của mỗi người, như sau:

$$R = \begin{array}{|c|c|} \hline Hoten & Ngaysinh \\ \hline Nguyễn Văn Hoàng & 10/01/1970 \\ \hline Hoàng Đình Lợi & 5/05/1980 \\ \hline Phan Thị Cẩm Ly & 2/10/1980 \\ \hline \end{array}$$

Khi đó, nếu có một ứng dụng P nào đó chỉ quan tâm đến tháng hoặc năm sinh của mỗi người, tức chúng ta chỉ giải quyết một phần của giá trị của thuộc tính $Ngaysinh$, thì các giá trị của thuộc tính $Ngaysinh$ không phải là nguyên tố đối với P . Trong trường hợp này, để các giá trị của thuộc tính $Ngaysinh$ là nguyên tố đối với P thì quan hệ R cần biến đổi thành:

$$R = \begin{array}{|c|c|c|c|} \hline Hoten & NgayS & ThangS & NamS \\ \hline Nguyễn Văn Hoàng & 10 & 1 & 1970 \\ \hline Hoàng Đình Lợi & 5 & 5 & 1980 \\ \hline Phan Thị Cẩm Ly & 2 & 10 & 1980 \\ \hline \end{array}$$

LĐQH $S = (U, F)$ được gọi là ở *dạng chuẩn một* (First Normal Form, hay 1NF) nếu các giá trị của các thuộc tính trong S đều là các giá trị nguyên tố.

Trong lý thuyết CSDL, chúng ta luôn giả thiết các LĐQH tối thiểu là 1NF. Do đó, nếu không nói gì thì các LĐQH luôn được hiểu là 1NF.

4.6.2.2. Dạng chuẩn hai

Cho tập PTH F trên U và PTH $X \rightarrow Y \in F^+$. Ta nói Y *phụ thuộc đầy đủ* vào X (ứng với F) nếu PTH $X \rightarrow Y$ là rút gọn trái. Trong trường hợp ngược lại, Y được gọi là *phụ thuộc bộ phận* vào X . Trường hợp nếu $Y = \{a\}$ và Y phụ thuộc đầy đủ vào X , thì ta chỉ cần nói thuộc tính a phụ thuộc đầy đủ vào X .

Ví dụ 4.26. Xét tập PTH $F = \{ab \rightarrow c, b \rightarrow cd, a \rightarrow c\}$ trên $U = \{a, b, c, d\}$. Ta có thuộc tính c phụ thuộc bộ phận vào $\{a, b\}$, và $\{c, d\}$ phụ thuộc đầy đủ vào $\{b\}$.

LĐQH $S = (U, F)$ được gọi là ở *dạng chuẩn hai* hay *2NF* nếu mọi thuộc tính không cơ bản đều phụ thuộc đầy đủ vào mọi khoá tối thiểu của S . Như vậy, từ định nghĩa suy ra nếu có một thuộc tính không cơ bản phụ thuộc bộ phận vào một khoá tối thiểu nào đó của S thì S không phải là 2NF. Ta nói các PTH làm cho S không phải 2NF là *PTH vi phạm 2NF*.

Các điều kiện đủ sau của 2NF là rõ ràng từ định nghĩa.

Mệnh đề 4.25. Cho LĐQH $S = (U, F)$.

- 1) Nếu mọi khoá tối thiểu $K \in \text{Key}(S)$, $|K| = 1$ thì S là 2NF.
- 2) Nếu $U_N = \emptyset$ thì S là 2NF.

Ví dụ 4.27. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c\}$ và $F = \{a \rightarrow b, b \rightarrow c\}$. Dễ thấy $\text{Key}(S) = \{\{a\}\}$. Do đó S là 2NF.

Ví dụ 4.28. Xét LĐQH ở Ví dụ 4.16, $S = \{ab \rightarrow d, c \rightarrow b\}$. Ta đã biết $\text{Key}(S) = \{\{a, c\}\}$. Suy ra $U_N = \{b, d\}$. Vì $c \rightarrow b \in F$, tức là thuộc tính không cơ bản b phụ thuộc bộ phận vào khoá tối thiểu $\{a, c\}$, do đó S không phải 2NF.

Bài toán kiểm tra một LĐQH $S = (U, F)$ có phải là 2NF hay không đến nay vẫn còn xem bài toán mở, người ta phỏng đoán bài toán này có độ phức tạp thời gian từ hàm mũ trở lên [28].

4.6.2.3. Dạng chuẩn ba

Cho tập PTH F trên U , $X \subseteq U$ và $a \in U$. Ta nói thuộc tính a *phụ thuộc bắc cầu* vào X nếu tồn tại $Y \subseteq U$ sao cho $X \rightarrow Y \in F^+$, $Y \rightarrow a \in F^+$ và $a \notin X \cup Y$. Trong trường hợp ngược lại và $X \rightarrow a \in F^+$ thì a được gọi là *phụ thuộc trực tiếp* vào X .

Ví dụ 4.29. Xét tập PTH $F = \{ab \rightarrow c, c \rightarrow d\}$ trên $U = \{a, b, c, d\}$. Khi đó, ta có thuộc tính d phụ thuộc bắc cầu vào $\{a, b\}$, và thuộc tính c phụ thuộc trực tiếp vào $\{a, b\}$.

LĐQH $S = (U, F)$ được gọi là ở *dạng chuẩn ba* hay *3NF* nếu mọi thuộc tính không cơ bản đều phụ thuộc trực tiếp vào mọi khoá tối thiểu của S . Như vậy, từ định nghĩa suy ra nếu có một thuộc tính không cơ bản phụ thuộc bắc cầu vào một khóa tối thiểu nào đó của S thì S không phải là 3NF. Ta sẽ nói các PTH làm cho S không phải 3NF là *PTH vi phạm 3NF*.

Tương tự 2NF điều kiện đủ sau của 3NF cũng rõ ràng.

Mệnh đề 4.26. Nếu $U_N = \emptyset$ thì LĐQH S là 3NF.

Ví dụ 4.30. Xét LĐQH ở Ví dụ 4.15, $S = (U, F)$ với $U = \{c, s, z\}$ và $F = \{cs \rightarrow z, z \rightarrow c\}$. Như ta đã biết $Key(S) = \{\{c, s\}, \{s, z\}\}$. Do đó $U_N = \emptyset$. Vậy S là 3NF.

Ví dụ 4.31. Xét LĐQH S ở Ví dụ 4.27. Đã biết $Key(S) = \{\{a\}\}$, do đó $U_N = \{b, c\}$. Theo giả thiết $a \rightarrow b, b \rightarrow c \in F$ điều này có nghĩa thuộc tính không cơ bản c phụ thuộc bắc cầu vào khóa tối thiểu $\{a\}$. Suy ra S không phải 3NF.

Bài toán kiểm tra một LĐQH $S = (U, F)$ có phải là 3NF hay không có độ phức tạp là NP-đầy đủ [28].

4.6.2.4. Dạng chuẩn Boyce-Codd

LĐQH $S = (U, F)$ được gọi là ở *dạng chuẩn Boyce-Codd* hay *BCNF* nếu mọi thuộc tính đều phụ thuộc trực tiếp vào mọi khoá tối thiểu của S . Như vậy, từ định nghĩa rõ ràng ta có ngay nếu LĐQH S là BCNF thì S cũng là 3NF, và S không phải BCNF nếu có một thuộc tính nào đó phụ thuộc bắc cầu vào một khóa tối thiểu của S . Chúng ta nói các PTH làm cho S không phải BCNF là *PTH vi phạm BCNF*.

Định lý 4.27. *LĐQH $S = (U, F)$ là BCNF khi và chỉ khi với mọi PTH $X \rightarrow a \in F^+$ và $a \notin X$ thì X là một khoá của S .*

Chứng minh. Ta chứng minh định lý bằng phản chứng. Giả sử X không phải khoá của S . Suy ra, tồn tại một khoá tối tiểu K sao cho $K \rightarrow X \in F^+$, $X \rightarrow K \notin F^+$, $X \rightarrow a \in F^+$ và $a \notin X$. Tức là, thuộc tính a phụ thuộc bắc cầu vào khoá tối tiểu K . Rõ ràng, điều này mâu thuẫn với giả thiết S là BCNF. Vậy, X phải là một khoá của S .

Ngược lại, giả sử S không phải BCNF. Suy ra tồn tại thuộc tính $a \in U$, tồn tại một khoá tối tiểu K sao cho $K \rightarrow X \in F^+$, $X \rightarrow K \notin F^+$, $X \rightarrow a \in F^+$ và $a \notin X$. Từ $X \rightarrow K \notin F^+$, suy ra $X^+ \neq U$. Mâu thuẫn với giả thiết X là một khoá của S . Như vậy, S là BCNF. $\square$

Định lý 4.27 cho thấy trong một LĐQH là BCNF các PTH không tầm thường chỉ có thể ở dạng một khoá xác định hàm một hay nhiều thuộc tính khác. Kết quả này dẫn chúng ta đến ngay kết quả khác: bài toán kiểm tra một LĐQH $S = (U, F)$ có phải là BCNF hay không chỉ là đa thức theo $|U|$ và $|F|$.

Ví dụ 4.32. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c, d\}$ và $F = \{ab \rightarrow c, c \rightarrow abd\}$. Theo Định lý 4.27, dễ thấy S là BCNF.

Ví dụ 4.33. Xét LĐQH ở Ví dụ 4.15: $S = (U, F)$, với $U = \{c, s, z\}$ và $F = \{cs \rightarrow z, z \rightarrow c\}$. Ta có $z \rightarrow c \in F$ với $\{z\}^+ = \{z, c\} \neq U$. Do đó, S không phải BCNF.

4.6.2.5. Mối quan hệ giữa các dạng chuẩn

Định lý 4.28.

- 1) Nếu S là 2NF thì S là 1NF.
- 2) Nếu S là 3NF thì S là 2NF.
- 3) Nếu S là BCNF thì S là 3NF.

Chứng minh. 1) Hiển nhiên từ định nghĩa.

2) Chứng minh S là 2NF bằng phản chứng. Giả sử S không phải 2NF. Suy ra, có một thuộc tính không cơ bản a phụ thuộc bộ phận vào một khóa tối thiểu K . Tức là, tồn tại một tập thuộc tính $X \subset K$ sao cho $X \rightarrow a \in F^+$ và $X \rightarrow K \notin F^+$. Hơn nữa, $a \notin X$ bởi nếu không mâu thuẫn với giả sử $a \in U_N$. Tóm lại, chúng ta có phụ thuộc bắc cầu giữa thuộc tính không cơ bản a với khóa tối thiểu K (qua tập thuộc tính X). Điều này mâu thuẫn với giả thiết S là 3NF.

3) Chứng minh S là 3NF bằng phản chứng. Giả sử S không phải 3NF. Suy ra, có một thuộc tính không cơ bản a phụ thuộc bắc cầu vào một $K \in Key(S)$. Nghĩa là, tồn tại một tập thuộc tính $Y \subseteq U$ sao cho $K \rightarrow Y \in F^+, Y \rightarrow K \notin F^+, Y \rightarrow a \in F^+$ và $a \notin Y$. Từ $Y \rightarrow K \notin F^+$, suy ra Y không phải là khóa. Như vậy, ta thu được $Y \rightarrow a \in F^+, a \notin Y$ và Y không phải là khóa. Theo Định lý 4.27, điều này có nghĩa S không phải là BCNF. Mâu thuẫn với giả thiết. $\square$

Các mệnh đề ngược lại trong Định lý 4.28 là không đúng.

Ví dụ 4.34. Xét LĐQH S trong Ví dụ 4.27. Ta có S là 2NF nhưng không phải 3NF. Do đó, mệnh đề ngược lại của 2) trong Định lý 4.28 là không đúng.

Ví dụ 4.35. Xét LĐQH S trong Ví dụ 4.15. Ta có S là 3NF nhưng không phải BCNF. Do đó, mệnh đề ngược lại của 3) trong Định lý 4.28 là không đúng.

Từ Định lý 4.28 chúng ta thấy BCNF là dạng chuẩn tốt nhất (đối với PTH). Đây là dạng chuẩn loại bỏ được nhiều dư thừa dữ liệu. Khi thiết kế CSDL quan hệ, người ta thường xây dựng các LĐQH thỏa mãn BCNF hoặc thấp hơn là 3NF. So với BCNF thì 3NF là ít khắt khe hơn, do đó trong thực tế dạng chuẩn được ưa chuộng khi thiết kế là 3NF.

4.6.3. Một số đặc trưng cơ bản của các dạng chuẩn

Hầu hết các kết quả trong mục này được đề xuất trong [5] (1996), người đọc cũng có thể tìm thấy trong tài liệu [28]. Các kết quả này được giáo

trình phát biểu lại qua khái niệm tập đóng với mục đích làm rõ hơn cấu trúc của các dạng chuẩn, và cũng chính vì vậy dẫn đến các chứng minh được sáng sủa.

Xét LĐQH $S = (U, F)$. Từ tập các khóa tối tiểu $Key(S)$, chúng ta xây dựng các họ như sau:

$$\begin{aligned} M(S) &= \{K \setminus \{a\} : a \in K, K \in Key(S)\} \\ Cl(S) &= \{B : B = C^+, C \in M(S)\}. \end{aligned}$$

Khi đó, ta có kết quả sau.

Định lý 4.29. *Các mệnh đề sau là tương đương:*

- 1) S là 2NF.
- 2) Với mỗi $C \in M(S)$ sao cho $C^+ \cap U_N = \emptyset$.
- 3) Với mỗi $B \in Cl(S)$ và $a \in U_N$ sao cho $B \setminus \{a\}$ là một tập đóng.

Chứng minh. 1) $\Rightarrow$ 2). Nếu $U_N = \emptyset$ thì 2) là rõ ràng. Giả sử $U_N \neq \emptyset$. Nếu có $C \in M(S)$ sao cho $C^+ \cap U_N \neq \emptyset$, thì có nghĩa tồn tại một thuộc tính không cơ bản phụ thuộc bộ phận vào khóa tối tiểu. Suy ra, S không phải 2NF. Điều này là mâu thuẫn.

2) $\Rightarrow$ 3). Giả sử có một $B \in Cl(S)$ và một thuộc tính không cơ bản a sao cho $B \setminus \{a\} \subset (B \setminus \{a\})^+$. Khi đó, theo định nghĩa của họ $Cl(S)$, tồn tại một $C \in M(S)$ sao cho $C^+ = B$. Suy ra $a \in (B \setminus \{a\})^+$, và do đó

$$(B \setminus \{a\})^+ = B = C^+.$$

Như vậy, chúng ta phải có $a \in C^+$. Điều này là mâu thuẫn với giả thiết $C^+ \cap U_N = \emptyset$.

3) $\Rightarrow$ 1). Giả sử tồn tại một $X \subset K \in Key(S)$, một thuộc tính không cơ bản a sao cho $a \notin X$ và $X \rightarrow a \in F^+$. Theo định nghĩa của họ $M(S)$, như vậy phải tồn tại một $C \in M(S)$ sao cho $X \subseteq C$. Suy ra

$a \in C^+ \setminus C$. Đặt $B = C^+$. Vì $C \subseteq B \setminus \{a\}$, nên

$$B \setminus \{a\} \subset C^+ \subseteq (B \setminus \{a\})^+.$$

Điều này là mâu thuẫn với giả thiết $(B \setminus \{a\}) \in \text{Closed}(F)$. $\square$

Bây giờ từ tập các phản khóa $\text{Antikey}(S)$, chúng ta xây dựng họ

$$H(S) = \{K^{-1} \setminus U_N : K^{-1} \in \text{Antikey}(S)\}.$$

Khi đó, chúng ta có ngay một điều kiện đủ sau cho 2NF.

Hệ quả 4.7. *Nếu với mọi $A \in H(S)$ sao cho A là tập đóng thì LDQH S là 2NF.*

Một LDQH $S = (U, F)$ được gọi là *đơn* nếu các PTH trong F đều có dạng $a \rightarrow b$, tức là vế trái và vế phải của các PTH chỉ có đúng một thuộc tính. Lúc này, chúng ta có thể kiểm chứng được ngay một LDQH S là đơn khi và chỉ khi với mọi tập đóng bất kỳ X, Y trong S thì $X \cup Y$ cũng là một tập đóng trong S [28]. Khi đó, chúng ta cũng có $(X \cup Y)^+ = X^+ \cup Y^+$.

Định lý 4.30. *LDQH đơn S là 2NF khi và chỉ khi với mọi $A \in H(S)$ thì A là một tập đóng.*

Chứng minh. Giả sử S là 2NF. Trường hợp nếu $U_N = \emptyset$, thì theo định nghĩa của phản khóa ta có A là một tập đóng với mọi $A \in H(S)$. Giả sử $U_N \neq \emptyset$ và tồn tại một $A \in H(S)$ sao cho A không phải là tập đóng. Suy ra phải có một phản khóa K^{-1} để cho $A \cup U_N = K^{-1}$. Vì U_N là giao của tất cả khóa tối thiểu nên suy ra $A \subset K^{-1}$. Do đó, $A^+ \subseteq K^{-1}$ và $A^+ \cap U_N \neq \emptyset$. Giả sử ký hiệu $A = \{a_1, a_2, \dots, a_k\}$. Vì S là LDQH đơn, nên

$$\{a_1\}^+ \cup \{a_2\}^+ \cup \dots \cup \{a_k\}^+ = A^+.$$

Suy ra, tồn tại một $a_i \in A$ để $\{a_i\}^+ \cap U_N \neq \emptyset$. Theo Định lý 4.29, điều này mâu thuẫn với giả thiết S là 2NF. Như vậy $A \in \text{Closed}(F)$.

Điều kiện đủ của định lý thì rõ ràng theo Hệ quả 4.7. $\square$

Trong lớp các LĐQH đơn, việc kiểm tra một LĐQH có phải là 2NF hay không chỉ mất thời gian đa thức theo kích thước của LĐQH [28]. Bây giờ chúng ta tìm hiểu tiếp về các đặc trưng 3NF và BCNF.

Định lý 4.31. *LĐQH S là 3NF khi và chỉ khi với mọi phản khóa K^{-1} và thuộc tính không cơ bản a , thì $K^{-1} \setminus \{a\}$ là một tập đóng.*

Chứng minh. Giả sử S không phải 3NF. Suy ra, tồn tại một tập thuộc tính không phải khóa X , một thuộc tính không cơ bản a sao cho $a \notin X$ và $X \rightarrow a \in F^+$. Theo đặc trưng cơ bản của phản khóa, Định lý 4.22, tồn tại một phản khóa K^{-1} sao cho $X^+ \subseteq K^{-1}$. Điều này dẫn đến $a \in K^{-1}$, và do đó $X \subseteq K^{-1} \setminus \{a\}$. Suy ra $a \in (K^{-1} \setminus \{a\})^+$, hay

$$K^{-1} \setminus \{a\} \subset (K^{-1} \setminus \{a\})^+.$$

Mẫu thuẫn với giả thiết $K^{-1} \setminus \{a\} \in \text{Closed}(F)$.

Điều kiện cần của định lý là rõ ràng theo định nghĩa 3NF. $\square$

Hệ quả 4.8. *LĐQH S là 3NF khi và chỉ khi với mọi tập đóng $X \neq U$, với mọi thuộc tính không cơ bản a sao cho $a \in X$, thì $X \setminus \{a\}$ là một tập đóng.*

Định lý 4.32. *LĐQH S là BCNF khi và chỉ khi với mọi phản khóa K^{-1} và $a \in K^{-1}$ thì $K^{-1} \setminus \{a\}$ là một tập đóng.*

Chứng minh. Giả sử S không phải BCNF. Suy ra, tồn tại một PTH $X \rightarrow a \in F^+$ với $a \notin X$ sao cho $X^+ \neq U$. Khi đó, theo đặc trưng cơ bản của phản khóa, tồn tại một $K^{-1} \in \text{Antikey}(S)$ sao cho $X^+ \subseteq K^{-1}$. Điều này dẫn đến $a \in K^{-1}$, và do đó $X \subseteq K^{-1} \setminus \{a\}$. Suy ra

$$(K^{-1} \setminus \{a\})^+ = K^{-1}.$$

Mẫu thuẫn với giả thiết $K^{-1} \setminus \{a\} \in \text{Closed}(F)$.

Điều kiện cần của định lý thì rõ ràng theo định nghĩa BCNF. $\square$

4.6.4. Chuẩn hóa lược đồ quan hệ qua phép tách

Nếu một LĐQH không phải ở dạng chuẩn cao (hiểu là 3NF hoặc BCNF), thì chúng ta sẽ tách nó về các LĐQH ở dạng chuẩn cao. Quá trình biến đổi các LĐQH như thế này được hiểu là quá trình chuẩn hóa. Thông thường, các tiêu chuẩn cho quá trình chuẩn hóa các LĐQH bao gồm:

- Bảo toàn thông tin (nếu không phép tách là vô nghĩa).
- Đạt được dạng chuẩn tối thiểu là 3NF (mục tiêu).
- Bảo toàn tập PTH F ban đầu (đặc trưng đầy đủ F là tốt).

Chuẩn hóa LĐQH sẽ dựa trên quá trình phân tích các LĐQH, khóa tối tiểu (và do đó thuộc tính không cơ bản) và các PTH.

4.6.4.1. Tách không mất thông tin về dạng chuẩn Boyce-Codd

Xét LĐQH $S = (U, F)$. Chúng ta bắt đầu với việc tìm hiểu một số tính chất cơ bản sau của lớp BCNF.

Bổ đề 4.7.

- 1) Nếu U chỉ gồm hai thuộc tính thì S là BCNF.
- 2) Nếu S không phải BCNF thì tồn tại hai thuộc tính $a, b \in U$ sao cho $U \setminus \{a, b\} \rightarrow a \in F^+$.

Chứng minh.

1) Giả sử $U = \{a, b\}$. Khi đó, các PTH không tầm thường của S chỉ có thể là $a \rightarrow b, b \rightarrow a$. Trường hợp nếu $F = \{a \rightarrow b\}$ hoặc $F = \{b \rightarrow a\}$ hoặc $F = \{a \rightarrow b, b \rightarrow a\}$, thì S là BCNF theo Định lý 4.27. Trường hợp nếu $F = \emptyset$, thì hiển nhiên theo định nghĩa.

2) Giả sử tồn tại PTH $X \rightarrow a \in F^+, a \notin X$ sao cho X không phải là khóa. Suy ra có thuộc tính $b \in U$ mà $b \notin X \cup \{a\}$, bởi vì nếu không $X \rightarrow X \cup \{a\} = U \in F^+$ và khi đó X là một khóa, điều này mâu thuẫn với giả sử. Từ $a, b \notin X$ nên $X \subseteq U \setminus \{a, b\}$. Suy ra $U \setminus \{a, b\} \rightarrow X \in F^+$, và do đó theo tính bắc cầu thì $U \setminus \{a, b\} \rightarrow a \in F^+$. $\square$

Để thấy đảo của tính chất 1) trong Bổ đề 4.7 là không đúng. Đối với tính chất 2) trong Bổ đề 4.7 thì cũng có thể xảy ra PTH $U \setminus \{a, b\} \rightarrow b \in F^+$. Phép tách không mất thông tin có tính chất cơ bản sau.

Bổ đề 4.8. Cho $\rho = \{X_1, X_2, \dots, X_k\}$ là một phép tách không mất thông tin của LĐQH S ứng với F và $\sigma = \{Z_1, Z_2, \dots, Z_l\}$ là một phép tách không mất thông tin của X_i ứng với $\Pi_{X_i}(F)$. Khi đó, phép tách của S thành $\{X_1, X_2, \dots, X_{i-1}, Z_1, Z_2, \dots, Z_l, X_{i+1}, \dots, X_k\}$ cũng không mất thông tin ứng với F .

Chứng minh. Lấy bất kỳ quan hệ $R \in \text{Sat}(F)$. Vì $\rho = \{X_1, X_2, \dots, X_k\}$ là một phép tách không mất thông tin ứng với F , nên chiếu R lên $X_1, X_2, \dots, X_k$ ta sẽ thu được các quan hệ tương ứng $R_1, R_2, \dots, R_k$ sao cho

$$R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_k.$$

Hơn nữa, vì $\sigma = \{Z_1, Z_2, \dots, Z_l\}$ là một phép tách không mất thông tin của X_i ứng với $\Pi_{X_i}(F)$, nên chiếu quan hệ R_i lên $Z_1, Z_2, \dots, Z_l$ ta cũng thu được các quan hệ tương ứng $R_1^i, R_2^i, \dots, R_l^i$ sao cho

$$R_i = R_1^i \bowtie R_2^i \bowtie \dots \bowtie R_l^i.$$

Suy ra

$$R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_{i-1} \bowtie R_1^i \bowtie R_2^i \bowtie \dots \bowtie R_l^i \bowtie R_{i+1} \bowtie \dots \bowtie R_k.$$

Điều này có nghĩa, phép tách S thành $\{X_1, X_2, \dots, X_{i-1}, Z_1, Z_2, \dots, Z_l, X_{i+1}, \dots, X_k\}$ là không mất thông tin ứng với F . $\square$

Trên cơ sở Bổ đề 4.7 và Bổ đề 4.8, thuật toán tách về dạng chuẩn Boyce-Codd sao cho không mất thông tin.

Thuật toán 4.15 (Thuật toán tách thành BCNF)

Vào: LĐQH $S = (U, F)$.

Ra: Một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của S thỏa các tính chất sau:

- Không mất thông tin.
- Các thành phần $X_i \in \rho$ đều là BCNF ứng với $\Pi_{X_i}(F)$.

Phương pháp:

Bước 1. $Z := U; \rho := \emptyset$;

Bước 2. (Lặp: cho đến khi Z với tập PTH trên nó là BCNF)

 Tìm một PTH $X \rightarrow a$ trên Z vi phạm BCNF;

 Tách tập thuộc tính Z thành hai tập thuộc tính $X \cup \{a\}$ và $Z \setminus \{a\}$;

$\rho := \rho \cup \{X \cup \{a\}\}$;

$Z := Z \setminus \{a\}$;

Bước 3. Kết luận ρ là phép tách cần tìm.

Rõ ràng, ý nghĩa của Thuật toán 4.15 là khá đơn giản. Lưu ý, sau quá trình tách tập thuộc tính Z , LĐQH con thu được ($U_1 = X \cup \{a\}, \Pi_{U_1}(F)$) luôn là BCNF. Do đó, chúng ta chỉ còn kiểm tra xem LĐQH con thứ hai ($W_1 = Z \setminus \{a\}, \Pi_{W_1}(F)$) có phải là BCNF nữa hay không mà thôi. Vì quá trình tính chiếu F lên các tập thuộc tính sau khi tách có độ phức tạp hàm mũ theo số thuộc tính, nên suy ra Thuật toán 4.15 có độ phức tạp thời gian hàm mũ theo số thuộc tính.

Ví dụ 4.36. Xét LĐQH $S = (U, F)$ với tập thuộc tính $U = \{c, t, h, r, s, g\}$ (với ý nghĩa các thuộc tính trong U tương ứng là lớp học phần, giáo viên, giờ học, phòng học, sinh viên, điểm học phần) và tập PTH F với ý nghĩa: $F = \{$

$c \rightarrow t$: mỗi lớp học phần do một giáo viên phụ trách,

$hr \rightarrow c$: tại mỗi phòng học, trong mỗi giờ chỉ có một lớp học phần,

$ht \rightarrow r$: tại mỗi giờ học, mỗi giáo viên chỉ có thể dạy được trong một phòng học,

$cs \rightarrow g$: đối với mỗi lớp học phần, mỗi sinh viên chỉ có một điểm đánh giá,

$hs \rightarrow r$: tại mỗi giờ học, mỗi sinh viên chỉ có mặt trong mỗi phòng

học }.

Dễ dàng tính được $Key(S) = \{\{h, s\}\}$, và do đó S không phải BCNF. Đặt $Z := U$ và $\rho := \emptyset$. Lúc đó, quá trình xây dựng phép tách ρ theo Thuật toán 4.15 được thể hiện qua các vòng lặp như sau:

- Vòng lặp 1: PTH $cs \rightarrow g$ vi phạm BCNF. Tách Z thành $U_1 = \{c, s, g\}$ và $W_1 = \{c, t, h, r, s\}$. LDQH con $S_1 = (W_1, \Pi_{W_1}(F))$ không phải BCNF.

- Vòng lặp 2: PTH vi phạm BCNF của S_1 chẳng hạn là $c \rightarrow t$. Tách $Z := W_1$ thành $U_2 = \{c, t\}$ và $W_2 = \{c, h, r, s\}$. LDQH con $S_2 = (W_2, \Pi_{W_2}(F))$ cũng không phải BCNF.

- Vòng lặp 3: PTH vi phạm BCNF của S_2 là $ch \rightarrow r \in \Pi_{W_2}(F)$. Tách $Z := W_2$ thành $U_3 = \{c, h, r\}$ và $W_3 = \{c, h, s\}$. Đến đây các LDQH con đều là BCNF.

Như vậy, một phép tách thành BCNF không mất thông tin của S là

$$\rho = \{\{c, s, g\}, \{c, t\}, \{c, h, r\}, \{c, h, s\}\}.$$

Đề ý, phép tách ρ trên không bảo toàn F , chẳng hạn PTH $ht \rightarrow r$ không suy diễn được từ các chiếu của F lên các thành phần của ρ . Như vậy, chúng ta thấy rằng không phải lúc nào cũng có thể tách một LDQH thành BCNF không mất thông tin mà vẫn bảo toàn được tập PTH. Tuy vậy, chúng ta luôn có thể tìm được một phép tách không thông tin và bảo toàn tập PTH thành 3NF.

4.6.4.2. Tách không mất thông tin và bảo toàn phụ thuộc hàm về dạng chuẩn ba

Sau đây là thuật toán tách về dạng chuẩn ba bảo toàn PTH.

Thuật toán 4.16 (Thuật toán tách thành 3NF)

Vào: LĐQH $S = (U, F)$ với F là một phủ tối tiểu.

Ra: Một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của S thỏa các tính chất sau:

- Bảo toàn F .
- Các thành phần $X_i \in \rho$ đều là 3NF ứng với $\Pi_{X_i}(F)$.

Phương pháp:

Bước 1. Nếu có thuộc tính $b \in U$ sao cho $b \notin X \cup \{a\}$ với mọi PTH $X \rightarrow a \in F$, thì $U := U \setminus \{a\}$.

Bước 2. Xét hai trường hợp. Trường hợp, nếu tồn tại một PTH $X \rightarrow a \in F$ sao cho $X \cup \{a\} = U$ thì $\rho := \{U\}$, và thuật toán dừng. Ngược lại, khởi tạo $\rho := \emptyset$, và với mỗi PTH $X \rightarrow a \in F$ ta đặt $\rho := \rho \cup \{X \cup \{a\}\}$.

Ngoài ra, nếu có các PTH $X \rightarrow a_1, X \rightarrow a_2, \dots, X \rightarrow a_k \in F$ thì thay vì đưa các thành phần $X \cup \{a_i\}$ ($i = 1, 2, \dots, k$) vào trong ρ , chúng ta chỉ cần đưa thành phần $X \cup \{a_1, a_2, \dots, a_k\}$.

Bước 3. Kết luận ρ là phép tách cần tìm.

Bổ đề 4.9. Thuật toán 4.16 cho ra một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của LĐQH S sao cho ρ bảo toàn F và các thành phần $X_i \in \rho$ đều là 3NF ứng với $\Pi_{X_i}(F)$.

Chứng minh. Chứng minh phản chứng. Giả sử có $X \cup \{a\} \in \rho$ đối với PTH $X \rightarrow a$ sao cho nó không phải 3NF ứng với $F_1 = \Pi_{X \cup \{a\}}(F)$. Suy ra, tồn tại một PTH $Y \rightarrow b$ trong LĐQH $S_1 = (X \cup \{a\}, F_1)$ sao cho Y không phải là khóa của S_1 , $b \notin Y$ và b là thuộc tính không cơ bản của LĐQH S_1 . Lúc này ta xét hai trường hợp:

Trường hợp nếu $b = a$, thì suy ra $Y \rightarrow a$. Hơn nữa $Y \subset X$, vì nếu không mâu thuẫn với $Y^+ \neq X \cup \{a\}$. Do đó, chúng ta có thể thay PTH $X \rightarrow a$ bằng PTH $Y \rightarrow a$ trong phủ tối tiểu F . Điều này mâu thuẫn với định nghĩa của phủ tối tiểu.

Trường hợp nếu $b \neq a$, thì suy ra $b \in X$. Vì X là khóa của S_1 nên tồn tại một tập thuộc tính $Z \subseteq X$ sao cho $Z \in \text{Key}(S_1)$. Khi đó, $Z \rightarrow a$ và $b \notin Z$. Do đó $Z \subset X$. Suy ra, chúng ta có thể thay PTH $X \rightarrow a$ bằng PTH $Z \rightarrow a$ trong phủ tối tiểu F . Tương tự, điều này mâu thuẫn với định nghĩa của phủ tối tiểu.

Như vậy, mọi thành phần trong ρ đều phải là 3NF. Ngoài ra, dễ thấy ρ bảo toàn tập PTH F vì $F \subseteq \bigcup_{i=1}^k \Pi_{X_i}(F)$. $\square$

Ví dụ 4.37. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c, d\}$ và $F = \{ab \rightarrow d, d \rightarrow b, d \rightarrow c\}$. Dễ kiểm chứng được S không phải 3NF và tập PTH F là một phủ tối tiểu. Vận dụng Thuật toán 4.16, chúng ta thu được ngay phép tách ρ thành 3NF bảo toàn tập PTH F là:

$$\rho = \{\{a, b, d\}, \{b, c, d\}\}.$$

Kết quả sau cho phép có thể tách một LĐQH không phải 3NF thành 3NF thỏa hai tính chất bảo toàn tập PTH và không mất thông tin.

Định lý 4.33. *Giả sử ρ là một phép tách của LĐQH S thành 3NF được xây dựng bằng Thuật toán 4.16 và K là một khóa tối tiểu của S . Khi đó, $\tau = \rho \cup \{K\}$ là một phép tách của S thỏa mãn các tính chất sau:*

- 1) τ bảo toàn tập PTH F .
- 2) τ không mất thông tin.
- 3) Mọi thành phần của ρ đều là 3NF.

Chứng minh. 1) Hiển nhiên vì ρ bảo toàn tập PTH F thì τ cũng bảo toàn tập PTH F .

3) Chúng ta chỉ cần chứng minh thêm thành phần $K \in \tau$ là 3NF ứng với $\Pi_K(F)$ là xong. Chứng minh phản chứng, giả sử K không phải là 3NF. Suy ra, tồn tại PTH $X \rightarrow a$ trên K sao cho $X \rightarrow K \notin F^+$, $a \notin X$ và a là thuộc tính không cơ bản. Khi đó, $X \subset K \setminus \{a\}$ và do đó $K \setminus \{a\} \rightarrow X \in F^+$. Theo Bổ đề 4.4, điều này là mâu thuẫn với giả thiết K là một khóa tối tiểu. Vậy K là 3NF.

2) Chứng minh của tính chất này được dành cho bài tập. $\square$

Lưu ý, khi vận dụng Định lý 4.33 để tìm phép tách τ , nếu tồn tại một thành phần $X_i \in \rho$ sao cho $K \subseteq X_i$ với một $K \in \text{Key}(S)$, thì

chúng ta không bỏ sung K vào trong ρ , tức là lúc này $\tau = \rho$. Ngoài ra τ tìm được theo Định lý 4.33 không phải là phép tách có số thành phần ít nhất, nghĩa là chúng ta có thể bỏ đi một số thành phần trong nó mà vẫn bảo toàn được các đặc tính mong muốn. Do đó, phép tách τ là không duy nhất.

Ví dụ 4.38. Xét LĐQH $S = (U, F)$ cho trong Ví dụ 4.37. Ở ví dụ trên chúng ta đã tìm được phép tách $\rho = \{\{a, b, d\}, \{b, c, d\}\}$. Ngoài ra, dễ thấy $\{a, b\} \in \text{Key}(S)$ và $\{a, b\} \subseteq \{a, b, d\} \in \rho$. Do đó $\rho = \{\{a, b, d\}, \{b, c, d\}\}$ cũng chính là phép tách thành 3NF bảo toàn tập PTH F và không mất thông tin.

Chúng ta chú ý không phải mọi phép tách không mất thông tin đều hữu ích, thậm chí có một số còn có hại. Sai lầm hay gặp nhất là đi tách một LĐQH đã là BCNF. Phép tách chỉ được sử dụng để giải quyết các vấn đề như dư thừa dữ liệu, bất thường khi chèn và xóa, chứ không phải là mục đích.

4.7. Phụ thuộc đa trị và dạng chuẩn bốn

4.7.1. Khái niệm

Xét quan hệ *DAYHOC* (dạy học) trên tập thuộc tính $U = \{Monhoc, Giaovien, Doituong\}$ như sau:

<i>Monhoc</i>	<i>Giaovien</i>	<i>Doituong</i>
Toán rời rạc	Nguyễn Văn Hùng	Đại học
Toán rời rạc	Trần Văn Tiến	Đại học
Toán rời rạc	Nguyễn Văn Hùng	Trung cấp
Toán rời rạc	Trần Văn Tiến	Trung cấp
Logic mờ	Lê Trung Dũng	Đại học
Logic mờ	Phạm Văn Hiệp	Trung cấp

Đây là giả thiết về dữ liệu trong CSDL của một trường Đại học. Rõ ràng trong thực tế, một môn học trong trường đại học có thể được

dạy bởi nhiều giáo viên và cho nhiều đối tượng. Chẳng hạn, với quan hệ ở trên, một môn học được dạy bởi hai giáo viên và không phụ thuộc vào dạy cho đối tượng nào (nghĩa là nếu Nguyễn Văn Hùng dạy cho đối tượng Đại học thì Trần Văn Tiến cũng dạy cho đối tượng Đại học, và do đó nếu Nguyễn Văn Hùng dạy cho đối tượng Trung cấp thì Trần Văn Tiến cũng dạy cho đối tượng Trung cấp). R. Fagin [2] (1977) gọi ràng buộc như thế này là phụ thuộc đa trị. Lưu ý, trong quan hệ này không có các PTH như $Monhoc \rightarrow Giaovien, Monhoc \rightarrow Doituong$. Định nghĩa hình thức của phụ thuộc đa trị là như sau. Xét tập các thuộc tính $U = \{a_1, a_2, \dots, a_n\}$ và quan hệ $R \in Rel(U)$. Một *phụ thuộc đa trị* (PTĐT) trên U là một mệnh đề có dạng $X \twoheadrightarrow Y$ trong đó $X, Y \subseteq U$ và đọc là “ X xác định đa trị Y ” hay “ Y phụ thuộc đa trị vào X ”. Chúng ta nói PTĐT $X \twoheadrightarrow Y$ *đúng* trên quan hệ R nếu

$$\forall t_1, t_2 \in R : t_1(X) = t_2(X) \Rightarrow \exists t_3 \in R \text{ sao cho}$$

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_1(Y) \\ t_3(Z) = t_2(Z) \end{cases}$$

với $Z = U \setminus (X \cup Y)$. Lúc đó, chúng ta còn nói lại quan hệ R *thỏa* PTĐT $X \twoheadrightarrow Y$ và ký hiệu $R(X \twoheadrightarrow Y)$.

Bởi tính đối xứng của hai bộ t_1 và t_2 (trên X), nên từ định nghĩa chúng ta thấy phải tồn tại thêm một bộ $t_4 \in R$ sao cho

$$\begin{cases} t_4(X) = t_1(X) (= t_2(X)) \\ t_4(Y) = t_2(Y) \\ t_4(Z) = t_1(Z) \end{cases}$$

Trường hợp nếu quan hệ R *không thỏa* PTĐT $X \twoheadrightarrow Y$, thì ta viết $R(\neg X \twoheadrightarrow Y)$.

Như vậy, Y PTĐT vào X khi và chỉ khi với mỗi giá trị của X có tương ứng một tập giá trị (có thể bằng rỗng) trên Y mà không liên quan gì đến các giá trị trên các thuộc tính còn lại $U \setminus X \setminus Y$.

Xét quan hệ *DAYHOC* ở trên chúng ta có các PTĐT *Monhoc* $\rightarrow \rightarrow$ *Giaovien* và *Monhoc* $\rightarrow \rightarrow$ *Doituong* đúng trên *DAYHOC*. Tuy nhiên, với quan hệ sau thì các PTĐT *Monhoc* $\rightarrow \rightarrow$ *Giaovien* và *Monhoc* $\rightarrow \rightarrow$ *Doituong* không còn đúng nữa:

<i>Monhoc</i>	<i>Giaovien</i>	<i>Doituong</i>
Toán rời rạc	Nguyễn Văn Hùng	Đại học
Toán rời rạc	Trần Văn Tiến	Đại học
Toán rời rạc	Trần Văn Tiến	Trung cấp
Logic mờ	Lê Trung Dũng	Đại học
Logic mờ	Phạm Văn Hiệp	Trung cấp

Trong quan hệ này, Trần Văn Tiến dạy cho hệ Trung cấp nhưng Nguyễn Văn Hùng lại không dạy cho hệ Trung cấp.

Bây giờ một cặp $S = (U, D)$ trong đó U là tập các thuộc tính, $D = F \cup M$ là tập gồm các PTH (F) và PTĐT (M) trên U . Xét quan hệ $R \in Rel(U)$. và $D = F \cup M$ là tập các PTH và PTĐT trên U . Ta nói R thỏa D , ký hiệu $R(D)$, nếu với mọi PTH hay PTĐT $f \in D$ thì $R(f)$. Cho f là một PTH hay PTĐT bất kỳ. Ta nói D suy diễn theo quan hệ f , ký hiệu $D \stackrel{FM}{\models} f$, nếu với mọi quan hệ $R \in Rel(U)$ sao cho $R(D)$ thì $R(f)$. Đặt

$$D^* = \{f : D \stackrel{FM}{\models} f\}.$$

Trường hợp nếu D không suy diễn theo quan hệ được f , thì ta viết $D \stackrel{FM}{\not\models} f$. Như vậy $f \notin D^*$ khi và chỉ khi $D \stackrel{FM}{\not\models} f$.

PTĐT thỏa tính bù sau.

Mệnh đề 4.34. Nếu $R(X \rightarrow \rightarrow Y)$ thì $R(X \rightarrow \rightarrow Z)$, với $Z = U \setminus (X \cup Y)$.

Chứng minh. Giả sử $t_1, t_2 \in R$ sao cho $t_1(X) = t_2(X)$. Vì $R(X \rightarrow \rightarrow Y)$,

nên tồn tại một bộ $t_3 \in R$ sao cho

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_2(Y) \\ t_3(Z) = t_1(Z) \end{cases}$$

Vì $U \setminus (X \cup Z) \subseteq Y$, nên ta cũng có

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Z) = t_1(Z) \\ t_3(U \setminus (X \cup Z)) = t_2(U \setminus (X \cup Z)) \end{cases}$$

hay $R(X \rightarrow\rightarrow Z)$. □

Như vậy, vế phải của PTĐT trong suy diễn là đóng đối với phép bù. Ngoài ra nó cũng đóng với phép hợp (cộng tính phải) và phép hiệu.

Mệnh đề 4.35. *Nếu $R(X \rightarrow\rightarrow Y)$ và $R(X \rightarrow\rightarrow Z)$ thì $R(X \rightarrow\rightarrow Y \cup Z)$.*

Chứng minh. Lấy bất kỳ hai bộ $t_1, t_2 \in R$ sao cho $t_1(X) = t_2(X)$. Ta sẽ chứng minh quan hệ R phải chứa một bộ t thỏa mãn

$$\begin{cases} t(X) = t_1(X) \\ t(Y \cup Z) = t_1(Y \cup Z) \\ t(W_1) = t_2(W_1) \end{cases}$$

với $W_1 = U \setminus (X \cup Y \cup Z)$. Thật vậy, vì $R(X \rightarrow\rightarrow Y)$ nên tồn tại một bộ $t_3 \in R$ sao cho

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_1(Y) \\ t_3(W_2) = t_2(W_2) \end{cases}$$

với $W_2 = U \setminus (X \cup Y)$. Cũng vì $R(X \rightarrow\rightarrow Z)$ nên R cũng phải chứa thêm một bộ t_4 thỏa mãn

$$\begin{cases} t_4(X) = t_1(X) \\ t_4(Z) = t_1(Z) \\ t_4(W_3) = t_3(W_3) \end{cases}$$

với $W_3 = U \setminus (X \cup Z)$. Bây giờ ta sẽ khẳng định $t = t_4$. Rõ ràng $t(X) = t_4(X)$. Mặt khác

$$\begin{cases} t_4(Z) = t_1(Z) = t(Z) \\ t_4(Y \cap W_3) = t_3(Y \cap W_3) = t_1(Y \cap W_3) = t(Y \cap W_3) \end{cases}$$

Suy ra

$$t_4(Y \cup Z) = t(Y \cup Z).$$

Vì $W_1 \subseteq W_2 \cap W_3$, nên

$$t_4(W_1) = t_3(W_1) = t_2(W_1) = t(W_1).$$

Như vậy, $t = t_4$. □

Mệnh đề 4.36. Nếu $R(X \rightarrow\rightarrow Y)$ và $R(X \rightarrow\rightarrow Z)$ thì $R(X \rightarrow\rightarrow Y \setminus Z)$ và $R(X \rightarrow\rightarrow Z \setminus Y)$.

Chứng minh. Vì $R(X \rightarrow\rightarrow Y)$ và $R(X \rightarrow\rightarrow Z)$ nên theo Mệnh đề 4.35 suy ra $R(X \rightarrow\rightarrow Y \cup Z)$. Từ đây, vận dụng tính bù ta thu được $R(X \rightarrow\rightarrow W_1)$, với $W_1 = U \setminus (X \cup Y \cup Z)$. Sau đó, áp dụng Mệnh đề 4.35 thêm lần nữa cho $R(X \rightarrow\rightarrow Y)$ và $R(X \rightarrow\rightarrow W_1)$ ta thu được $R(X \rightarrow\rightarrow W_1 \cup Y)$. Cuối cùng, vận dụng tính bù thêm lần nữa cho $R(X \rightarrow\rightarrow W_1 \cup Y)$ ta thu được $R(X \rightarrow\rightarrow U \setminus (X \cup W_1 \cup Y))$.

Mặt khác

$$\begin{aligned}
U \setminus (X \cup W_1 \cup Y) &= U \setminus (X \cup (U \setminus (X \cup Y \cup Z)) \cup Y) \\
&= U \setminus (X \cup (U \setminus Z) \cup Y) \\
&= Z \setminus (X \cup Y) \\
&= (Z \setminus Y) \setminus X.
\end{aligned}$$

Do đó, $R(X \rightarrow\rightarrow (Z \setminus Y) \setminus X)$, hay dễ thấy $R(X \rightarrow\rightarrow Z \setminus Y)$ (Xem Mệnh đề 4.37 ngay sau). Bởi tính đối xứng của Y và Z , chúng ta cũng có $R(X \rightarrow\rightarrow Y \setminus Z)$. $\square$

Mệnh đề 4.37. $R(X \rightarrow\rightarrow Y)$ khi và chỉ khi $R(X \rightarrow\rightarrow Y \setminus X)$.

Chứng minh. Lấy bất kỳ hai bộ $t_1, t_2 \in R$ sao cho $t_1(X) = t_2(X)$. Vì $R(X \rightarrow\rightarrow Y)$ nên phải có một bộ $t_3 \in R$ sao cho

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_2(Y) \\ t_3(U \setminus (X \cup Y)) = t_1(U \setminus (X \cup Y)) \end{cases}$$

hay

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_2(Y \setminus X) \\ t_3(U \setminus ((X \cup (Y \setminus X)))) = t_1(U \setminus (X \cup (Y \setminus X))) \end{cases}$$

Suy ra, $R(X \rightarrow Y \setminus X)$.

Lập luận hoàn toàn tương tự, nếu $R(X \rightarrow\rightarrow Y \setminus X)$ thì ta có $R(X \rightarrow\rightarrow Y)$. $\square$

Để ý, trong định nghĩa PTĐT $X \rightarrow\rightarrow Y$, chúng ta không giả thiết X và Y là phải phân biệt.

4.7.2. Một số tính chất cơ bản

PTĐT và phép tách không mất thông tin có mối quan hệ như sau.

Định lý 4.38. Cho quan hệ $R \in \text{Rel}(U)$ và $X, Y, Z \subseteq U$ với $Z = U \setminus (X \cup Y)$. Khi đó $R(X \rightarrow\rightarrow Y)$ khi và chỉ khi R tách không mất thông tin trên $X_1 = X \cup Y$ và $X_2 = X \cup Z$.

Chứng minh. Giả sử ta có $R(X \rightarrow\rightarrow Y)$. Gọi $R_1 = \Pi_{X_1}(R)$, $R_2 = \Pi_{X_2}(R)$ và $t \in R_1 \bowtie R_2$. Từ $t \in R_1 \bowtie R_2$, suy ra tồn tại $t_1 \in R_1, t_2 \in R_2$ sao cho

$$\begin{cases} t(X) = t_1(X) = t_2(X) \\ t(Y) = t_1(Y) \\ t(Z) = t_2(Z) \end{cases}$$

Vì R_1, R_2 lần lượt là chiếu của R lên X_1, X_2 , nên phải có các bộ $t'_1, t'_2 \in R$ sao cho

$$\begin{cases} t'_1(X \cup Y) = t_1(X \cup Y) \\ t'_2(X \cup Z) = t_2(X \cup Z) \end{cases}$$

Suy ra

$$t'_1(X) = t_1(X) = t_2(X) = t'_2(X).$$

Từ đây và giả thiết $R(X \rightarrow\rightarrow Y)$, suy ra tồn tại $t_3 \in R$ thỏa mãn

$$\begin{cases} t_3(X) = t'_1(X) \\ t_3(Y) = t'_1(Y) \\ t_3(Z) = t'_2(Z) \end{cases}$$

Như vậy, $t_3 = t$, hay $t \in R$. Do đó, $R_1 \bowtie R_2 \subseteq R$. Ngoài ra, chúng ta luôn có $R \subseteq R_1 \bowtie R_2$. Nên R tách không mất thông tin trên $X_1 = X \cup Y$ và $X_2 = X \cup Z$.

Ngược lại, giả sử $R = R_1 \bowtie R_2$. Lấy bất kỳ hai bộ $t_1, t_2 \in R$ sao cho $t_1(X) = t_2(X)$. Rõ ràng, hai quan hệ R_1, R_2 tương ứng phải chứa các bộ t'_1, t'_2 sao cho $t'_1 = t_1(X \cup Y)$ và $t'_2 = t_2(X \cup Z)$. Vì R tách không mất thông tin trên X_1 và X_2 , nên tồn tại $t \in R$ thỏa mãn

$$\begin{cases} t(X \cup Y) = t'_1 = t_1(X \cup Y) \\ t(X \cup Z) = t'_2 = t_2(X \cup Z) \end{cases}$$

Suy ra

$$\begin{cases} t(X) = t_1(X) \\ t(Y) = t_1(Y) \\ t(Z) = t_2(Z) \end{cases}$$

Đề ý, bộ t là kết quả nối của hai bộ t'_1 và t'_2 . Vậy, ta phải có $R(X \rightarrow\rightarrow Y)$. $\square$

Định lý 4.38 cho chúng ta một phương pháp kiểm tra một quan hệ R có thỏa một PTĐT $X \rightarrow\rightarrow Y$ hay không bằng cách chiếu quan hệ R lên các thành phần $X \cup Y$ và $X \cup (U \setminus (X \cup Y))$. Sau đó, kết nối hai quan hệ chiếu này và kiểm tra xem kết quả đó có bằng quan hệ R hay không. Chẳng hạn, xét quan hệ $R = DAYHOC$ ở trên ta có

$$\Pi_{\{Monhoc, Giaovien\}}(R) \bowtie \Pi_{\{Monhoc, Doituong\}}(R) = R.$$

Do đó, ta có $R(Monhoc \rightarrow\rightarrow Giaovien)$, và do đó $R(Monhoc \rightarrow\rightarrow Doituong)$.

Hệ quả 4.9. Nếu $R(X \rightarrow Y)$ thì $R(X \rightarrow\rightarrow Y)$.

Chứng minh. Biết rằng, nếu $R(X \rightarrow Y)$ thì R tách không mất thông tin trên hai thành phần $X \cup Y$ và $X \cup (U \setminus (X \cup Y))$. Khi đó, theo Định lý 4.38 thì $R(X \rightarrow\rightarrow Y)$. $\square$

Như vậy, PTH cũng là một PTĐT. Điều ngược lại chỉ đúng khi với

mỗi giá trị của X , thì tập giá trị tương ứng trên Y có nhiều nhất là một giá trị.

Định lý 4.39. Cho F là tập PTH trên U và $X, Y, Z \subseteq U$ với $Z = U \setminus (X \cup Y)$. Nếu $Y \not\subseteq X^+$ và $Z \not\subseteq X^+$, thì tồn tại một quan hệ $R \in \text{Rel}(U)$ sao cho $R(F)$ nhưng $R(\neg X \rightarrow \rightarrow Y)$.

Chứng minh. Xét quan hệ $R = \{t_1, t_2\}$ trên $U = \{a_1, a_2, \dots, a_n\}$ như sau:

$$t_1 = (x_1, x_2, \dots, x_n)$$

và

$$t_2(a_i) = \begin{cases} x_i & \text{nếu } a_i \in X^+ \\ y_i & \text{ngược lại} \end{cases}$$

với mọi $i = 1, 2, \dots, n$.

Đầu tiên ta chứng minh $R(F)$ bằng phản chứng. Giả sử tồn tại một PTH $Z \rightarrow W \in F$ sao cho $R(\neg Z \rightarrow W)$. Suy ra, $Z \subseteq X^+$ và $W \not\subseteq X^+$. Bởi $Z \subseteq X^+$, nên $X \rightarrow Z \in F^+$. Áp dụng tính bắc cầu cho $Z \rightarrow W \in F^+$ và $X \rightarrow Z \in F^+$, ta thu được $X \rightarrow W \in F^+$, hay $W \subseteq X^+$. Điều này là mâu thuẫn. Vậy $R(F)$.

Tiếp theo ta chứng minh $R(\neg X \rightarrow \rightarrow Y)$ cũng bằng phản chứng. Giả sử $R(X \rightarrow \rightarrow Y)$. Vì $Y \not\subseteq X^+$ và $Z \not\subseteq X^+$, nên tương ứng phải có một thuộc tính $b_1 \in Y$ nhưng $b_1 \notin X^+$ và một thuộc tính $b_2 \in Z$ nhưng $b_2 \notin X^+$. Từ $b_1, b_2 \notin X^+$, suy ra $t_2(b_1) = y_j$ và $t_2(b_2) = y_k$ với $j, k \in \{1, 2, \dots, n\}$. Vì $X \subseteq X^+$ nên $t_1(X) = t_2(X)$. Từ đây và giả thiết $R(X \rightarrow \rightarrow Y)$, suy ra tồn tại một bộ $t_3 \in R$ sao cho

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_1(Y) \\ t_3(Z) = t_2(Z) \end{cases}$$

Tuy nhiên, quan hệ R chỉ chứa hai bộ, nên hoặc $t_3 = t_1$ hoặc

$t_3 = t_2$. Ta giả sử $t_3 = t_1$. Khi đó, ta có

$$t_3(b_2) = t_1(b_2) \neq t_2(b_2).$$

Suy ra $t_3(Z) \neq t_2(Z)$. Điều này là vô lý. Do đó $R(\neg X \rightarrow \rightarrow Y)$. $\square$

Như vậy, Định lý 4.39 cho thấy chỉ có các PTĐT $X \rightarrow \rightarrow Y$ được suy diễn từ tập PTH F nếu $Y \subseteq X^+$ hoặc $U \setminus (X \cup Y) \subseteq X^+$.

Ví dụ 4.39. Xét tập PTH $F = \{a \rightarrow bc, c \rightarrow h\}$ trên tập thuộc tính $U = \{a, b, c, e, h, i\}$. Ta có $\{a\}^+ = \{a, b, c, h\}$ nên $F \models a \rightarrow \rightarrow bch$ và $F \models a \rightarrow \rightarrow c$. Tuy nhiên, $\{a\}^+ \not\subseteq \{e, h\}$ và $\{a\}^+ \not\subseteq \{b, c, i\}$, do đó $F \not\models a \rightarrow \rightarrow eh$.

4.7.3. Hệ tiên đề của phụ thuộc hàm và phụ thuộc đa trị

Sau đây là hệ tiên đề dùng để suy diễn PTH và PTĐT. Phần tiên đề về PTH do W. W. Armstrong đưa ra 1974 và đã được trình bày ở phần trước. Phần tiên đề về PTĐT do C. Beeri, R. Fagin và J. H. Howard đưa ra sau đó 3 năm [2]. Xét tập PTH và PTĐT $D = F \cup M$ trên tập thuộc tính U . Bao đóng của D , ký hiệu D^+ , là tập các PTH và PTĐT trên U nhỏ nhất chứa D sao cho thỏa các tính chất sau: với mọi $X, Y, Z, W \subseteq U$

F1) *Tính phản xạ đối với PTH:*

Nếu $Y \subseteq X$ thì $X \rightarrow Y \in F^+$.

F2) *Tính gia tăng đối với PTH:*

Nếu $X \rightarrow Y \in F^+$ thì $X \cup Z \rightarrow Y \cup Z \in F^+$.

F3) *Tính bắc cầu đối với PTH:*

Nếu $X \rightarrow Y \in F^+$ và $Y \rightarrow Z \in F^+$ thì $X \rightarrow Z \in F^+$.

M1) *Tính bù đối với PTĐT:*

Nếu $X \rightarrow \rightarrow M^+$ thì $X \rightarrow \rightarrow U \setminus (X \cup Y) \in M^+$.

M2) *Tính gia tăng đối với PTĐT:*

Nếu $X \rightarrow \rightarrow Y$ và $Z \subseteq W$ thì $X \cup W \rightarrow \rightarrow Y \cup Z \in M^+$.

M3) *Tính bắc cầu đối với PTĐT*:

Nếu $X \rightarrow \rightarrow Y \in M^+$ và $Y \rightarrow \rightarrow Z \in M^+$ thì $X \rightarrow \rightarrow Z \setminus Y \in M^+$.

FM1) *Tính lặp lại*:

Nếu $X \rightarrow Y \in F^+$ thì $X \rightarrow \rightarrow Y \in M^+$.

FM2) *Tính liên kết*:

Nếu $X \rightarrow \rightarrow Y \in M^+$ và $Z \rightarrow W \in F^+$, với $Z \cap Y = \emptyset$ và $W \subseteq Y$, thì $X \rightarrow W \in M^+$.

Ta gọi các tính chất (F1)-(F3), (M1)-(M3), (FM1) và (FM2) là *tập quy tắc suy diễn Armstrong-BFH* hay *hệ tiên đề Armstrong-BFH*. Các PTĐT $X \rightarrow \rightarrow Y$ thỏa $Y \subseteq X$ hoặc $X \cup Y = U$ được gọi là *PTĐT tầm thường* trên U . Tập D được gọi *suy diễn theo tiên đề f* , ký hiệu $D \stackrel{FM}{\underset{A}{\models}} f$, nếu $f \in D^+$. Nói một cách khác, f được suy diễn theo tiên đề từ D nếu chúng ta xuất phát từ D rồi áp dụng các qui tắc suy diễn (F1)-(F3), (M1)-(M3), (FM1) và (FM2) sau một số hữu hạn lần thì thu được f . Khi D không suy diễn theo tiên đề được f thì ta viết $D \stackrel{FM}{\underset{A}{\not\models}} f$.

Bổ đề 4.10 (Tính đúng của hệ tiên đề). *Hệ tiên đề Armstrong-BFH là đúng, nghĩa là $D^+ \subseteq D^*$.*

Chứng minh. Tính đúng của các tiên đề (F1)-(F3) đã chứng minh ở Bổ đề 4.1. Đối với các tiên đề của PTĐT (M1)-(M3) chúng ta chứng minh như sau: nếu PTĐT $X \rightarrow \rightarrow Y$ được suy diễn theo tiên đề từ M bằng các tiên đề từ (M1) đến (M3), thì PTĐT $X \rightarrow \rightarrow Y$ đúng trong mọi quan hệ mà các PTĐT M đúng.

M1) Đã chứng minh ở Mệnh đề 4.34.

M2) Xét quan hệ $R \in Rel(U)$ sao cho $R(X \rightarrow \rightarrow Y)$, sau đó lấy hai bộ bất kỳ $t_1, t_2 \in R$ thỏa mãn $t_1(X \cup W) = t_2(X \cup W)$. Suy ra

$t_1(X) = t_2(X)$, và do đó tồn tại một bộ $t_3 \in R$ sao cho

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_1(Y) \\ t_3(U \setminus (X \cup Y)) = t_2(U \setminus (X \cup Y)) \end{cases}$$

hay

$$\begin{cases} t_3(X \cup W) = t_1(X \cup W) \\ t_3(Y \cup Z) = t_1(Y \cup Z) \\ t_3(U \setminus (X \cup Y \cup W)) = t_2(U \setminus (X \cup Y \cup W)) \end{cases}$$

Như vậy, $R(X \cup W \rightarrow\rightarrow Y \cup Z)$.

M3) Giả sử ta có $R(X \rightarrow\rightarrow Y)$ và $R(Y \rightarrow\rightarrow Z)$. Đầu tiên chúng ta sẽ chứng minh $R(X \rightarrow\rightarrow Y \cup Z)$. Lấy bất kỳ hai bộ $t_1, t_2 \in R$ sao cho $t_1(X) = t_2(X)$. Vì $R(X \rightarrow\rightarrow Y)$, nên tồn tại một bộ $t_3 \in R$ thỏa mãn

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_1(Y) \\ t_3(U \setminus (X \cup Y)) = t_2(U \setminus (X \cup Y)) \end{cases}$$

Thêm nữa $R(Y \rightarrow\rightarrow Z)$, nên quan hệ R cũng phải chứa thêm một bộ t_4 sao cho

$$\begin{cases} t_4(Y) = t_1(Y) \\ t_4(Z) = t_1(Z) \\ t_4(U \setminus (Y \cup Z)) = t_3(U \setminus (Y \cup Z)) \end{cases}$$

Như vậy

$$\begin{cases} t_4(X) = t_1(X) \\ t_4(Y \cup Z) = t_1(Y \cup Z) \\ t_4(U \setminus (X \cup Y \cup Z)) = t_2(U \setminus (X \cup Y \cup Z)) \end{cases}$$

Suy ra $R(X \rightarrow\rightarrow Y \cup Z)$. Bây giờ chúng ta sẽ chứng minh $R(X \rightarrow\rightarrow Z \setminus Y)$. Vì $R(X \rightarrow\rightarrow Y)$ và $R(X \rightarrow\rightarrow Y \cup Z)$, nên theo Mệnh đề 4.36 suy ra $R(X \rightarrow\rightarrow Z \setminus Y)$.

FM1) Đã chứng minh ở Hệ quả 4.9.

FM2) Giả sử $R(X \rightarrow\rightarrow Y)$ và $R(Z \rightarrow W)$. Sau đó, lấy hai bộ bất kỳ $t_1, t_2 \in R$ sao cho $t_1(X) = t_2(X)$. Vì $R(X \rightarrow\rightarrow Y)$, nên tồn tại một bộ $t_3 \in R$ thỏa mãn

$$\begin{cases} t_3(X) = t_1(X) \\ t_3(Y) = t_1(Y) \\ t_3(U \setminus (X \cup Y)) = t_2(U \setminus (X \cup Y)) \end{cases}$$

Vì $Z \subseteq U \setminus Y = (U \setminus (X \cup Y)) \cup X$, nên $t_3(Z) = t_2(Z)$. Hơn nữa, ta có $R(Z \rightarrow W)$ suy ra $t_3(W) = t_2(W)$. Biết $W \subseteq Y$, do đó $t_3(W) = t_1(W)$. Vậy nên, $t_1(W) = t_2(W)$. Theo định nghĩa PTH, ta có $R(X \rightarrow W)$. $\square$

Trong hệ tiên đề trên, chúng ta thấy ngay tiên đề (FM2) không áp dụng được khi PTH là tầm thường. Trường hợp nếu Y và Z là phân biệt thì tính bắc cầu của PTĐT tương tự như tính bắc cầu của PTH. Tuy nhiên, trong trường hợp tổng quát, Y và Z bất kỳ, thì tính bắc cầu của PTH không còn đúng cho PTĐT nữa. Tìm phản ví dụ cho trường hợp này được xem như làm bài tập.

Từ hệ tiên đề Armstrong-BFH chúng ta suy ra thêm được một số quy tắc suy diễn khác như sau. Một số các quy tắc trong chúng đã được

phát biểu và chứng minh ở mục trên.

Mệnh đề 4.40 (Một số quy tắc suy diễn khác).

M4) Tính cộng tính phải:

Nếu $X \rightarrow\rightarrow Y \in M^+$ và $X \rightarrow\rightarrow Z \in M^+$ thì $X \rightarrow Y \cup Z \in M^+$.

M5) Tính thu hẹp phải (tách):

Nếu $X \rightarrow\rightarrow Y \in M^+$ và $X \rightarrow\rightarrow Z \in M^+$ thì $X \rightarrow\rightarrow Y \setminus Z \in M^+$, $X \rightarrow\rightarrow Z \setminus Y \in M^+$, $X \rightarrow\rightarrow Y \cap Z \in M^+$.

M6) Tính giả bắc cầu:

Nếu $X \rightarrow\rightarrow Y \in M^+$ và $Y \cup W \rightarrow\rightarrow Z \in M^+$ thì $X \cup W \rightarrow\rightarrow Z \setminus (Y \cup W) \in M^+$.

Chứng minh.

M4) Tính chất này chính là Mệnh đề 4.35 và như vậy là đã được chứng minh. Tuy vậy, tính đúng của nó hoàn toàn có thể được suy ra từ hệ tiên đề Armstrong-BFH. Thật vậy, áp dụng tính giao tăng cho $X \rightarrow\rightarrow Z \in M^+$ và $X \rightarrow\rightarrow Y \in M^+$ tương ứng ta thu được $X \rightarrow\rightarrow X \cup Z \in M^+$ và $X \cup Z \rightarrow\rightarrow Y \cup Z \in M^+$. Sau đó, áp dụng tính bù cho $X \cup Z \rightarrow\rightarrow Y \cup Z \in M^+$, ta thu được $X \cup Z \rightarrow\rightarrow U \setminus (X \cup Y \cup Z) \in M^+$. Từ $X \rightarrow\rightarrow X \cup Z \in M^+$ và $X \cup Z \rightarrow\rightarrow U \setminus (X \cup Y \cup Z) \in M^+$, vận dụng tính bắc cầu ta thu được $X \rightarrow\rightarrow U \setminus (X \cup Y \cup Z) \in M^+$. Cuối cùng, vận dụng tính bù thêm lần nữa cho $X \rightarrow\rightarrow U \setminus (X \cup Y \cup Z) \in M^+$, ta thu được $X \rightarrow\rightarrow Y \cup Z \in M^+$.

M5) Tính đóng đối với phép hiệu đã được chứng minh ở Mệnh đề 4.36. Do đó, bây giờ ta chỉ còn chứng minh tính đóng đối với phép giao nữa là xong. Thật vậy, từ $X \rightarrow\rightarrow Y \in M^+$, áp dụng tính bù ta thu được $X \rightarrow\rightarrow W_2 \in M^+$ với $W_2 = U \setminus (X \cup Y)$. Sau đó, vận dụng tính cộng tính phải cho $X \rightarrow\rightarrow Y \setminus Z \in M^+$ và $X \rightarrow\rightarrow W_2 \in M^+$, chúng ta thu được $X \rightarrow\rightarrow W_2 \cup (Y \setminus Z) \in M^+$. Áp dụng tính bù lần nữa cho $X \rightarrow\rightarrow W_2 \cup (Y \setminus Z) \in M^+$, ta thu được $X \rightarrow\rightarrow U \setminus (X \cup W_2 \cup (Y \setminus Z)) \in M^+$.

Hơn nữa

$$\begin{aligned}
U \setminus (X \cup W_2 \cup (Y \setminus Z)) &= U \setminus (X \cup (U \setminus (X \cup Y)) \cup (Y \setminus Z)) \\
&= U \setminus (X \cup (U \setminus Y) \cup (Y \setminus Z)) \\
&= Y \setminus (X \cup (Y \setminus Z)) \\
&= (Y \cap Z) \setminus X.
\end{aligned}$$

Như vậy $X \rightarrow\rightarrow (Y \cap Z) \setminus X \in M^+$, hay $X \rightarrow\rightarrow Y \cap Z \in M^+$.

M6) Suy ra từ tính gia tăng và tính bắc cầu của PTĐT. Đó là áp dụng tính gia tăng cho $X \rightarrow\rightarrow Y \in F^+$, thu được $X \cup W \rightarrow\rightarrow Y \cup W \in M^+$. Sau đó, vận dụng tính bắc cầu cho $X \cup W \rightarrow\rightarrow Y \cup W \in M^+$ và $Y \cup W \rightarrow\rightarrow Z \in M^+$, ta thu được $X \cup W \rightarrow\rightarrow Z \setminus (Y \cup W) \in M^+$. $\square$

Như vậy, chúng ta có thể thấy tính tách của PTĐT không mạnh bằng tính tách của PTH. Trong tính tách của PTH, từ PTH $X \rightarrow Y$ chúng ta suy diễn ra được $X \rightarrow a$ với mọi $a \in Y$. Tuy nhiên trong PTĐT, nếu ta muốn PTĐT $X \rightarrow\rightarrow a$ được suy diễn từ PTĐT $X \rightarrow\rightarrow Y$, thì ta phải tìm được tập Z sao cho $X \rightarrow\rightarrow Z$ và hoặc $Y \cap Z = \{a\}$, hoặc $Y \setminus Z = \{a\}$, hoặc $Z \setminus Y = \{a\}$.

Ví dụ 4.40. Xét tập PTĐT $D = M = \{a \rightarrow\rightarrow bc, eh \rightarrow\rightarrow c\}$ trên tập thuộc tính $U = \{a, b, c, e, h\}$.

Vận dụng hệ tiên đề Armstrong-BFH, từ PTĐT $a \rightarrow\rightarrow bc$ ta suy diễn được PTĐT $a \rightarrow\rightarrow eh$ theo tính bù. Sau đó, áp dụng tính bắc cầu ta suy diễn được PTĐT $a \rightarrow\rightarrow c$. Bởi tính gia tăng, ta suy diễn tiếp được PTĐT $ah \rightarrow c$. Áp dụng tính bù thêm lần nữa, ta thu được PTĐT $ah \rightarrow\rightarrow be$.

Như vậy, ta có $D \stackrel{FM}{\underset{A}{\models}} ah \rightarrow\rightarrow be$.

Ví dụ 4.41. Xét tập PTĐT $D = M = \{a \rightarrow\rightarrow bc, i \rightarrow c\}$ trên tập thuộc tính $U = \{a, b, c, e, i\}$.

Vận dụng tính liên kết, ta sẽ thu được PTH $a \rightarrow c$. Chẳng hạn, quan hệ sau $R \in Rel(U)$ là thỏa D :

$$R = \begin{array}{|c|c|c|c|c|} \hline a & b & c & e & i \\ \hline 0 & 1 & 2 & 5 & 3 \\ \hline 0 & 2 & 2 & 6 & 4 \\ \hline 0 & 2 & 2 & 5 & 3 \\ \hline 0 & 1 & 2 & 6 & 4 \\ \hline \end{array}$$

Định lý 4.41 (Tính đúng và đủ của hệ tiên đề). *Hệ tiên đề Armstrong-BFH là*

- 1) *Đúng, nghĩa là $D^+ \subseteq D^*$*
- 2) *Đủ, nghĩa là $D^+ = D^*$.*

Tính đúng của hệ tiên đề đã chứng minh ở Bổ đề 4.10. Chứng minh về tính đủ của hệ tiên đề người đọc có thể xem trong công trình [2]. Lưu ý, trong [2] người ta cũng chứng minh rằng hệ tiên đề đủ cho chỉ PTĐT (còn gọi là *hệ tiên đề BFH*) là bao gồm (M1)-(M3), (M7) với

M7) *Tính phản xạ:*

Nếu $Y \subseteq X$ thì $X \rightarrow \rightarrow Y \in M^+$.

4.7.4. Cơ sở phụ thuộc

4.7.4.1. Cơ sở phân tách tối thiểu

Xét $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$ là một họ các tập con của tập hữu hạn bất kỳ V sao cho $V = P_1 \cup P_2 \cup \dots \cup P_k$. Một *cơ sở phân tách tối thiểu* của $\mathcal{P}$, ký hiệu $mdsb(\mathcal{P})$, là một phân hoạch $Q_1, Q_2, \dots, Q_l$ của V thỏa mãn:

- i) Với mọi i , P_i là hợp của một số Q_j nào đó.
- ii) Không tồn tại một phân hoạch nào của V với ít phần tử hơn có tính chất (i).

Từ định nghĩa ta thấy ngay cơ sở phân tách tối thiểu $mdsb(\mathcal{P})$ là duy nhất.

Chẳng hạn, với họ $\mathcal{P} = \{\{1, 3, 5, 7, 9\}, \{7, 8, 9\}, \{1, 2, 4, 6, 8\}\}$ trên

tập số nguyên dương $V = \{1, 2, \dots, 9\}$, thì họ

$$mdsb(\mathcal{P}) = \{\{1\}, \{8\}, \{3, 5\}, \{7, 9\}, \{2, 4, 6\}\}$$

là một phân tách tối thiểu của $\mathcal{P}$.

4.7.4.2. Định nghĩa cơ sở phụ thuộc

Xét tập PTĐT M trên U và một tập con thuộc tính $X \subseteq U$. Gọi

$$G(X) = \{Y : Y \subseteq U \text{ và } X \rightarrow\rightarrow Y \in M^+\}.$$

Lúc đó, cơ sở phụ thuộc của X ứng với M là $mdsb(G(X))$, và ký hiệu là $Dep(X, M)$. Khi không sợ nhầm lẫn, ta viết $Dep(X, M)$ bằng $Dep(X)$. Như vậy

$$Dep(X) = mdsb(G(X)).$$

Trường hợp nếu $X = \{a_1, a_2, \dots, a_k\}$ và $Dep(X) = \{\{a_1\}, \{a_2\}, \dots, \{a_k\}, Y_1, Y_2, \dots, Y_l\}$, thì chúng ta viết $X \rightarrow\rightarrow Y_1|Y_2|\dots|Y_l$.

Lưu ý, trong định nghĩa về cơ sở phụ thuộc $Dep(X)$, thay vì xét trên tập $D = F \cup M$ chúng ta chỉ cần xét trên tập M là đủ, trong đó M phải bao gồm:

- Tất cả các PTĐT trong D .
- Với mỗi PTH $X \rightarrow Y \in D$, giả sử $Y = \{a_1, a_2, \dots, a_k\}$, thì ta thay bằng các PTĐT $X \rightarrow\rightarrow a_1, X \rightarrow\rightarrow a_2, \dots, X \rightarrow\rightarrow a_k \in M$.

Bổ đề 4.11. $mdsb(G(X)) \subseteq G(X)$.

Chứng minh. Có thể thấy, nếu tồn tại $Y_1, Y_2 \in G(X)$ sao cho Y_1 chứa các thuộc tính $a \in Y_2$ và $b \notin Y_2$ thì theo tính thu hẹp phải đối với PTĐT, ta có các tập thuộc tính

$$Y_3 = Y_1 \setminus Y_2 \in G(X), Y_4 = Y_1 \cap Y_2 \in G(X).$$

Rõ ràng, $mdsb(G(X))$ sẽ bao gồm các tập khác rỗng Y_3 và Y_4 này. Điều này có nghĩa $mbsb(G(X)) \subseteq G(X)$. $\square$

Ví dụ 4.42. Cho tập PTĐT $M = \{a \rightarrow\rightarrow bc, ei \rightarrow\rightarrow c\}$ trên tập thuộc tính $U = \{a, b, c, e, i\}$.

Với $X = \{a\}$, thì ta có họ

$$G(X) = \{\{a\}, \{b, c\}, \{e, i\}, \{c\}\{b, e, i\}, \{b\}, \{b, c, e, i\}, \{c, e, i\}\}.$$

Lúc này

$$mdsb(G(X)) = \{\{a\}, \{b\}, \{c\}, \{e, i\}\}$$

và do đó cơ sở phụ thuộc của X là

$$Dep(X) = \{\{a\}, \{b\}, \{c\}, \{e, i\}\}.$$

Bổ đề 4.12. $X \rightarrow\rightarrow Y \in M^+$ khi và chỉ khi tồn tại một số tập thuộc tính $Y_i \in Dep(X)$ sao cho $Y = \bigcup Y_i$.

Chứng minh. Nếu $X \rightarrow\rightarrow Y \in M^+$, thì rõ ràng $Y \in G(X)$. Suy ra, Y là hợp của một số tập thuộc tính trong $Dep(X)$.

Ngược lại, nếu $Y = \bigcup Y_i$ với một số $Y_i \in Dep(X)$, thì theo Bổ đề 4.11 ta có $Y_i \in G(X)$. Do đó $X \rightarrow\rightarrow Y_i \in M^+$ với $Y_i \in Dep(X)$. Theo tính cộng tính phải, suy ra $X \rightarrow\rightarrow \bigcup Y_i = Y \in M^+$. $\square$

Như vậy, để kiểm tra một PTĐT $X \rightarrow\rightarrow Y$ đúng hay sai, chúng ta cần xác định cơ sở phụ thuộc $Dep(X)$ và xem $Y \setminus X$ có phải hợp của một số tập thuộc tính trong cơ sở phụ thuộc đó hay không.

4.7.4.3. Thuật toán tìm cơ sở phụ thuộc

Sau đây là thuật toán tìm cơ sở phụ thuộc của một tập thuộc tính.

Thuật toán 4.17 (Thuật toán tìm cơ sở phụ thuộc)

Vào: Tập PTĐT M trên U và $X \subseteq U$.

Ra: Cơ sở phụ thuộc $Dep(X)$.

Phương pháp:

Bước 1. Đầu tiên, đặt $G(X) := \{\{a\} : \forall a \in X\}$. Trường hợp nếu tồn tại PTĐT $X' \rightarrow\rightarrow Y' \in M$ sao cho $X' \subseteq X$, thì

$$G(X) := G(X) \cup \{Y', U \setminus (X' \cup Y')\}.$$

Bước 2. Đặt $Dep(X) := G(X)$.

Bước 3. Sử dụng tính bắc cầu đối với PTĐT để làm mịn $Dep(X)$. Đó là tìm một PTĐT $Z \rightarrow\rightarrow W \in M$ sao cho

$$Z \subseteq Y_1 \cup Y_2 \cup \dots \cup Y_k, \text{ với } Y_1, Y_2, \dots, Y_k \in Dep(X).$$

Đặt $Y = Y_1 \cup Y_2 \cup \dots \cup Y_k$. Suy ra $X \rightarrow\rightarrow W \setminus Y \in M^+$.

Nếu $W \setminus Y$ không phải hợp của một số tập trong $Dep(X)$, thì đặt

$$Dep(X) := mdsb(Dep(X) \cup \{W \setminus Y\}).$$

Ngược lại, nếu $W \setminus Y$ là hợp của một số tập trong $Dep(X)$ thì không thể làm mịn $Dep(X)$ bằng PTĐT $Z \rightarrow\rightarrow W$. Lặp lại quá trình làm mịn trên cho đến khi trong M không có PTĐT nào làm thay đổi được $Dep(X)$.

Bước 4. Kết luận, tập $Dep(X)$ sau cùng chính là cơ sở phụ thuộc cần tìm.

Có thể kiểm chứng được độ phức tạp thời gian của Thuật toán 4.17 là đa thức theo kích thước của M . Để ý, $Dep(X)$ chứa tối đa $|U|$ phần tử nên $Dep(X)$ chỉ làm mịn được nhiều nhất $|U \setminus X| - 1$ lần trong Bước 3. Lưu ý, $G(X)$ tính được trong Thuật toán 4.17 có thể không giống $G(X)$ theo định nghĩa, do đó $mdsb(G(X)) \subseteq G(X)$ không còn đúng. Tuy nhiên, cơ sở phân tách tối tiểu của chúng là như nhau.

Định lý 4.42. Thuật toán 4.17 tính đúng $Dep(X)$.

Chứng minh. Đầu tiên ta chứng minh thuật toán tính đúng $G(X)$. Thật vậy, rõ ràng với mọi $a \in X$ thì $\{a\} \in G(X)$. Trường hợp nếu tồn tại PTĐT $X' \rightarrow\rightarrow Y' \in M$ sao cho $X' \subseteq X$, thì vận dụng tính gia tăng cho

$X' \rightarrow\rightarrow Y' \in M$, ta thu được $X \rightarrow\rightarrow Y' \in M^+$. Ngoài ra, áp dụng tính bù cho $X' \rightarrow\rightarrow Y' \in M$, ta thu được $X' \rightarrow\rightarrow U \setminus (X' \cup Y') \in M^+$. Sau đó, vận dụng tính gia tăng thêm lần nữa cho $X' \rightarrow\rightarrow U \setminus (X' \cup Y') \in M^+$ ta thu được $X \rightarrow\rightarrow U \setminus (X' \cup Y') \in M^+$. Như vậy $G(X) = G(X) \cup \{Y', U \setminus (X' \cup Y')\}$.

Bây giờ từ PTĐT $Z \rightarrow\rightarrow W \in M$, vận dụng tính gia tăng ta thu được $Y \rightarrow\rightarrow W \in M^+$. Hơn nữa, theo Bổ đề 4.12 ta có $X \rightarrow\rightarrow Y \in M^+$. Từ đây áp dụng tính bắc cầu cho $X \rightarrow\rightarrow Y \in M^+$ và $Y \rightarrow\rightarrow W \in M^+$, ta thu được $X \rightarrow\rightarrow W \setminus Y \in M^+$.

Trường hợp tính $Dep(X)$ thì rõ ràng theo thuật toán. □

Ví dụ 4.43. Xét tập PTĐT $M = \{a \rightarrow\rightarrow bc, ei \rightarrow\rightarrow c\}$ trên tập thuộc tính $U = \{a, b, c, e, i\}$ và $X = \{a\}$.

Ta có $G(X) = \{\{a\}, \{b, c\}, \{e, i\}\}$. Đặt $Dep(X) = \{\{a\}, \{b, c\}, \{e, i\}\}$. Vận dụng tính bắc cầu cho PTĐT $ei \rightarrow\rightarrow c \in M^+$, ta thu được $a \rightarrow\rightarrow c \in M^+$. Suy ra

$$\begin{aligned} Dep(X) &= mdsb(\{\{a\}, \{b, c\}, \{e, i\}\} \cup \{c\}) \\ &= mdsb(\{\{a\}, \{b, c\}, \{e, i\}, \{c\}\}) \\ &= \{\{a\}, \{b\}, \{c\}, \{e, i\}\}. \end{aligned}$$

Không còn PTĐT nào trong M có thể làm mịn $Dep(X)$ được nữa. Như vậy

$$Dep(X) = \{\{a\}, \{b\}, \{c\}, \{e, i\}\}.$$

Tương tự như PTH, đối với PTĐT chúng ta cũng có *bài toán thành viên* như sau: Cho tập PTĐT M trên tập thuộc tính U và một PTĐT $X \rightarrow\rightarrow Y$. Xác định xem $X \rightarrow Y \in M^+$ hay không?

Sau đây là thuật toán thành viên giải bài toán thành viên.

Thuật toán 4.18 (Thuật toán thành viên đối với PTĐT)

Vào: Tập PTĐT M trên U và $X \rightarrow \rightarrow Y$.

Ra: Cho biết PTĐT $X \rightarrow \rightarrow Y$ có thuộc M^+ hay không?

Phương pháp:

Bước 1. Tính cơ sở phụ thuộc $Dep(X)$.

Bước 2. Nếu $Y = \bigcup Y_i$ với một số $Y_i \in Dep(X)$, thì kết luận $X \rightarrow \rightarrow Y \in M^+$. Ngược lại, kết luận $X \rightarrow \rightarrow Y \notin M^+$.

4.7.5. Chuẩn hóa lược đồ quan hệ về dạng chuẩn bốn

4.7.5.1. Tách không mất thông tin ứng với phụ thuộc hàm và phụ thuộc đa trị

Thuật toán 4.19 (Thuật toán kiểm tra tính không mất thông tin ứng với PTH và PTĐT)

Vào: LĐQH $S = (U, D = F \cup M)$ với $U = \{a_1, a_2, \dots, a_n\}$ và một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của S .

Ra: Cho biết ρ có mất thông tin hay không.

Phương pháp:

Bước 1. (*Lập bảng ban đầu*) Lập bảng ban đầu như Thuật toán 4.13.

Bước 2. (*Lập: biến đổi bảng*) Lần lượt xét các PTH và PTĐT trong D . Trường hợp nếu phụ thuộc là:

- PTH (thuộc F) thì thực hiện như Thuật toán 4.13.
- PTĐT $X \rightarrow \rightarrow Y \in M$ và có hai hàng bất kỳ t_1, t_2 trong bảng sao cho $t_1(X) = t_2(X)$, thì thêm một hàng t_3 thỏa mãn

$$\begin{cases} t_3(X) & = t_1(X) \\ t_3(Y) & = t_1(Y) \\ t_3(U \setminus (X \cup Y)) & = t_2(U \setminus (X \cup Y)) \end{cases}$$

nếu t_3 chưa có trong bảng.

Lặp lại quá trình biến đổi như trên cho đến khi không thể biến đổi bảng được nữa.

Bước 3. (*Kết luận*) Nếu trong bảng cuối cùng có chứa một hàng gồm $x_1, x_2, \dots, x_n$ thì phép tách ρ là không mất thông tin, ngược lại ρ là mất thông tin.

Lưu ý, khái niệm tách không mất thông tin ứng với PTH và PTĐT hoàn toàn tương tự như ứng với chỉ PTH. Do đó, chúng ta cũng có ngay thuật toán kiểm tra tính không mất thông tin ứng với PTH và PTĐT dựa vào các biến đổi trên bảng như trên. Ở đây tính đúng của Thuật toán 4.19 được dành làm bài tập. Trường hợp nếu phép tách ρ chỉ gồm hai thành phần, thì việc kiểm tra tính không mất thông tin đơn giản hơn qua kết quả sau. Chứng minh kết quả này là dễ dàng.

Định lý 4.43. Nếu $\rho = \{X_1, X_2\}$ là một phép tách của LĐQH $S = (U, D)$, thì ρ không mất thông tin khi và chỉ khi $X_1 \cap X_2 \rightarrow \rightarrow X_1 \setminus X_2 \in D$ hoặc $X_1 \cap X_2 \rightarrow \rightarrow X_2 \setminus X_1 \in D$.

Ví dụ 4.44. Xét LĐQH $S = (U, D)$ với tập thuộc tính $U = \{a, b, c, d, e, i\}$ và $D = \{a \rightarrow b, cd \rightarrow a, bc \rightarrow d, ae \rightarrow i, ce \rightarrow d, a \rightarrow \rightarrow cd\}$. Xét phép tách $\rho = \{\{a, d\}, \{a, b\}, \{b, e\}, \{c, d, e\}, \{a, e\}\}$ của S . Ta có bảng ban đầu:

	a	b	c	d	e	i
$\{a, c, d\}$	x_1	y_{12}	x_3	x_4	y_{15}	y_{16}
$\{a, b\}$	x_1	x_2	y_{23}	y_{24}	y_{25}	y_{26}
$\{a, e, i\}$	x_1	y_{32}	y_{33}	y_{34}	x_5	x_6

Lần lượt xét các PTH và PTĐT và thực hiện các biến đổi:

$a \rightarrow b$: thay y_{12} và y_{32} bằng x_2

$a \rightarrow \rightarrow cd$: có $t_1(a) = t_2(a)$, do đó thêm vào bảng một hàng mới t_4 thỏa mãn

$$\begin{cases} t_4(\{a\}) &= t_1(\{a\}) \\ t_4(\{c, d\}) &= t_1(\{c, d\}) \\ t_4(\{b, e, i\}) &= t_2(\{b, e, i\}) \end{cases}$$

Trong bảng chưa có một hàng nào gồm $x_1, x_2, \dots, x_5$ và các phụ thuộc vẫn còn làm thay đổi bảng:

$a \rightarrow \rightarrow cd$: có $t_1(a) = t_3(a)$ do đó thêm vào bảng một hàng

mới t_5 thỏa mãn

$$\begin{cases} t_5(\{a\}) &= t_1(\{a\}) \\ t_5(\{c, d\}) &= t_1(\{c, d\}) \\ t_5(\{b, e, i\}) &= t_3(\{b, e, i\}) \end{cases}$$

Khi đó, ta có bảng

	a	b	c	d	e	i
$\{a, c, d\}$	x_1	x_2	x_3	x_4	y_{15}	y_{16}
$\{a, b\}$	x_1	x_2	y_{23}	y_{24}	y_{15}	y_{26}
$\{a, e, i\}$	x_1	x_2	y_{33}	y_{34}	x_5	x_6
	x_1	x_2	x_3	x_4	y_{25}	y_{26}
	x_1	x_2	x_3	x_4	x_5	x_6

Hàng thứ 5 trong bảng cuối cùng gồm $x_1, x_2, \dots, x_5$. Do vậy, phép tách ρ là không mất thông tin.

4.7.5.2. *Dạng chuẩn bốn*

Sự tổng quát hoá của dạng chuẩn Boyce-Codd áp dụng cho các LĐQH có các PTĐT được gọi là dạng chuẩn bốn. LĐQH $S = (U, D = F \cup M)$ được gọi là ở *dạng chuẩn bốn* hay $4NF$ (ứng với D) nếu với mọi PTĐT không tầm thường $X \twoheadrightarrow Y \in D^+$ thì X là một khoá của S . Như vậy, từ định nghĩa suy ra nếu có một PTĐT không tầm thường $X \twoheadrightarrow Y \in D^+$ sao cho $X^+ \neq U$ thì S không phải $4NF$. Ta nói những PTĐT không tầm thường làm cho S không phải $4NF$ là *PTĐT vi phạm $4NF$* .

Lưu ý, khái niệm khóa và khóa tối tiểu không có gì thay đổi khi có PTĐT, nó vẫn chỉ phụ thuộc vào PTH và không phụ thuộc vào PTĐT.

Ví dụ 4.45. Xét LĐQH $S = (U, D)$ với tập thuộc tính $U = \{a, b, c\}$ và $D = \{a \twoheadrightarrow b\}$. Dễ thấy $Key(S) = \{U\}$, nên suy ra S không phải $4NF$. Tuy nhiên chúng ta biết S là BCNF.

Ví dụ 4.46. Xét LĐQH $S = (U, D)$ với tập thuộc tính $U = \{a, b, c\}$ và $D = \{a \rightarrow bc, a \twoheadrightarrow b\}$. Dễ tính được $Key(S) = \{\{a\}\}$. Suy ra S là

4NF.

Định lý 4.44. *Nếu LĐQH S là 4NF thì S cũng là BCNF.*

Chứng minh. Ta sẽ chứng minh nếu S là 4NF ứng với D , thì S cũng là BCNF ứng với tập PTH suy diễn từ D . Chứng minh bằng phản chứng. Giả sử S không phải BCNF. Suy ra, tồn tại PTH $X \rightarrow a \in F^+, a \notin X$ sao cho $X^+ \neq U$. Nên $X \cup \{a\} \subset U$ (bởi nếu không thì có nghĩa X là khóa, mâu thuẫn với giả sử). Áp dụng tính lặp lại (FM1), suy ra tồn tại PTĐT $X \rightarrow\rightarrow a \in M^+, a \notin X$ sao cho $X \cup \{a\} \neq U$. Vì S là 4NF, nên theo định nghĩa ta có X phải là một khóa của S . Điều này mâu thuẫn với giả sử. $\square$

Mệnh đề ngược lại của Định lý 4.44 là không đúng. Chẳng hạn, LĐQH S trong Ví dụ 4.45 là BCNF nhưng không phải 4NF. Như vậy, nếu xét thêm PTĐT nữa thì dạng chuẩn cao nhất của LĐQH là 4NF.

4.7.5.3. Tách không mất thông tin về dạng chuẩn bốn

Sau đây là thuật toán tách về dạng chuẩn bốn không mất thông tin.

Thuật toán 4.20 (Thuật toán tách thành 4NF)

Vào: LĐQH $S = (U, D = F \cup M)$.

Ra: Một phép tách $\rho = \{X_1, X_2, \dots, X_k\}$ của S thỏa các tính chất sau:

- ρ không mất thông tin ứng với D .
- Các thành phần $X_i \in \rho$ là 4NF ứng với $\Pi_{X_i}(D)$.

Phương pháp:

Bước 1. Đặt $\rho := \{U\}$.

Bước 2. Tìm một thành phần $X_i \in \rho$ không phải 4NF (ứng với $\Pi_{X_i}(D)$), tức là tồn tại một PTĐT không tầm thường $X \rightarrow\rightarrow Y$ trên X_i sao cho $X^+ \neq X_i$. Sau đó tách X_i thành hai thành phần con $X_1 = X \cup Y$ và $X_2 = X \cup Z$ với $Z = U \setminus (X \cup Y)$. Lúc này $\rho = \rho \setminus \{X_i\} \cup \{X_1, X_2\}$.

Lặp lại quá trình trên nếu X_1 hoặc X_2 không phải 4NF. Trường hợp nếu tất cả các thành phần của ρ đều là 4NF, thì thuật toán dừng. Kết luận ρ là phép tách cần tìm.

Vì U là hữu hạn, nên sau một số hữu hạn bước Thuật toán 4.20 sẽ dừng với các thành phần của ρ đều là 4NF. Bởi Định lý 4.43, phép tách ρ thu được là không mất thông tin ứng với D . Có thể kiểm chứng được độ phức tạp thời gian của Thuật toán 4.20 là hàm mũ theo số thuộc tính trong U . Lưu ý, để tính các chiếu $\Pi_{X_i}(D)$ chúng ta có thể dùng kết quả sau với chứng minh được dành làm bài tập.

Định lý 4.45. Cho LĐQH $S = (U, D)$ và $X_i \subseteq U$. Các chiếu $\Pi_{X_i}(D)$ được tính như sau:

- 1) Tính D^+ .
- 2) Với mỗi $X \rightarrow Y \in D^+$, nếu $X \subseteq X_i$ thì $X \rightarrow Y \cap X_i$ có trong X_i .
- 3) Với mỗi $X \twoheadrightarrow Y \in D^+$, $X \subseteq X_i$ thì $X \twoheadrightarrow Y \cap X_i$ có trong X_i .
- 4) Không còn PTH hay PTĐT nào khác của X_i có thể suy diễn được từ D thỏa trên S .

Ví dụ 4.47. Xét LĐQH $S = (U, D)$ với tập thuộc tính $U = \{a, b, c, d, e, i\}$, tập PTH và PTĐT $D = \{a \rightarrow b, cd \rightarrow a, bc \rightarrow d, ae \rightarrow i, ce \rightarrow d, a \twoheadrightarrow cd\}$.

Dễ dàng tính được ngay $Key(S) = \{\{c, e\}\}$. Vận dụng Thuật toán 4.20, đầu tiên ta đặt $\rho = \{U\}$. Ta có PTĐT không tầm thường $a \twoheadrightarrow cd$ vi phạm 4NF, nên tách U thành hai thành phần con $X_1 = \{a, c, d\}$ và $X_2 = \{a, b, e, i\}$. Lúc đó $\rho = \{X_1, X_2\}$. Dễ thấy X_1 là 4NF ứng với $\Pi_{X_1}(D)$. Tuy nhiên, X_2 không phải 4NF ứng với $\Pi_{X_2}(D)$, vì X_2 có khóa tối tiểu duy nhất là $\{a, e\}$ và PTĐT $a \twoheadrightarrow b \in D^+$ vi phạm 4NF. Do đó, ta tách tiếp X_2 thành hai thành phần con nữa $X_{21} = \{a, b\}$ và $X_{22} = \{a, e, i\}$. Các thành phần con này đều là 4NF. Như vậy, một phép tách thành 4NF của LĐQH S thỏa tính chất không mất thông tin là $\rho = \{\{a, c, d\}, \{a, b\}, \{a, e, i\}\}$. Để ý, nếu loại bỏ PTĐT $a \twoheadrightarrow cd$ trong tập D thì chưa chắc phép tách ρ thu được là thỏa tính chất không mất thông tin.

BÀI TẬP CHƯƠNG 4

4.1. Cho F là tập PTH trên tập thuộc tính U và $X, Y \subseteq U$. Chứng minh rằng $X \rightarrow Y \in F^+$ nếu và chỉ nếu $Y^+ \subseteq X^+$.

4.2. Cho tập PTH F trên tập thuộc tính U và $X, Y, Z \subseteq U$. Chứng minh các hệ tiên đề sau là tương đương với hệ tiên đề Armstrong ((F1)-(F3)):

a) Hệ tiên đề (C):

(C1) Nếu $X \rightarrow Y \in F^+$ và $Y \cup V \rightarrow Z \in F^+$ thì $X \cup V \rightarrow Z \in F^+$

(C2) $X \rightarrow X \in F^+$

(C3) Nếu $X \rightarrow Y \in F^+$ thì $X \cup V \rightarrow Y \in F^+$.

b) Hệ tiên đề (D):

(D1) Nếu $X \rightarrow Y \in F^+$ và $Y \rightarrow Z \in F^+$ thì $X \rightarrow Z \in F^+$

(D2) $X \rightarrow X \in F^+$

(D3) Nếu $X \rightarrow Y \in F^+$ thì $X \cup V \rightarrow Y \setminus Z \in F^+$

(D4) Nếu $X \rightarrow Y \in F^+$ và $V \rightarrow Z \in F^+$ thì $X \cup V \rightarrow Y \cup Z \in F^+$.

4.3. Cho tập PTH $F = \{a \rightarrow c, ab \rightarrow c, c \rightarrow di, cd \rightarrow i, ec \rightarrow ab, ei \rightarrow c\}$ trên tập thuộc tính $U = \{a, b, c, d, e, i\}$. Tìm một phủ không dư của F .

4.4. Tìm một phủ tối thiểu của tập PTH $F = \{ab \rightarrow c, c \rightarrow a, bc \rightarrow d, acd \rightarrow b, d \rightarrow eg, be \rightarrow c, cg \rightarrow bd, ce \rightarrow ag\}$ trên tập thuộc tính $U = \{a, b, c, d, e, g\}$.

4.5. Cho LĐQH $S = (U, F)$ và các tập con $X \subseteq M \subseteq U$. Chứng minh các điều kiện sau là tương đương:

a) $X^+ \cap M = X$

b) $X^+ \cap (M \setminus X) = \emptyset$

c) $M \setminus X^+ = M \setminus X$.

4.6. Xét LĐQH $S = (U, F)$ và $U_I = \bigcap Key(S), U_J = \bigcup Key(S)$. Chứng minh rằng nếu $X \subseteq U_I$ thì $X^+ \cap U_J = X$.

4.7. Cho LĐQH $S = (U, F)$. Chứng minh rằng

a) U_N là một tập đóng của S .

b) $X \subseteq U_N$ khi và chỉ khi $X^+ \subseteq U_N$.

4.8. Xét LĐQH $S = (U, F)$ và $U_I = \bigcap Key(S)$. Chứng minh rằng với mọi $X \subseteq U_N$, $Y \subseteq U_I$ thì $(X \cup Y)^+ \setminus U_I \subseteq U_N$.

4.9. Chứng minh nếu $|U_L \cap U_R| \leq 1$ thì LĐQH $S = (U, F)$ có một khóa tối tiểu duy nhất.

4.10. Giả sử $S = (U, F)$ là LĐQH có một khóa tối tiểu duy nhất. Chứng minh rằng S là BCNF khi và chỉ khi S là 3NF.

4.11. Cho LĐQH $S = (U, F)$. Với mọi $X \subseteq U$ gọi

$$Ex(X, F) = \{a \in X : X \setminus \{a\} \rightarrow a \notin F^+\}.$$

a) Chứng minh với mọi $K \in Key(S)$ ta có $Ex(U, F) \subseteq K$.

b) Có nhận xét gì về các tập $Ex(X, F)$ với $|X| = 1$.

c) $a \in U$ được gọi là *thuộc tính siêu cơ bản* của S nếu a thuộc tất cả khóa tối tiểu của S . Chứng minh a là thuộc tính siêu cơ bản của S khi và chỉ khi $a \in Ex(U, F)$.

4.12. Cho tập thuộc tính $U = \{a, b, c, d, e\}$. Xây dựng một tập PTH F trên U để LĐQH $S = (U, F)$ thỏa các điều kiện sau:

a) S có đúng hai khóa tối tiểu.

b) S là 2NF.

c) S không phải 3NF.

4.13. Cho LĐQH $S = (U, F)$, với $U = \{a, b, c, d\}$ và $F = \{a \rightarrow c, d \rightarrow b, c \rightarrow abd\}$.

a) Tìm một khóa tối tiểu của S .

b) Xác định dạng chuẩn cao nhất của S .

c) S có một khóa tối tiểu duy nhất hay không? Nếu không hãy loại bỏ khỏi F một PTH để S có đúng một khóa tối tiểu. Giải thích cách làm.

- 4.14. Cho một tập PTH và PTĐT $D = F \cup M$. Tìm một ví dụ chỉ ra tồn tại một PTH $X \rightarrow Y$ được suy diễn từ D bằng hệ tiên đề Armstrong-BFH (cụ thể từ tính liên kết (FM2)) nhưng $X \rightarrow Y$ không được suy diễn từ F bằng hệ tiên đề Armstrong.
- 4.15. Chứng minh tính đủ của hệ tiên đề BFH.
- 4.16. Chứng minh tính đủ của hệ tiên đề Armstrong-BFH (Định lý 4.41).
- 4.17. Chứng minh một LDQH có một PTĐT duy nhất là BCNF.
- 4.18. Cho $X, Y \subseteq U$. Chứng minh rằng PTĐT $X \rightarrow\rightarrow Y$ là tầm thường trên U nếu và chỉ nếu bất kỳ quan hệ $R \in Rel(U)$ thì $R(X \rightarrow\rightarrow Y)$.
- 4.19. Chứng minh Định lý 4.45.
- 4.20. Chứng minh tính bắc cầu của PTH (nếu $X \rightarrow Y, Y \rightarrow Z$ thì $X \rightarrow Z$) không còn đúng nữa cho PTĐT khi hai tập Y và Z là không phân biệt.

Chương 5.

NHỮNG VẤN ĐỀ LIÊN QUAN ĐẾN THIẾT KẾ

5.1. Chuyển dịch lược đồ quan hệ

Như chúng ta đã biết, các đối tượng như bao đóng, khóa và phản khóa của LĐQH đóng một vai trò rất quan trọng trong lý thuyết thiết kế CSDL. Trong khi đó, rõ ràng nếu các LĐQH ban đầu có kích thước “gọn nhỏ” hơn, chẳng hạn tập thuộc tính ít hơn hay về trái và về phải của các PTH có ít thuộc tính hơn, thì việc tính toán trên các đối tượng bao đóng, khóa và phản khóa của LĐQH sẽ nhanh hơn. Một trong những cách để LĐQH ban đầu gọn nhỏ hơn là chúng ta chuyển dịch LĐQH ban đầu về một LĐQH với tập thuộc tính ít hơn, về trái và về phải của các PTH có ít thuộc tính hơn. Những thuộc tính bỏ đi trong LĐQH này không quan trọng theo nghĩa chúng không làm ảnh hưởng đến kết quả tính toán trên các đối tượng bao đóng, khóa và phản khóa của LĐQH ban đầu. Các đối tượng này sẽ thu được bằng cách thêm hay bớt một số thuộc tính tương ứng trên LĐQH gọn nhỏ hơn ban đầu.

5.1.1. Định nghĩa

Xét LĐQH $S = (U, F)$ và tập thuộc tính $T \subseteq U$. LĐQH $\tilde{S}_T = (\tilde{U}_T, \tilde{F}_T)$ trong đó $\tilde{U}_T = U \setminus T$ và

$$\tilde{F}_T = \{X \setminus T \rightarrow Y \setminus T : X \rightarrow Y \in F, Y \setminus T \neq \emptyset\}$$

được gọi là *LĐQH chuyển dịch của S theo T*. Trường hợp nếu không sợ nhầm lẫn về tập T , ta sẽ ký hiệu lại LĐQH $\tilde{S}_T = (\tilde{U}_T, \tilde{F}_T)$ bằng

$\tilde{S} = (\tilde{U}, \tilde{F})$ và gọi là *LĐQH chuyển dịch* của S . Dễ thấy, thời gian xác định LĐQH chuyển dịch $\tilde{S}$ là tuyến tính theo $|U|$ và $|F|$.

Ví dụ 5.1. Xét LĐQH $S = (U, F)$ với tập thuộc tính $U = \{a, b, c, d, e\}$ và tập PTH $F = \{ab \rightarrow e, c \rightarrow d, be \rightarrow ad, d \rightarrow ae\}$. Khi đó, nếu $T = \{a, b, e\}$ thì LĐQH chuyển dịch theo T sẽ là $\tilde{S} = (\tilde{U}, \tilde{F})$ với $\tilde{U} = \{c, d\}$ và $\tilde{F} = \{c \rightarrow d, \emptyset \rightarrow d\}$.

Trong LĐQH chuyển dịch trên, chúng ta chấp nhận các PTH có vế trái bằng rỗng. Điều này sẽ giúp chúng ta thu được nhiều tính chất quan trọng về bao đóng và khóa của LĐQH ban đầu.

5.1.2. Bao đóng qua phép chuyển dịch lược đồ quan hệ

Xét LĐQH $S = (U, F)$ và LĐQH chuyển dịch $\tilde{S} = (\tilde{U}, \tilde{F})$ theo T . Trong mục này, để tránh nhầm lẫn bao đóng của tập thuộc tính X ứng với F chúng ta sẽ viết rõ X_F^+ thay vì X^+ . Đầu tiên, chúng ta có biểu diễn của bao đóng tập thuộc tính theo LĐQH chuyển dịch.

Mệnh đề 5.1. Nếu T và X là hai tập con thuộc tính rời nhau trong LĐQH S , thì

$$(T \cup X)_F^+ = T \cup X_F^+.$$

Chứng minh. Trước hết để ý rằng với giả thiết $T \cap X = \emptyset$ và định nghĩa của LĐQH chuyển dịch $\tilde{S}$, thì $T \cap X_F^+ = \emptyset$. Ta sẽ chứng minh mệnh đề bằng quy nạp theo số bước xây dựng các dãy tập thuộc tính $(T \cup X)^{(k)}$ và $X^{(k)}$ ($k = 0, 1, 2, \dots$) theo thuật toán tính bao đóng của các tập thuộc tính $(T \cup X)$ và X tương ứng với các tập PTH F và $\tilde{F}$. Cụ thể chúng ta sẽ chứng minh bất biến

$$(T \cup X)^{(k)} = T \cup X^{(k)}, \quad k = 0, 1, 2, \dots$$

Thật vậy, với $k = 0$, ta có $(T \cup X)^{(0)} = T \cup X$ và $X^{(0)} = X$. Khi đó, rõ ràng

$$(T \cup X)^{(0)} = T \cup X = T \cup X^{(0)}.$$

Bây giờ giả sử với $k \geq 0$, ta có $(T \cup X)^{(k)} = T \cup X^{(k)}$. Chúng ta cần chứng minh $(T \cup X)^{(k+1)} = T \cup X^{(k+1)}$. Đó là ta cần chứng minh khi chuyển từ bước k sang bước $k+1$, hai tập thuộc tính $(T \cup X)^{(k)}$ và $T \cup X^{(k)}$ sẽ được bổ sung thêm cùng một số thuộc tính. Đầu tiên, để ý rằng do $T \cap X = \emptyset$ và T không có mặt trong LĐQH chuyển dịch $\tilde{S} = (\tilde{U}, \tilde{F})$, nên với mọi $k \geq 0$ ta luôn có $T \cap X^{(0)} = \emptyset$. Ngoài ra, do dãy $(T \cup X)^{(k)}$ đơn điệu không giảm và $(T \cup X)^{(0)} = T \cup X \supseteq T$, nên với mọi $k \geq 0$ ta luôn có $(T \cup X)^{(k)} \supseteq T$. Từ các nhận xét này và giả thiết quy nạp, suy ra

$$\forall Y \subseteq U : Y \subseteq (T \cup X)^{(k)} = T \cup X^{(k)} \Rightarrow Y \setminus T \subseteq X^{(k)}$$

và

$$\forall Z \subseteq U \setminus T : Z \subseteq X^{(k)} \Rightarrow T \cup Z \subseteq T \cup X^{(k)} = (T \cup X)^{(k)}.$$

Xét các PTH $Y \rightarrow M \in F$ sao cho $Y \subseteq (T \cup X)^{(k)}$. Khi đó, $Y \setminus T \rightarrow M \setminus T \in \tilde{F}$ và $Y \setminus T \subseteq X^{(k)}$. Từ giả thiết quy nạp, ta suy ra

$$\begin{aligned} M \cup (T \cup X)^{(k)} &= M \cup (T \cup X^{(k)}) \\ &= (M \setminus T) \cup (T \cup X^{(k)}). \end{aligned}$$

Ngược lại, xét các PTH $Z \rightarrow N \in \tilde{F}$ và $Z \subseteq X^{(k)}$. Theo định nghĩa chuyển dịch LĐQH theo T , phải tồn tại PTH $Y \rightarrow M \in F$ sao cho $Z = Y \setminus T$ và $N = M \setminus T$. Khi đó, ta có $Y \subseteq T \cup Z \subseteq T \cup X^{(k)} = (T \cup X)^{(k)}$, và do đó

$$\begin{aligned} M \cup (T \cup X)^{(k)} &= M \cup (T \cup X^{(k)}) \\ &= (M \setminus T) \cup (T \cup X^{(k)}) \\ &= N \cup (T \cup X)^{(k)}. \end{aligned}$$

□

Từ Mệnh đề 5.1, chúng ta thu được công thức tính bao đóng tập thuộc tính qua phép chuyển dịch LĐQH như sau.

Hệ quả 5.1. Với mọi tập con thuộc tính T trong LĐQH S , thì

$$T_F^+ = T \cup \emptyset_F^+.$$

Ví dụ 5.2. Xét LĐQH $S = (U, F)$ với tập thuộc tính $U = \{a, b, c, d, e\}$ và tập PTH $F = \{ab \rightarrow e, c \rightarrow d, be \rightarrow ad, d \rightarrow ae\}$. Bây giờ, chúng ta sẽ tính bao đóng T_F^+ với $T = \{a, b, e\}$ qua phép dịch chuyển LĐQH theo T . Trong Ví dụ 5.1 ở trên chúng ta đã biết $\tilde{F}_T = \{c \rightarrow d, \emptyset \rightarrow d\}$. Khi đó $\emptyset_{\tilde{F}_T}^+ = \{d\}$, suy ra $T_F^+ = \{a, b, e\} \cup \{d\} = \{a, b, d, e\}$. Vậy $T_F^+ = \{a, b, d, e\}$. Trường hợp nếu $T = \{b, c\}$, thì $\tilde{F}_T = \{a \rightarrow e, \emptyset \rightarrow d, e \rightarrow ad, d \rightarrow ae\}$. Khi đó, $\emptyset_{\tilde{F}_T}^+ = \{d, a, e\}$ và $T_F^+ = U$.

5.1.3. Khóa qua phép chuyển dịch lược đồ quan hệ

Xét LĐQH $S = (U, F)$ và LĐQH chuyển dịch $\tilde{S} = (\tilde{U}, \tilde{F})$ theo T . Trước hết, chúng ta có một số bổ đề cơ sở về mối quan hệ giữa khóa, khóa tối tiểu của S và $\tilde{S}$.

Bổ đề 5.1.

- 1) Nếu K là một khóa của S thì $K \setminus T$ là một khóa của $\tilde{S}$.
- 2) Nếu K' là một khóa của $\tilde{S}$ thì $T \cup K'$ là một khóa của S .

Chứng minh.

1) Giả sử K là một khóa của S . Đặt $K' = K \setminus T$. Khi đó, ta có $T \cap K' = \emptyset$ và $K \subseteq T \cup K'$. Vận dụng tính đơn điệu của bao đóng và Mệnh đề 5.1, suy ra

$$T \cup \tilde{U} = U = K_F^+ \subseteq (T \cup K')_F^+ = T \cup (K')_{\tilde{F}}^+.$$

Do đó $T \cup \tilde{U} \subseteq T \cup (K')_{\tilde{F}}^+$. Hơn nữa, $T \cap \tilde{U} = \emptyset$ và $T \cap (K')_{\tilde{F}}^+ = \emptyset$. Suy ra $\tilde{U} \subseteq (K')_{\tilde{F}}^+$. Như vậy, $(K')_{\tilde{F}}^+ = \tilde{U}$.

2) Giả sử K' là một khóa của $\tilde{S}$. Suy ra $K' \cap T = \emptyset$ và $(K')_{\tilde{F}}^+ = \tilde{U}$. Đặt $K = T \cup K'$. Theo Mệnh đề 5.1, ta có

$$K_F^+ = (T \cup K')^+ = T \cup (K')_F^+ = T \cup \tilde{U} = U.$$

□

Trường hợp nếu K' là một khóa của $\tilde{S}$ và $T \subseteq U_N$, thì lúc đó rõ ràng K' cũng là một khóa của S . Bây giờ, ta xét LDQH chuyển dịch $\tilde{S} = (\tilde{U}, \tilde{F})$ theo $T = X^+$ với $X \subseteq U$. Vận dụng tính chất 5) trong Mệnh đề 4.5, lúc này nếu K' là một khóa của $\tilde{S}$, thì theo Bổ đề 5.1 ta thu được

$$(X \cup K')^+ = (X^+ \cup K')^+ = (T \cup K')^+ = U.$$

Điều này có nghĩa, lúc này $X \cup K'$ cũng là một khóa của S . Để ý, kết quả này không hoàn toàn tương tự như bổ đề trên. Điểm khác nhau chính là lượng dịch chuyển ở đây là $T = X^+$ thay vì $T \subseteq X^+$.

Bổ đề 5.2. *Nếu tập thuộc tính dịch chuyển $T \subseteq U_N$, thì*

$$Key(S) = Key(\tilde{S}).$$

Chứng minh. Giả sử K là một khóa tối tiểu của S . Suy ra, K cũng là một khóa của S . Vì $T \subseteq U_N$, nên $K \cap T = \emptyset$. Khi đó, theo Bổ đề 5.1 ta có $K = K \setminus T$ cũng là một khóa của $\tilde{S}$. Bây giờ, giả sử K chứa một khóa K' của $\tilde{S}$. Theo nhận xét trên, suy ra K' cũng là một khóa của S . Theo định nghĩa khóa tối tiểu, rõ ràng $K = K'$. Vậy K phải là một khóa tối tiểu của $\tilde{S}$.

Ngược lại, nếu K là một khóa tối tiểu của $\tilde{S}$ thì rõ ràng $K \cap T = \emptyset$, và khi đó K là một khóa của S . Lúc này, giả sử K chứa một khóa K' của S . Suy ra $K' \cap T = \emptyset$. Theo Bổ đề 5.1, ta có $K' = K' \setminus T$ là một khóa của $\tilde{S}$. Theo định nghĩa khóa tối tiểu, suy ra $K = K'$. Vậy, K phải là một khóa tối tiểu của S . □

Bây giờ xét V là một tập hợp hữu hạn bất kỳ, tập con $X \subseteq V$ và họ $\mathcal{T} \subseteq \mathcal{P}(V)$. Ta định nghĩa phép toán $\oplus$ như sau

$$X \oplus \mathcal{T} = \{X \cup Y : Y \in \mathcal{T}\}.$$

Ở đây nhắc lại U_I là tập ký hiệu giao tất cả khóa tối tiểu của LDQH S . Lúc này từ các bổ đề cơ sở trên, chúng ta thu được các biểu diễn sau đây của khóa tối tiểu.

Định lý 5.2.

- 1) $Key(S) = Key(\tilde{S})$ khi và chỉ khi $T \subseteq U_N$.
- 2) $Key(S) = T \oplus Key(\tilde{S})$ khi và chỉ khi $T \subseteq U_I$.

Chứng minh.

- 1) Theo Bổ đề 5.2, nếu $T \subseteq U_N$ thì $Key(S) = Key(\tilde{S})$.

Bây giờ, giả sử $Key(S) = Key(\tilde{S})$ và $a \in T$ nhưng $a \notin U_N$. Suy ra $a \in \bigcup Key(S)$, hay tồn tại một khóa tối tiểu $K \in Key(S)$ để $a \in K$. Lúc này, ta cũng có $K \in Key(\tilde{S})$, và do đó $K \subseteq \tilde{U} = U \setminus T$, hay $K \cap T = \emptyset$. Rõ ràng, điều này mâu thuẫn với $a \in T \cap K$. Như vậy, $a \in U_N$ hay $T \subseteq U_N$.

- 2) Nếu $Key(S) = T \oplus Key(\tilde{S})$, thì có nghĩa T xuất hiện trong mọi khóa tối tiểu của S , và lúc đó $T \subseteq U_I$.

Bây giờ, giả sử $T \subseteq U_I$. Ta sẽ chứng minh mọi khóa tối tiểu $K \in Key(S)$ đều được biểu diễn dưới dạng $T \cup K'$ trong đó $K' \in Key(\tilde{S})$, và ngược lại mọi khóa tối tiểu $K' \in Key(\tilde{S})$ thì $T \cup K' \in Key(S)$.

Lấy bất kỳ $K \in Key(S)$. Suy ra $T \subseteq K$. Đặt $K' = K \setminus T$, khi đó ta có $K' \cap T = \emptyset$ và $K = T \cup K'$. Theo Bổ đề 5.1, suy ra $K' = K \setminus T$ là một khóa của $\tilde{S}$. Giả sử K' chứa một khóa M của $\tilde{S}$. Khi đó, $T \cup M$ lại là một khóa của S và $T \cup M \subseteq K$. Theo định nghĩa khóa tối tiểu, suy ra $T \cup M = K = T \cup K'$. Vì $T \cap K' = \emptyset$, nên $T \cap M = \emptyset$, và do đó $M = K'$. Vậy, $K' \in Key(\tilde{S})$.

Ngược lại, lấy bất kỳ $K' \in Key(\tilde{S})$. Lúc đó $T \cap K' = \emptyset$. Theo Bổ

đề 5.1, suy ra $T \cup K'$ là một khóa của S . Giả sử tồn tại một $K \in Key(S)$ sao cho $K \subseteq T \cup K'$. Lại theo Bổ đề 5.1, suy ra $K \setminus T$ là một khóa của $\tilde{S}$. Vì $T \subseteq U_I$, nên $T \subseteq K$. Do đó $K \setminus T \subseteq K'$. Áp dụng tính tối tiểu của khóa tối tiểu K' , ta suy ra $K' = K \setminus T$ hay $K = T \cup K'$. $\square$

Ví dụ 5.3. Xét LĐQH $S = (U, F)$ với tập thuộc tính $U = \{a, b, c\}$ và tập PTH $F = \{a \rightarrow b, b \rightarrow c\}$. Dễ thấy giao các khóa tối tiểu $U_I = U \setminus R = \{a\}$. Xét LĐQH chuyển dịch $\tilde{S} = (\tilde{U}, \tilde{F})$ theo $T = U_I$ ta được $\tilde{U} = \{b, c\}$ và $\tilde{F} = \{\emptyset \rightarrow b, b \rightarrow c\}$. Dễ dàng tính được $Key(\tilde{S}) = \{\emptyset\}$. Do vậy $Key(S) = T \oplus Key(\tilde{S}) = \{\{a\}\}$.

Trường hợp ta tính được

$$U_N = \bigcap \max(Closed(S) \setminus \{U\}) = \bigcap \max(\{\emptyset, \{b, c\}, \{c\}\}) = \{b, c\}.$$

Khi đó, xét LĐQH chuyển dịch $\tilde{S} = (\tilde{U}, \tilde{F})$ theo $T = U_N$, ta thu được $\tilde{U} = \{a\}$ và $\tilde{F} = \emptyset$. Suy ra ngay $Key(\tilde{S}) = \{\{a\}\}$, và do đó $Key(S) = Key(\tilde{S}) = \{\{a\}\}$.

Hệ quả 5.2. Nếu phép dịch chuyển theo $T = X \cup Y$, trong đó $X \subseteq U_N$ và $Y \subseteq U_I$, thì $Key(S) = Y \oplus Key(\tilde{S})$.

Bây giờ, xét LĐQH chuyển dịch của S như sau: $\tilde{S} = (\tilde{U}, \tilde{F})$ với

$$\tilde{U} = (U_L \cap U_R) \setminus (U_L \setminus U_R)^+$$

và

$$\tilde{F} = \{X \cap \tilde{U} \rightarrow Y \cap \tilde{U} : X \rightarrow Y \in F, X \cap \tilde{U} \neq \emptyset, Y \cap \tilde{U} \neq \emptyset\}.$$

Lúc này, chúng ta có các bổ đề cơ sở sau.

Bổ đề 5.3. Với mọi thuộc tính $a \in \tilde{U}$, tồn tại PTH $X' \rightarrow Y' \in \tilde{F}$ sao cho $a \in Y'$.

Bổ đề 5.4. Nếu $X_F^+ = U$ thì $(U_L \setminus U_R) \cup X \rightarrow \tilde{U} \in F^+$.

Bổ đề 5.5. Với mọi thuộc tính $a \in \tilde{U}$, PTH $X \rightarrow Y \in F$ sao cho $a \in Y$ thì $X \cap \tilde{U} \neq \emptyset$ và $X \cap \tilde{U} \rightarrow Y \cap \tilde{U} \in \tilde{F}$.

Bổ đề 5.6. K là khóa của $\tilde{S}$ khi và chỉ khi $K \cup (U \setminus U_R)$ là khóa của S .

Chứng minh các Bổ đề 5.3-5.6 người đọc có thể tham khảo trong [15] (2002). Cũng trong [15], từ các Bổ đề 5.3-5.6, chúng ta thu được tương quan về khóa tối tiểu như sau.

Định lý 5.3. K là khóa tối tiểu của $\tilde{S}$ khi và chỉ khi $K \cup (U \setminus U_R)$ là khóa tối tiểu của S .

Chứng minh. Giả sử $K \in \text{Key}(\tilde{S})$. Theo Bổ đề 5.6, suy ra $K \cup (U \setminus U_R)$ là một khóa của S . Rõ ràng, nếu $K \cup (U \setminus U_R) \notin \text{Key}(S)$ thì tồn tại một tập con thuộc tính $X \subset K \cup (U \setminus U_R)$ sao cho $X \in \text{Key}(S)$. Ở đây, $X = K' \cup (U \setminus U_R)$ với $K' \subset K$. Cũng theo Bổ đề 5.6, suy ra K' là một khóa của $\tilde{S}$. Điều này mâu thuẫn với giả thiết K là một khóa tối tiểu của $\tilde{S}$.

Ngược lại, giả sử $K \cup (U \setminus U_R) \in \text{Key}(S)$. Theo Bổ đề 5.6, suy ra K là một khóa của $\tilde{S}$. Trường hợp nếu $K \notin \text{Key}(\tilde{S})$, thì phải tồn tại một tập con thuộc tính $X \subset K$ sao cho $X \in \text{Key}(\tilde{S})$. Xét $K' = X \cup (U \setminus U_R)$. Rõ ràng K' là một khóa của S . Vì $X \subset K$, nên $K' \subset K \cup (U \setminus U_R)$. Điều này trái với giả thiết $K \cup (U \setminus U_R)$ là một khóa tối tiểu của S . $\square$

Rõ ràng, vận dụng Định lý 5.3 vào Thuật toán 4.11, chúng ta sẽ có ngay thuật toán tìm một khóa tối tiểu với độ phức tạp thời gian tốt hơn [15].

5.2. Các mô tả tương đương của phụ thuộc hàm

Mục này sẽ giới thiệu tổng quan về một số công cụ toán học tương đương với PTH như toán tử bao đóng, hệ bao đóng và hàm chọn. Các công cụ này giúp chúng ta có thể hiểu và nghiên cứu sâu hơn nữa về cấu trúc logic của PTH. Ngoài ra, nếu người đọc muốn tìm hiểu sâu về

khía cạnh đại số, tổ hợp cũng như các ứng dụng của toán tử bao đóng và hàm chọn thì có thể tham khảo thêm các công trình, chẳng hạn như [4, 10, 13, 18, 19]. Trong khuôn khổ CSDL, giáo trình không thể trình bày sâu theo các hướng này.

5.2.1. Toán tử bao đóng

Xét tập hữu hạn bất kỳ V . Ánh xạ $L: \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ thỏa các tính chất sau:

- L1) *tính phản xạ*: $X \subseteq L(X)$,
- L2) *tính đơn điệu*: nếu $X \subseteq Y$ thì $L(X) \subseteq L(Y)$,
- L3) *tính lũy đẳng*: $L(L(X)) = L(X)$,

với mọi $X, Y \subseteq V$, được gọi là *toán tử bao đóng* (TTBĐ) trên V . Ký hiệu $Closure(V)$ là tập tất cả các TTBĐ trên V .

Dễ dàng thấy bao đóng tập thuộc tính, bao đóng tập PTH và ánh xạ chiếu-kết nối là các TTBĐ.

Ví dụ 5.4. Sau đây là một số TTBĐ cơ bản:

- 1) TTBĐ *tối đại*: $\Omega(X) = V$ với mọi $X \subseteq V$.
- 2) TTBĐ *đồng nhất*: $i(X) = X$ với mọi $X \subseteq V$.
- 3) TTBĐ *tịnh tiến*: $h_T(X) = T \cup X$ với mọi $X \subseteq V$ và T là tập con cố định tùy ý cho trước của V .

TTBĐ tịnh tiến có mối liên hệ với các TTBĐ tối đại và đồng nhất như sau:

- Nếu $T = V$ thì TTBĐ tịnh tiến theo T trở thành TTBĐ tối đại.
- Nếu $T = \emptyset$ thì TTBĐ tịnh tiến theo T trở thành TTBĐ đồng nhất.

Như vậy, chúng ta có thể dùng TTBĐ tịnh tiến h_T làm cơ sở đặc tả cho họ các TTBĐ $\{\Omega, i, h_T\}$.

TTBĐ có một số tính chất cơ bản sau.

Mệnh đề 5.4. Giả sử $L \in Closure(V)$. Khi đó với mọi $X, Y \subseteq V$ ta có:

$$1) L(L(X) \cup Y) = L(X \cup L(Y)) = L(X \cup Y).$$

$$2) L(X \cup Y) \supseteq L(X) \cup L(Y).$$

$$3) L(X \cap Y) \subseteq L(X) \cap L(Y).$$

Chứng minh.

1) Ta chỉ cần chứng minh $L(L(X) \cup Y) = L(X \cup Y)$, sau đó hoán đổi vai trò của X và Y ta sẽ thu được điều cần chứng minh. Thật vậy, theo tính phản xạ và tính đơn điệu của f ta có

$$\begin{aligned} L(X) &\supseteq X \\ L(X) \cup Y &\supseteq X \cup Y \\ L(L(X) \cup Y) &\supseteq L(X \cup Y). \end{aligned}$$

Mặt khác, từ tính phản xạ, tính đơn điệu và tính lũy đẳng của L ta cũng có

$$\begin{aligned} \left\{ \begin{array}{l} X \subseteq X \cup Y \\ Y \subseteq X \cup Y \subseteq L(X \cup Y) \end{array} \right. &\Rightarrow \left\{ \begin{array}{l} L(X) \subseteq L(X \cup Y) \\ Y \subseteq L(X \cup Y) \end{array} \right. \\ &\Rightarrow L(X) \cup Y \subseteq L(X \cup Y) \\ &\Rightarrow L(L(X) \cup Y) \subseteq L(L(X \cup Y)) \\ &\Rightarrow L(L(X) \cup Y) = L(X \cup Y). \end{aligned}$$

Như vậy $(L(X) \cup Y) = L(X \cup Y)$.

2) Theo tính đơn điệu của L , ta có

$$\left\{ \begin{array}{l} L(X \cup Y) \supseteq L(X) \\ L(X \cup Y) \supseteq L(Y) \end{array} \right. \Rightarrow L(X \cup Y) \supseteq L(X) \cup L(Y).$$

3) Tương tự (2), theo tính đơn điệu của L , chúng ta cũng có

$$\begin{cases} L(X \cap Y) \subseteq L(X) \\ L(X \cap Y) \subseteq L(Y) \end{cases} \Rightarrow L(X \cap Y) \subseteq L(X) \cap L(Y).$$

□

Từ Mệnh đề 5.4, suy ra $L(L(X) \cup L(Y)) = L(X \cup Y)$.

Ví dụ sau đây cho thấy các bao hàm thức ngược lại của các tính chất (2) và (3) trong Mệnh đề 5.4 là không đúng.

Ví dụ 5.5. Giả sử $V = \{a, b, c\}$ và ánh xạ $L = L_{ab,c} : 2^V \longrightarrow 2^V$ xác định bởi:

$$L(X) = \begin{cases} X \cup \{c\}, & \text{nếu } \{a, b\} \subseteq X \\ X, & \text{ngược lại.} \end{cases}$$

Dễ dàng kiểm chứng được $L \in \text{Closure}(V)$. Khi đó, với $X = \{a\}$ và $Y = \{b\}$ thì ta có:

$$L(X \cup Y) = L(\{a, b\}) = \{a, b, c\}.$$

$$L(X) = L(\{a\}) = \{a\}.$$

$$L(Y) = L(\{b\}) = \{b\}.$$

Suy ra $L(X) \cup L(Y) = \{a, b\}$. Do đó $L(X \cup Y) \neq L(X) \cup L(Y)$.

Trường hợp với $X = \{a, b\}$ và $Y = \{b, c\}$, thì ta có:

$$L(X \cap Y) = L(\{b\}) = \{b\}.$$

$$L(X) = L(\{a, b\}) = \{a, b, c\}.$$

$$L(Y) = L(\{a, c\}) = \{a, c\}.$$

Suy ra $L(X) \cap L(Y) = \{a, c\}$. Như vậy, $L(X \cap Y) \neq L(X) \cap L(Y)$.

Bây giờ sẽ chúng ta tìm hiểu mối quan hệ giữa các TTBD với các PTH. Xét tập thuộc tính U và một tập F các PTH trên U . Tập F được gọi là một f -họ trên U nếu F thỏa tính phản xạ chặt, tính bắc cầu, tính mở rộng trái và thu hẹp phải, và tính cộng tính đầy đủ. Dễ dàng thấy $FD(R)$ và F^+ là các f -họ trên U .

Định lý 5.5. Nếu F là một f -họ trên U , thì ánh xạ L_F xác định bởi

$$L_F(X) = \{a \in U : X \rightarrow a \in F\}$$

là một TTBD trên U . Ngược lại, nếu L là một TTBD trên U thì tồn tại duy nhất một f -họ F trên U sao cho $L_F = L$, ở đây F được xác định:

$$F = \{X \rightarrow Y : X, Y \subseteq U, Y \subseteq L(X)\}.$$

Chứng minh. Giả sử F là một f -họ trên U . Ta sẽ chứng minh $L_F \in \text{Closure}(U)$. Vì tính phản xạ chặt của F , nên rõ ràng $X \subseteq L_F(X)$ với mọi $X \subseteq U$. Tức là L_F thỏa tính phản xạ. Trường hợp nếu $X \subseteq Y \subseteq U$ và $a \in L_F(X)$, thì suy ra $Y \rightarrow X \in F, X \rightarrow a \in F$, và do đó theo tính bắc cầu của F ta có $Y \rightarrow a \in F$, hay $a \in L_F(Y)$. Điều này có nghĩa, L_F thỏa tính đơn điệu. Từ tính phản xạ và tính đơn điệu của L_F , ta có ngay $L_F(X) \subseteq L_F(L_F(X))$ với mọi $X \subseteq U$. Bao hàm thức ngược lại được lập luận như sau. Theo định nghĩa, nếu $a \in L_F(L_F(X))$ thì $L_F(X) \rightarrow a \in F$. Ngoài ra, theo tính cộng đầy đủ của F ta có $X \rightarrow L_F(X) \in F$, và do đó theo tính bắc cầu của F , suy ra $X \rightarrow a \in F$ hay $a \in L_F(X)$.

Ngược lại, giả sử $L_F \in \text{Closure}(U)$. Đầu tiên ta sẽ chứng minh

$$F = \{X \rightarrow Y : X, Y \subseteq U, Y \subseteq L(X)\}$$

là một f -họ trên U . Vì tính phản xạ của L nên $X \rightarrow X \in F$, tức là F thỏa tính phản xạ chặt. Giả sử $X \rightarrow Y \in F$ và $Y \rightarrow Z \in F$. Suy ra

$$\begin{aligned} \begin{cases} Y \subseteq L(X) \\ Z \subseteq L(Y) \end{cases} &\Rightarrow \begin{cases} L(Y) \subseteq L(X) \\ Z \subseteq L(Y) \end{cases} \\ &\Rightarrow Z \subseteq L(X) \\ &\Rightarrow X \rightarrow Z \in F. \end{aligned}$$

Hay F thỏa tính bắc cầu. Trường hợp nếu $X \rightarrow Y \in F$ thì $Y \subseteq L(X)$. Hơn nữa $L(X) \subseteq L(X \cup W)$. Suy ra $Y \setminus Z \subseteq L(X \cup W)$, hay $X \cup W \rightarrow Y \setminus Z \in F$. Như vậy, F thỏa tính mở rộng trái và thu hẹp phải. Bây giờ nếu $X \rightarrow Y \in F$ và $W \rightarrow Z \in F$, thì suy ra

$$\begin{aligned} Y \cup Z &\subseteq L(X) \cup L(W) \\ &\subseteq L(X \cup W). \end{aligned}$$

Như vậy $X \cup W \rightarrow Y \cup Z \in F$. Do đó, F thỏa tính cộng tính đầy đủ. Tóm lại, F là một f -họ trên U .

Tiếp theo chúng ta sẽ chứng minh $L_F = L$. Theo định nghĩa L_F , nếu $a \in L_F(X)$ với $X \subseteq U$ thì $X \rightarrow a \in F$, và do đó $a \in L(X)$. Ngược lại, nếu $a \in L(X)$ thì từ định nghĩa của F , ta suy ra $X \rightarrow a \in F$, hay $a \in L_F(X)$.

Cuối cùng, chúng ta chứng minh F là f -họ duy nhất thỏa mãn $L_F = L$. Giả sử tồn tại một f -họ F' sao cho $L_{F'} = L$ nhưng $F' \neq F$. Suy ra, tồn tại một PTH $X \rightarrow Y \in F'$ nhưng $X \rightarrow Y \notin F$. Từ $X \rightarrow Y \notin F$, định nghĩa của f -họ và giả thiết $L_{F'} = L$, chúng ta suy ra ngay $X \rightarrow Y \notin F'$. Điều này rõ ràng là mâu thuẫn. Vậy $F' = F$. $\square$

Như vậy, TTBD là một mô tả tương đương của PTH. Điều này có nghĩa để nghiên cứu và phân tích các đặc trưng của họ các PTH chúng ta có thể dùng công cụ TTBD.

5.2.2. Hệ bao đóng

Xét tập hữu hạn bất kỳ V và $\mathcal{F}$ là một họ các tập con của V . Họ $\mathcal{M}$ được gọi là một *hệ bao đóng*¹ (HBD) trên V nếu nó thỏa mãn các điều kiện sau:

S1) $V \in \mathcal{M}$.

S2) $X, Y \in \mathcal{M} \Rightarrow X \cap Y \in \mathcal{M}$.

¹Một số tác giả còn gọi là *họ Moore* hay *nửa dàn giao*.

Dễ thấy, HBĐ cũng có thể được đặc trưng lại như sau:

$$\mathcal{M} \subseteq \mathcal{P}(V) : (\forall \mathcal{N} \subseteq \mathcal{M} \Rightarrow \bigcap \mathcal{N} \in \mathcal{M}).$$

Như vậy, có thể thấy HBĐ $\mathcal{M}$ trên V là một họ các tập con của V đóng đối với phép giao và chứa V như là phần tử lớn nhất. Rõ ràng, tập tất cả các tập đóng $Closed(S)$ của LDQH $S = (U, F)$ là một HBĐ trên U .

Định lý 5.6. *Mỗi HBĐ $\mathcal{M}$ trên V xác định một TTBD $L \in Closure(V)$ như sau:*

$$L(X) = \bigcap \{Y \in \mathcal{M} : X \subseteq Y\} \quad (5.1)$$

Ngược lại, mỗi TTBD $L \in Closure(V)$ xác định một HBĐ

$$\mathcal{M} = \{X \subseteq V : L(X) = X\} \quad (5.2)$$

và tương ứng $L \leftrightarrow \mathcal{M}$ xác định như trên giữa các HBĐ và các TTBD là một tương ứng 1-1.

Chứng minh. Dễ thấy L thỏa tính phản xạ và tính đơn điệu. Do đó, ta chỉ cần chứng minh thêm L thỏa tính lũy đẳng. Rõ ràng

$$L(X) = X \Leftrightarrow X \in \mathcal{M} \quad (5.3)$$

Vì $L(X) \in \mathcal{M}$ nên suy ra L thỏa tính lũy đẳng.

Ngược lại, giả sử $L \in Closure(V)$. Khi đó ta có nếu với mọi $X_i \in \mathcal{M}$ và $\bigcap X_i = X$, thì $X \subseteq X_i$. Theo tính đơn điệu của L suy ra $L(X) \subseteq L(X_i) = X_i$ với mọi i . Do đó $L(X) \subseteq \bigcap X_i = X$. Kết hợp với tính phản xạ của L , suy ra $L(X) = X$, tức là $X \in \mathcal{M}$.

Tiếp theo, chúng ta chứng tỏ tương ứng $L \leftrightarrow \mathcal{M}$ như trên là tương ứng 1-1. Giả sử $\mathcal{M}$ là một HBĐ tùy ý, L là toán tử xác định bởi (5.1) và $\mathcal{M}'$ là HBĐ xác định bởi toán tử L theo công thức (5.2). Thế thì

$\mathcal{M} = \mathcal{M}'$ theo (5.3). Sau đó, ta lấy một TTBD L tùy ý và giả sử $\mathcal{M}$ là HBĐ xác định bởi toán tử L theo công thức (5.2), còn L' là toán tử xác định bởi hệ $\mathcal{M}$ theo công thức (5.1). Như vậy $\mathcal{M}$ cũng được xác định bởi toán tử L' , do đó

$$L(X) = X \Leftrightarrow L'(X) = X \quad (5.4)$$

Theo tính lũy đẳng thì $L(L(X)) = L(X)$, vì vậy từ (5.4) suy ra $L'(L(X)) = L(X)$. Nhưng $X \subseteq L(X)$ và áp dụng tính đơn điệu của L' , ta được

$$L'(X) \subseteq L'(L(X)) = L(X).$$

Do tính đối xứng, ta cũng có $L(X) \subseteq L(L'(X)) = L'(X)$. $\square$

Như vậy, HBĐ là một mô tả tương đương của TTBD. Từ Định lý 5.5 và Định lý 5.6 suy ra HBĐ cũng là một mô tả tương đương của PTH. Do đó, để nghiên cứu và phân tích các đặc trưng của họ các PTH chúng ta có thể dùng công cụ HBĐ.

5.2.3. Hàm chọn

Xét tập hữu hạn bất kỳ V . Ánh xạ $H : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ được gọi là *hàm chọn* (HC) trên V nếu $H(X) \subseteq X$ với mọi $X \subseteq V$.

Ví dụ 5.6. Sau đây là một số HC cơ bản:

- 1) HC *rỗng*: $e(X) = \emptyset$ với mọi $X \subseteq V$.
- 2) HC *đồng nhất*: $i(X) = X$ với mọi $X \subseteq V$.
- 3) HC *lùi 1*: $l_T^1(X) = X \setminus T$ với mọi $X \subseteq V$ và T là một tập con cố định của V .
- 4) HC *lùi 2*: $l_T^2(X) = X \cap T$ với mọi $X \subseteq V$ và T là một tập con cố định của V .

Các HC lùi có quan hệ với các HC rỗng và HC đồng nhất như sau:

- Nếu $T = \emptyset$ thì HC lùi 1 (tương ứng HC lùi 2) trở thành HC đồng nhất (tương ứng HC rỗng).
- Nếu $T = \emptyset$ thì HC lùi 1 (tương ứng HC lùi 2) trở thành HC rỗng (tương ứng HC đồng nhất).

Như vậy, chúng ta thấy có thể dùng HC lùi 1 hoặc HC lùi 2 làm cơ sở đặc tả cho họ các HC tương ứng $\{e, i, l_T^1\}$ hoặc $\{e, i, l_T^2\}$.

Xét TTBD $L \in \text{Closure}(V)$. Với mỗi $X \subseteq V$ ta đặt

$$H_L(X) = V \setminus L(V \setminus X). \quad (5.5)$$

Rõ ràng, H_L là một HC.

Định lý 5.7. Công thức (5.5) xác định tương ứng 1-1 giữa TTBD và HC thỏa các điều kiện sau: với mọi $X, Y \subseteq V$

C1) Nếu $H_L(X) \subseteq Y \subseteq X$ thì $H_L(X) = H_L(Y)$.

C2) Nếu $X \subseteq Y$ thì $H_L(X) \subseteq H_L(Y)$.

Chứng minh. Giả sử $L \in \text{Closure}(V)$. Ta chứng minh HC H_L xác định bởi công thức (5.5) thỏa hai tính chất (C1) và (C2). Giả sử $H_L(X) \subseteq Y \subseteq X$. Suy ra $V \setminus L(V \setminus X) \subseteq Y$ và $L(V \setminus X) \subseteq L(V \setminus Y)$. Nên $V \setminus Y \subseteq L(V \setminus X)$, và do đó $L(V \setminus Y) \subseteq L(V \setminus X)$. Như vậy $L(V \setminus X) = L(V \setminus Y)$. Hay $H_L(X) = H_L(Y)$. Trường hợp nếu $X \subseteq Y$ thì $L(V \setminus Y) \subseteq L(V \setminus X)$, và do đó $H_L(X) \subseteq H_L(Y)$.

Ngược lại, giả sử H là một HC trên V thỏa hai tính chất (C1) và (C2). Xét $L(X) = V \setminus H(V \setminus X)$. Ta sẽ chứng minh $L \in \text{Closure}(V)$. Rõ ràng tính phản xạ của L được suy ra từ định nghĩa HC của H . Tính đơn điệu (C2) của H suy ra ngay tính đơn điệu của L . Hơn nữa, với mọi $X \subseteq V$ ta có:

$$\begin{aligned} L(L(X)) &= L(V \setminus H(V \setminus X)) \\ &= V \setminus H(V \setminus (V \setminus H(V \setminus X))) \\ &= V \setminus H(H(V \setminus X)). \end{aligned}$$

Vì HC H thỏa tính chất (C1) nên nó có tính lũy đẳng. Do đó

$$\begin{aligned} L(L(X)) &= V \setminus H(V \setminus X) \\ &= L(X). \end{aligned}$$

Như vậy, L cũng thỏa tính lũy đẳng. Lúc này $L(V \setminus X) = V \setminus H(X)$ hay $H(X) = V \setminus L(V \setminus X)$ với mọi $X \subseteq V$. Điều này có nghĩa $H_L = H$. Cuối cùng, chúng ta chỉ cần chứng minh thêm ánh xạ xác định bởi (5.5) là một đơn ánh. Điều này là hiển nhiên vì với hai TTBD phân biệt L_1 và L_2 thì

$$\begin{aligned} H_{L_1}(X) &= V \setminus L_1(V \setminus X) \\ &\neq V \setminus L_2(V \setminus X) \\ &= H_{L_2}(X) \end{aligned}$$

với mọi $X \subseteq V$. □

Các HC thỏa hai điều kiện (C1) và (C2) còn được gọi là *HC đặc biệt* trên V . Ký hiệu $Choice(V)$ là tập tất cả các HC đặc biệt trên V . Dễ thấy các HC rỗng, HC đồng nhất và các HC lùi đều là các HC đặc biệt. Như vậy, HC đặc biệt là một mô tả tương đương của TTBD. Từ Định lý 5.5 và Định lý 5.7 suy ra HC đặc biệt cũng là một mô tả tương đương của PTH. Do đó, để nghiên cứu và phân tích các đặc trưng của họ các PTH chúng ta có thể dùng công cụ HC đặc biệt.

5.3. Họ các tập tối tiểu của thuộc tính

Khái niệm họ các tập tối tiểu của thuộc tính được đề xuất trong [28] (1993) khi các tác giả nghiên cứu về quan hệ Armstrong, đặc biệt trong việc giảm độ phức tạp của bài toán xây dựng quan hệ Armstrong. Gần đây họ các tập tối tiểu của thuộc tính đã bắt đầu được sử dụng trong các nghiên cứu về lĩnh vực khai phá dữ liệu và thu được nhiều kết quả

thứ vị [6, 7, 27]. Ý nghĩa và tầm quan trọng của họ các tập tối tiểu của thuộc tính có thể xem thêm trong [16].

5.3.1. Định nghĩa

Xét LĐQH $S = (U, F)$, tập con thuộc tính $X \subseteq U$ và một thuộc tính $a \in U$. Nếu PTH $X \rightarrow a \in F^+$ rút gọn trái thì tập thuộc tính X được gọi là *tập tối tiểu của thuộc tính a* . Tập tất cả các tập tối tiểu của a được gọi là *họ các tập tối tiểu của thuộc tính a* và ký hiệu $K(a, F)$ hay $K(a)$.

Một cách hình thức

$$K(a) = \{X \subseteq U : X \rightarrow a \in F^+, (\forall Y \subset X) \Rightarrow (Y \rightarrow a \notin F^+)\}.$$

Dễ thấy, $\{a\} \in K(a)$ và $K(a)$ là một hệ Sperner trên U . Trường hợp nếu $|U| > 1$, thì $U \notin K(a)$.

Ví dụ 5.7. Xét LĐQH $S = (U, F)$ với $U = \{a, b, c, d\}$ và $F = \{b \rightarrow ac, cd \rightarrow a, bd \rightarrow ac\}$.

Từ định nghĩa, ta có một số họ các tập tối tiểu như:

$$K(a) = \{\{a\}, \{b\}, \{c, d\}\}$$

$$K(b) = \{\{b\}\}.$$

5.3.2. Thuật toán tìm họ các tập tối tiểu

Cho LĐQH $S = (U, F)$, với $U = \{a_1, a_2, \dots, a_n\}$ và thuộc tính $a \in U$. Tìm một tập tối tiểu của a là thủ tục khá đơn giản. Chúng ta có thể thực hiện bằng quy nạp trong thời gian đa thức theo kích thước của S như sau:

$$X_0 := U$$

$$X_i := \begin{cases} X_{i-1} \setminus \{a_i\} & \text{nếu } X_{i-1} \setminus \{a_i\} \rightarrow a \in F^+ \\ X_{i-1} & \text{ngược lại} \end{cases}$$

với mọi $i = 1, 2, \dots, n$.

Đặt $X = X_n$. Khi đó $X \in K(a)$. Tính đúng của thủ tục này là dễ dàng và được dành làm bài tập. Độ phức tạp thời gian của thủ tục này là $O(|U|^2|F|)$. Tìm họ các tập tối tiểu $K(a)$ thì phức tạp hơn. Cơ sở cho thuật toán tìm họ $K(a)$ là kết quả sau đây.

Định lý 5.8. *Giả sử $L(a)$ là một họ khác rỗng gồm một số tập tối tiểu của thuộc tính a trong LDQH $S = (U, F)$ và $\{a\} \in L(a)$. Điều kiện cần và đủ để tồn tại một tập tối tiểu $X \in K(a) \setminus L(a)$ là*

$$(\exists X_i \in L(a))(Y \rightarrow Z \in F) : (\forall X_j \in L(a))(X_j \not\subseteq Y \cup (X_i \setminus Z)).$$

Chứng minh. Giả sử X là một tập tối tiểu mới của a chưa có trong $L(a)$. Để ý, $\{a\} \in L(a)$. Từ X ta xây dựng một tập thuộc tính cực đại W thỏa mãn $X \subseteq W \subset U$ và $L(a) \cup \{W\}$ vẫn là một hệ Sperner trên U . Theo định nghĩa, suy ra $W \rightarrow a \in F^+$ và $a \notin W$. Rõ ràng, nếu mọi PTH $Y \rightarrow Z \in F$ kéo theo ($Y \subseteq W$ và $Z \subseteq W$) hoặc ($Y \not\subseteq W$), thì $W \in \text{Closed}(S)$. Lúc đó $W \rightarrow a \notin F^+$. Điều này là mâu thuẫn. Do đó, phải tồn tại một PTH $Y \rightarrow Z \in F$ sao cho $Y \subseteq W$ và $Z \not\subseteq W$. Theo cách xây dựng của W , suy ra phải tồn tại một $X_i \in L(a)$ sao cho $X_i \setminus Z \subseteq W$. Như vậy, $Y \cup (X_i \setminus Z) \subseteq W$. Điều này có nghĩa, $Y \cup (X_i \setminus Z) \not\subseteq X_j$ với mọi $X_j \in L(a)$.

Ngược lại, giả sử tồn tại tập tối tiểu $X_i \in L(a)$, tồn tại một PTH $Y \rightarrow Z \in F$ sao cho $Y \cup (X_i \setminus Z) \not\subseteq X_j$ với mọi $X_j \in L(a)$. Theo định nghĩa họ các tập tối tiểu của a , suy ra $Y \cup (X_i \setminus Z) \rightarrow a \in F^+$. Điều này có nghĩa, tồn tại một tập tối tiểu X_k của a sao cho $X_k \subseteq Y \cup (X_i \setminus Z)$. Tuy nhiên, theo giả thiết $Y \cup (X_i \setminus Z)$ không chứa bất kỳ một tập tối tiểu nào của a ở trong $L(a)$, do đó X_k phải là một tập tối tiểu mới của a chưa có trong $L(a)$. $\square$

Dễ thấy, nếu với mỗi PTH $Y \rightarrow Z \in F$ chúng ta đều có $a \in Y$ hoặc $a \notin Z$, thì khi đó $K(a) = \{\{a\}\}$.

Thuật toán 5.1 (Tìm họ các tập tối tiểu của thuộc tính)

Vào: LĐQH $S = (U, F)$ và một thuộc tính $a \in U$.

Ra: Họ các tập tối tiểu $K(a)$.

Phương pháp:

Bước 1. Đặt $L(a) := \{\{a\}\}$.

Bước 2. (*Lặp*) Nếu có một tập tối tiểu $X_i \in L(a)$, một PTH $Y \rightarrow Z \in F$ sao cho $Y \cup (X_i \setminus Z) \not\subseteq X_j$ với mọi $X_j \in L(a)$ thì ta xây dựng tập tối tiểu mới $X \in K(a)$ thỏa mãn $X \subseteq Y \cup (X_i \setminus Z)$. Sau đó đặt $L(a) := L(a) \cup \{X\}$ và lặp lại Bước 2.

Trong trường hợp ngược lại, ta đặt $K(a) := L(a)$. Thuật toán dừng.

Vì U và F là hữu hạn nên theo Định lý 5.8, sau một số hữu hạn bước Thuật toán 5.1 sẽ dừng và cho ra đúng họ $K(a)$. Có thể kiểm chứng được độ phức tạp thời gian của Thuật toán 5.1 là $O(|U||F||K(a)|(|U| + |K(a)|))$. Như vậy, độ phức tạp thời gian của Thuật toán 5.1 là đa thức theo $|U|$, $|F|$ và $|K(a)|$. Do đó, rõ ràng nếu số tập tối tiểu của a là đa thức theo kích thước của S thì Thuật toán 5.1 rất hiệu quả, đặc biệt khi $K(a)$ là nhỏ.

Xét LĐQH S ở Ví dụ 5.7. Vận dụng Thuật toán 5.1, ta thu được các họ các tập tối tiểu của thuộc tính trên S như sau:

$$K(a) = \{\{a\}, \{b\}, \{c, d\}\}, K(b) = \{\{b\}\}, K(c) = \{\{c\}, \{b\}\}, K(d) = \{\{d\}\}.$$

5.3.3. Họ các tập tối tiểu và quan hệ Armstrong

Chúng ta đã biết, tập $Closed(F)$ đóng kín đối với phép toán giao. Lúc đó một tập $X \in Closed(F)$ được gọi là *bất khả quy (thành giao)* của F nếu $X = Y \cap Z$, với $Y, Z \in Closed(F)$, thì kéo theo $X = Y$ hoặc $X = Z$. Họ tất cả các tập bất khả quy của F ký hiệu $MI(F)$. Như vậy $MI(F)$ chính là họ con tối tiểu duy nhất của $Closed(F)$ sao cho các phần tử của $Closed(F)$ đều biểu diễn được qua giao của các phần tử trong $MI(F)$ [12]. Do vậy, ta còn ký hiệu $MI(F)$ lại là $Gen(F)$, và gọi $Gen(F)$ là *hệ sinh* của $Closed(F)$ hay gọn hơn hệ sinh của F . Để ý

$U \notin Gen(F)$. Một cách hình thức các biểu diễn của $Gen(F)$ sẽ là:

$$Gen(F) = \{X \in Closed(F) : X \neq U, (Y, Z \in Closed(F), Y \neq X, Z \neq X) \Rightarrow (Y \cap Z \neq X)\}$$

hay

$$Gen(F) = \{X \in Closed(F) : X \subset \bigcap \{Y \in Closed(F) : X \subset Y\}\}.$$

Bây giờ ta gọi

$$M(F, a) = \{Y \subseteq U : Y \text{ là tập khác rỗng cực đại (ứng với } \subseteq) \text{ sao cho } Y \rightarrow a \notin F^+\}.$$

Theo [12] ta có $Gen(F) = M(F) = \bigcup_{a \in U} M(F, a)$. Có thể kiểm chứng được $\overline{Tran(M(a))} = M(F, a)$ với $a \in U$. Từ đây và theo [12], ta có thuật toán xây dựng quan hệ Armstrong như sau.

Thuật toán 5.2 (Xây dựng quan hệ Armstrong)

Vào: LĐQH $S = (U, F)$.

Ra: Quan hệ Armstrong của S .

Phương pháp:

Bước 1. Với mỗi thuộc tính $a \in U$, tính họ các tập tối tiểu $K(a)$.

Bước 2. Với mỗi $K(a)$, xây dựng họ $\overline{Tran(K(a))}$.

Bước 3. Đặt $Gen(F) := \bigcup_{a \in U} \overline{Tran(K(a))}$, và giả sử các phần tử của họ $Gen(F)$ là $A_1, A_2, \dots, A_k$.

Bước 4. Từ $Gen(F)$, xây dựng quan hệ $R = \{t_0, t_1, \dots, t_k\}$ như sau:

$$t_0(a) = 0, \forall a \in U$$

$$t_i(a) = \begin{cases} 0 & \text{nếu } a \in A_i \\ i & \text{ngược lại} \end{cases}$$

với mọi $i = 1, 2, \dots, k$.

Kết luận, R là quan hệ Armstrong của S .

Độ phức tạp thời gian của Thuật toán 5.2 chính là độ phức tạp

của Bước 1 và Bước 2. Do đó, độ phức tạp thời gian của Thuật toán 5.2 sẽ là

$$O\left(n \sum_{i=1}^n \left(n \sum_{q=1}^{m_i-1} t_{iq} u_{iq} + |F| m_i (m_i + n)\right)\right),$$

ở đây $m_i = |K(a_i)|$ và

$$u_{iq} = \begin{cases} l_{iq} - t_{iq} & \text{nếu } l_{iq} > t_{iq} \\ 1 & \text{ngược lại.} \end{cases}$$

Trong trường hợp nếu $l_{iq} \leq l_{m_i} (\forall i, \forall q : 1 \leq q \leq m_i)$, độ phức tạp thời gian của Thuật toán 5.2 sẽ là

$$O\left(n \sum_{i=1}^n |K(a_i)| (n|F| + |K(a_i)||F| + n|\overline{Tran(K(a_i))}|^2)\right).$$

Như vậy, độ phức tạp thời gian của Thuật toán 5.2 là đa thức theo $|U|, |F|, |K(a_i)|$ và $|Tran(K(a_i))|$. Rõ ràng trong trường này, nếu $|K(a_i)|, |Tran(K(a_i))|$ là đa thức (đặc biệt nếu chúng là nhỏ) theo $|U|$ và $|F|$ thì Thuật toán 5.2 rất hiệu quả.

Xét LĐQH S ở Ví dụ 5.7. Ta đã biết

$$K(a) = \{\{a\}, \{b\}, \{c, d\}\}$$

$$K(b) = \{\{b\}\}$$

$$K(c) = \{\{c\}, \{b\}\}$$

$$K(d) = \{\{d\}\}.$$

Suy ra

$$\overline{Tran(K(a))} = \{\{c\}, \{d\}\}, \overline{Tran(K(b))} = \{\{a, c, d\}\}$$

$$\overline{Tran(K(c))} = \{\{a, d\}\}, \overline{Tran(K(d))} = \{\{a, b, c\}\}.$$

Lúc đó

$$Gen(F) := \{\{a, b, c\}, \{a, c, d\}, \{a, d\}, \{c\}, \{d\}\}.$$

Như vậy, quan hệ Armstrong R của S sẽ là:

$$R = \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 2 & 0 & 0 \\ \hline 0 & 3 & 3 & 0 \\ \hline 4 & 4 & 0 & 4 \\ \hline 5 & 5 & 5 & 0 \\ \hline \end{array}$$

5.4. Mở rộng phụ thuộc hàm

Như chúng ta đã biết, PTH đảm bảo tính chính xác cũng như tính nhất quán của dữ liệu. Nhờ có các PTH các hệ quản trị CSDL mới quản lý tốt được chất lượng của dữ liệu. Tuy vậy, trong thực tế do có một số giá trị dữ liệu không chính xác hoặc một số ngoại lệ nào đó làm cho các PTH không thỏa. Sự phụ thuộc tuyệt đối theo nghĩa hàm này có vẻ quá nghiêm ngặt khi chúng ta hình dung tới một quan hệ có hàng ngàn bộ, trong khi đó chỉ có khoảng vài bộ vi phạm PTH. Điều này làm mất tính chất phụ thuộc vốn có giữa các thuộc tính. Vì vậy, một cách tự nhiên nếu chúng ta xem xét đến sự phụ thuộc bộ phận của các bộ dữ liệu hoặc cho phép một số lượng lỗi nhất định của các bộ dữ liệu đối với PTH. Người ta gọi trường hợp đầu tiên là phụ thuộc cấp k , trường hợp thứ hai là PTH xấp xỉ. Bên cạnh đó, các trường hợp này hoàn toàn có thể dụng lý thuyết tập thô để tiếp cận.

5.4.1. Cơ sở tập thô

Lý thuyết tập thô do nhà toán tin học người Ba Lan Z. Pawlak đề xuất vào đầu những năm 1980 với mục đích xử lý những thông tin mơ hồ và không chắc chắn. Triết lý của tập thô dựa trên giả sử rằng mọi đối tượng trong vũ trụ đều gắn với một thông tin nào đó (như dữ liệu, tri thức). Chẳng hạn, nếu các đối tượng là các bệnh nhân bị một bệnh nhất định nào đó, thì các triệu chứng của bệnh tạo thành thông tin về

bệnh nhân. Lúc đó các đối tượng được đặc trưng bởi cùng thông tin thì không thể phân biệt. Quan hệ không phân biệt này chính là một quan hệ tương đương, các đối tượng không thể phân biệt (tương tự) được gọi là tập cơ sở và tạo thành nguyên tử của tri thức về vũ trụ. Một hợp bất kỳ của các tập cơ sở được gọi là tập chính xác (hay tập xác định, tập rõ), ngược lại được gọi là tập không chính xác (hay tập không xác định, tập thô). Trong lý thuyết tập thô, bất kỳ các khái niệm không rõ ràng nào đều có thể thay bằng một cặp xấp xỉ dưới và xấp xỉ trên của khái niệm không rõ ràng. Ở đây xấp xỉ dưới là gồm tất cả các đối tượng chắc chắn thuộc về khái niệm và xấp xỉ trên là gồm tất cả các đối tượng có thể thuộc về khái niệm. Hiệu của xấp xỉ trên và xấp xỉ dưới tạo thành miền biên của khái niệm không rõ ràng.

Như vậy theo quan điểm logic, lý thuyết tập thô được xây dựng để có thể hiểu tốt hơn tính mơ hồ. Ở quan điểm thực hành, lý thuyết tập thô chính là ý tưởng nền tảng cho các lĩnh vực như trí tuệ nhân tạo, học máy, phát hiện tri thức, phân tích quyết định, nhận dạng mẫu, hệ trợ giúp quyết định, cơ sở dữ liệu, khai phá dữ liệu, phân tích dữ liệu ... Do đó, hiện nay lý thuyết tập thô đã thu hút sự quan tâm của rất nhiều nhà khoa học của nhiều lĩnh vực trên toàn thế giới. Nhiều hệ thống phần mềm sử dụng lý thuyết toán học này đã được cài đặt và nhiều ứng dụng quan trọng trong đời sống của lý thuyết này cũng đã được xây dựng như y học, dược học, kỹ thuật, ngân hàng, nhận dạng mẫu, biểu thị gen.

5.4.1.1. Quan hệ không phân biệt

Xét tập thuộc tính $U = \{a_1, a_2, \dots, a_n\}$, quan hệ $R \in Rel(U)$ và tập con thuộc tính $X \subseteq U$. Quan hệ hai ngôi sau

$$IND(R, X) = \{(t_1, t_2) \in R^2 : t_1(X) = t_2(X)\}$$

được gọi là *quan hệ X -không phân biệt trên R* . Nếu không sợ nhầm lẫn, quan hệ $IND(R, X)$ sẽ được viết gọn lại là $IND(X)$. Rõ ràng $IND(X)$ là một quan hệ tương đương trên R sinh bởi X . Lớp tương đương của $t \in R$ theo quan hệ $IND(X)$ ký hiệu là $[t]_{IND(X)}$ hay $[t]_X$. Tập thương của R theo quan hệ tương đương $IND(X)$ ký hiệu là $R/IND(X)$ hay R/X . Như vậy, tập thương R/X sẽ phân hoạch quan hệ R thành các lớp tương đương, mỗi lớp tương đương là một tập con của R chứa các bộ có giá trị giống nhau trên X .

Lý thuyết tập thô đặt ra câu hỏi: với các tri thức được cho bởi tập thuộc tính X , liệu chúng ta có thể biểu diễn tập các bộ (là các đối tượng nào đó) $O \subseteq R$ bằng các tri thức có sẵn X hay không? Hay nói cách khác với tập thuộc tính X cho trước, chúng ta có các lớp tương đương theo quan hệ $IND(X)$, thế thì một tập các bộ O có thể diễn đạt thông qua các lớp tương đương này như thế nào? Để làm điều này, tức biểu diễn O bằng tri thức có sẵn X , lý thuyết tập thô sẽ xấp xỉ chúng bởi hợp một số hữu hạn các lớp tương đương theo quan hệ $IND(X)$. Có hai cách xấp xỉ như vậy, cách thứ nhất cho tương ứng bởi “miền trong”, cách thứ hai có thể xấp xỉ bởi “bao đóng” của O . Các xấp xỉ này sẽ được định nghĩa ở mục ngay sau đây.

5.4.1.2. Các xấp xỉ

Xét tập con các bộ $O \subseteq R$ và tập con các thuộc tính $X \subseteq U$. Các tập hợp sau

$$\underline{X}(O) = \{t \in R : [t]_X \subseteq O\}$$

và

$$\overline{X}(O) = \{t \in R : [t]_X \cap O \neq \emptyset\}$$

tương ứng được gọi là *X -xấp xỉ dưới* và *X -xấp xỉ trên* của O . Các xấp xỉ $\underline{X}(O)$ và $\overline{X}(O)$ còn được ký hiệu gọn lại tương ứng là $\underline{X}O$ và $\overline{X}O$. Như vậy, xấp xỉ dưới $\underline{X}O$ bao gồm các bộ chắc chắn thuộc về O và xấp xỉ trên $\overline{X}O$ chứa tất cả các bộ có khả năng thuộc về O . Tập $B_X(O) = \overline{X}O \setminus \underline{X}O$

được gọi là X -miền biên của O và tập $R \setminus \overline{XO}$ được gọi là X -miền ngoài của O . Ở đây tập $B_X(O)$ chứa các bộ không chắc chắn thuộc hay không thuộc O và X -miền ngoài của O chứa các đối tượng chắc chắn không thuộc O .

Bởi các tính chất của lớp tương đương nên các xấp xỉ trên còn có thể biểu diễn lại như sau:

$$\underline{XO} = \bigcup \{[t]_X \in R/X : [t]_X \subseteq O\}$$

và

$$\overline{XO} = \bigcup \{[t]_X \in R/X : [t]_X \cap O \neq \emptyset\}.$$

Ví dụ 5.8. Xét quan hệ $R = \{t_1, t_2, t_3\} \in Rel(U)$, với $U = \{a, b, c\}$, như sau:

$$R = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline 2 & 0 & 0 \\ \hline \end{array}$$

Xét tập thuộc tính $X = \{c\}$. Ta có $R/X = \{\{t_1, t_2\}, \{t_3\}\}$.

Chẳng hạn, xét tập các bộ $O = \{t \in R : t(a) = 1 \vee t(a) = 2\} = \{t_2, t_3\}$. Khi đó

$$\underline{XO} = \{t_3\}, \overline{XO} = \{t_1, t_2, t_3\}, B_X(O) = \{t_1, t_2\}.$$

Các xấp xỉ dưới và trên có những tính chất cơ bản sau đây.

Mệnh đề 5.9.

- 1) $\underline{XO} \subseteq O \subseteq \overline{XO}$.
- 2) $\underline{X}\emptyset = \overline{X}\emptyset = \emptyset, \underline{X}R = \overline{X}R = R$.
- 3) $\underline{X}(O_1 \cap O_2) = \underline{X}O_1 \cap \underline{X}O_2$.
- 4) $\overline{X}(O_1 \cup O_2) = \overline{X}O_1 \cup \overline{X}O_2$.
- 5) $O_1 \subseteq O_2 \Rightarrow \underline{X}O_1 \subseteq \underline{X}O_2, \overline{X}O_1 \subseteq \overline{X}O_2$.
- 6) $\underline{X}(O_1 \cup O_2) \supseteq \underline{X}O_1 \cup \underline{X}O_2$.

$$7) \overline{X}(O_1 \cap O_2) \subseteq \overline{X}O_1 \cap \overline{X}O_2.$$

$$8) \underline{X}(R \setminus O) = R \setminus \overline{X}O.$$

$$9) \overline{X}(R \setminus O) = R \setminus \underline{X}O.$$

$$10) \underline{X}(\underline{X}O) = \overline{X}(\underline{X}O) = \underline{X}O.$$

$$11) \underline{X}(\overline{X}O) = \overline{X}(\overline{X}O) = \overline{X}O.$$

Chứng minh.

1) Nếu $t \in \underline{X}O$ thì $[t]_X \subseteq O$. Vì $t \in [t]_X$, do đó $t \in O$.

Trường hợp nếu $t \in O$, thì $[t]_X \cap O \neq \emptyset$. Suy ra $t \in \overline{X}O$. Tóm lại, ta thu được $\underline{X}O \subseteq O \subseteq \overline{X}O$.

2) Theo (1) ta có $\underline{X}\emptyset \subseteq \emptyset$, và do đó $\underline{X}\emptyset = \emptyset$. Trường hợp $\overline{X}\emptyset = \emptyset$, ta chứng minh bằng phản chứng. Giả sử tồn tại $t \in \overline{X}\emptyset$. Suy ra $[t]_X \cap \emptyset \neq \emptyset$. Điều này là vô lý.

Cũng theo (1), ta có $\underline{X}R \subseteq R$. Ngoài ra nếu $t \in R$, thì $[t]_X \in R$. Do đó $t \in \underline{X}R$. Như vậy, $\underline{X}R = R$. Trường hợp $\overline{X}R = R$, suy ra từ (1).

3) Ta có

$$\begin{aligned} t \in \underline{X}(O_1 \cap O_2) &\Leftrightarrow [t]_X \subseteq O_1 \cap O_2 \\ &\Leftrightarrow [t]_X \subseteq O_1 \wedge [t]_X \subseteq O_2 \\ &\Leftrightarrow t \in \underline{X}O_1 \cap \underline{X}O_2. \end{aligned}$$

Suy ra $\underline{X}(O_1 \cap O_2) = \underline{X}O_1 \cap \underline{X}O_2$.

4) Ta có

$$\begin{aligned} t \in \overline{X}(O_1 \cup O_2) &\Leftrightarrow [t]_X \cap (O_1 \cup O_2) \neq \emptyset \\ &\Leftrightarrow (([t]_X \cap O_1) \cup ([t]_X \cap O_2)) \neq \emptyset \\ &\Leftrightarrow ([t]_X \cap O_1) \neq \emptyset \vee ([t]_X \cap O_2) \neq \emptyset \\ &\Leftrightarrow t \in \overline{X}O_1 \vee t \in \overline{X}O_2 \\ &\Leftrightarrow t \in \overline{X}O_1 \cup \overline{X}O_2. \end{aligned}$$

Do đó $\overline{X}(O_1 \cup O_2) = \overline{X}O_1 \cup \overline{X}O_2$.

5) Rõ ràng nếu $O_1 \subseteq O_2$, thì $\underline{X}(O_1 \cap O_2) = \underline{X}O_1$. Theo (3), suy ra $\underline{X}O_1 \subseteq \underline{X}O_2$. Cũng lập luận tương tự, dựa vào (4) nếu $O_1 \subseteq O_2$, thì ta cũng có $\overline{X}O_1 \subseteq \overline{X}O_2$.

Tính chất (6) và (7) dễ dàng suy ra ngay từ tính đơn điệu (5).

8) Ta có

$$\begin{aligned} t \in \overline{X}O &\Leftrightarrow [t]_X \cap O \neq \emptyset \\ &\Leftrightarrow [t]_X \not\subseteq R \setminus O \\ &\Leftrightarrow t \notin \underline{X}(R \setminus O). \end{aligned}$$

Như vậy, $\overline{X}O = R \setminus \underline{X}(R \setminus O)$. Hay $\underline{X}(R \setminus O) = R \setminus \overline{X}O$.

9) Lập luận tương tự (8) ta cũng có

$$\begin{aligned} t \in \underline{X}O &\Leftrightarrow [t]_X \subseteq O \\ &\Leftrightarrow [t]_X \cap (R \setminus O) = \emptyset \\ &\Leftrightarrow t \notin \overline{X}(R \setminus O). \end{aligned}$$

Như vậy, $\underline{X}O = R \setminus \overline{X}(R \setminus O)$ hay $\overline{X}(R \setminus O) = R \setminus \underline{X}O$.

10) Ta có nếu $t \in \underline{X}O$ thì $[t]_X \subseteq O$. Suy ra

$$\begin{aligned} \underline{X}O &\supseteq \underline{X}([t]_X) \\ &= [t]_X. \end{aligned}$$

Nên $t \in \underline{X}(\underline{X}O)$. Ngoài ra theo (1) thì $\underline{X}(\underline{X}O) \subseteq \underline{X}O$. Như vậy, $\underline{X}(\underline{X}O) = \underline{X}O$.

Bây giờ nếu $t \in \overline{X}(\underline{X}O)$, thì $[t]_X \cap \underline{X}O \neq \emptyset$. Suy ra, tồn tại $u \in [t]_X$ sao cho $u \in \underline{X}O$, và do đó $[u] \subseteq O$ hay $[t] \subseteq O$. Nên $t \in \underline{X}O$. Hơn nữa theo (1) ta có $\underline{X}O \subseteq \overline{X}(\underline{X}O)$. Vậy, $\overline{X}(\underline{X}O) = \underline{X}O$.

11) Ta có nếu $t \in \overline{X}O$ thì $[t]_X \cap O \neq \emptyset$. Suy ra $[t]_X \subseteq \overline{X}O$, và do đó $t \in \underline{X}(\overline{X}O)$. Ngoài ra theo (1) thì $\underline{X}(\overline{X}O) \subseteq \overline{X}O$. Như vậy, $\underline{X}(\overline{X}O) = \overline{X}O$.

Bây giờ nếu $t \in \overline{X}(\overline{XO})$, thì $[t]_X \cap \overline{XO} \neq \emptyset$. Suy ra, tồn tại $u \in [t]_X$ sao cho $u \in \overline{XO}$, và do đó $[u]_X \cap O \neq \emptyset$ hay $[t]_X \cap O \neq \emptyset$. Vậy nên $t \in \overline{XO}$. Mặt khác theo (1), ta có $\overline{XO} \subseteq \overline{X}(\overline{XO})$. Như vậy, $\overline{X}(\overline{XO}) = \overline{XO}$. $\square$

Từ mệnh đề suy ra ngay các xấp xỉ trên cũng là các TTBD, nhưng các xấp xỉ dưới thì không phải.

5.4.1.3. Tập thô

Tập con các bộ $O \subseteq R$ còn được gọi là một *phạm trù*. Trường hợp nếu $O \in IND(X)$, thì O được gọi là *phạm trù X -cơ sở*. Khi O là hợp của một số phạm trù X -cơ sở, thì O được gọi là *X -xác định* hay *X -rõ*. Trong trường hợp ngược lại, O được gọi là *X -không xác định* hay *X -thô*.

Như vậy, các tập X -rõ chính là các tập con các bộ của quan hệ R được xác định một cách chính xác qua cơ sở X , ngược lại các tập X -thô là không thể xác định chính xác được qua cơ sở X . Tuy vậy, trong trường hợp này, chúng ta có thể xác định các tập X -thô bằng cách xấp xỉ qua hai tập rõ là $\underline{XO}$ và $\overline{XO}$.

Mệnh đề 5.10.

- 1) O là X -rõ khi và chỉ khi $\underline{XO} = \overline{XO}$.
- 2) O là X -thô khi và chỉ khi $\underline{XO} \neq \overline{XO}$.

Chứng minh.

1) Giả sử $O = \bigcup_{[t]_X \in R/X} [t]_X$. Suy ra

$$\begin{aligned} \underline{XO} &= \bigcup \{[t]_X \in R/X : [t]_X \subseteq O\} \\ &= O \\ &= \bigcup \{[t]_X \in R/X : [t]_X \cap O \neq \emptyset\} \\ &= \overline{XO}. \end{aligned}$$

Ngược lại, nếu $\underline{XO} = \overline{XO}$ thì với mọi $[t]_X \in R/X$ sao cho $[t]_X \cap O \neq \emptyset$, ta có $[t]_X \subseteq O$. Suy ra $O = \bigcup_{[t]_X \in R/X} [t]_X$.

2) Suy ra ngay từ (1). □

Tập thô cũng có thể đặc trưng bằng số qua hệ số sau, còn được gọi là độ chính xác của xấp xỉ. *Độ chính xác* của tập thô O đối với X , ký hiệu $\alpha_X(O)$, là tỷ số xác định bởi

$$\alpha_X(O) = \frac{|\underline{X}O|}{|\overline{X}O|}.$$

Rõ ràng $0 \leq \alpha_X(O) \leq 1$. Khi $\alpha_X(O) = 1$ thì O là X -rõ. Ngược lại, nếu $\alpha_X(O) < 1$ thì O là X -thô.

Cũng dựa vào các xấp xỉ, tập thô được phân làm bốn loại như sau.

- 1) Tập O được gọi là *X -xác định thô* nếu $\underline{X}O \neq \emptyset$ và $\overline{X}O \neq R$.
- 2) Tập O được gọi là *X -không xác định trong* nếu $\underline{X}O = \emptyset$ và $\overline{X}O \neq R$.
- 3) Tập O được gọi là *X -không xác định ngoài* nếu $\underline{X}O \neq \emptyset$ và $\overline{X}O = R$.
- 4) Tập O được gọi là *X -không xác định toàn phần* nếu $\underline{X}O = \emptyset$ và $\overline{X}O = R$.

Ví dụ 5.9. Xét quan hệ $R = \{t_0, t_1, \dots, t_{10}\}$ trên một tập thuộc tính bất kỳ U và một tập con tùy ý $X \subseteq U$. Giả sử quan hệ $IND(X)$ phân hoạch R thành các lớp tương đương như sau:

$$R/X = \{\{t_0, t_1\}, \{t_2, t_6, t_9\}, \{t_3, t_5\}, \{t_4, t_8\}, \{t_7, t_{10}\}\}.$$

Khi đó với:

- $O = \{t_0, t_1, t_4, t_8\}$ thì $\underline{X}O = \overline{X}O = O$. Suy ra O là X -rõ.
- $O = \{t_1, t_7, t_8, t_9\}$ thì $\underline{X}O = \{t_7, t_{10}\}$ và $\overline{X}O = \{t_0, t_1, t_4, t_7, t_8, t_{10}\}$. Do đó O là X -xác định thô.
- $O = \{t_0, t_2, t_3, t_4, t_8, t_{10}\}$ thì $\underline{X}O = \{t_4, t_8\}$ và $\overline{X}O = R$. Như vậy O là X -không xác định ngoài.

- $O = \{t_2, t_3, t_4\}$ thì $\underline{X}O = \emptyset$ và $\overline{X}O = \{t_2, t_3, \dots, t_6, t_8, t_9\}$. Như vậy O là X -không xác định trong.
- $O = \{t_0, t_2, t_3, t_4, t_7\}$ thì $\underline{X}O = \emptyset$ và $\overline{X}O = R$. Như vậy O là X -không xác định toàn phần.

Mệnh đề 5.11.

1) Tập O là X -xác định (tương ứng X -xác định thô, X -không xác định toàn phần) khi và chỉ khi tập $R \setminus O$ cũng như thế.

2) Tập O là X -không xác định ngoài (tương ứng trong) khi và chỉ khi tập $R \setminus O$ là X -không xác định trong (tương ứng ngoài).

Chứng minh.

1) Ta có O là X -xác định thô khi và chỉ khi:

$$\begin{aligned}
&\Leftrightarrow \begin{cases} \underline{X}O \neq \emptyset \\ \overline{X}O \neq R \end{cases} \\
&\Leftrightarrow \begin{cases} \exists t \in O : [t]_X \subseteq O \\ \exists t \in R : [t]_X \cap \overline{X}O = \emptyset \end{cases} \\
&\Leftrightarrow \begin{cases} [t]_X \cap (R \setminus O) = \emptyset \\ [t]_X \subseteq (R \setminus \overline{X}O) \end{cases} \\
&\Leftrightarrow \begin{cases} \overline{X}(R \setminus O) \neq R \\ [t]_X \subseteq \underline{X}(R \setminus O) \end{cases} \\
&\Leftrightarrow \begin{cases} \overline{X}(R \setminus O) \neq R \\ \underline{X}(R \setminus O) \neq \emptyset \end{cases} \\
&\Leftrightarrow \begin{cases} \overline{X}(R \setminus O) \neq R \\ \underline{X}(R \setminus O) \neq \emptyset \end{cases} \\
&\Leftrightarrow R \setminus O \text{ là } X\text{-xác định thô.}
\end{aligned}$$

Trường hợp O là X -xác định (tương ứng X -không xác định toàn phần) khi và chỉ khi $R \setminus O$ cũng như thế là lập luận hoàn toàn tương tự.

2) Giả sử O là X -không xác định ngoài. Khi đó, ta có

$$\begin{aligned}
& \begin{cases} \underline{X}O \neq \emptyset \\ \overline{X}O = R \end{cases} \\
& \Leftrightarrow \begin{cases} R \setminus \overline{X}O = \emptyset \\ R \setminus \underline{X}O \neq R \end{cases} \\
& \Leftrightarrow \begin{cases} \underline{X}(R \setminus O) = \emptyset \\ \overline{X}(R \setminus O) \neq R \end{cases} \\
& \Leftrightarrow R \setminus O \text{ là } X\text{-không xác định trong.}
\end{aligned}$$

Trường hợp O là X -không xác định trong khi và chỉ khi $R \setminus O$ là X -không xác định ngoài lập luận tương tự. $\square$

5.4.2. Phụ thuộc cấp k

5.4.2.1. Định nghĩa

Xét quan hệ $R \in \text{Rel}(U)$ và các tập con thuộc tính $X, Y \subseteq U$. Tập các bộ trong R sao cho có thể phân lớp một cách duy nhất đúng vào các lớp tương đương trong R/Y bằng cách dùng (tri thức) tập X được gọi là X -miền khẳng định của Y trên R , và ký hiệu là $POS_X(R, Y)$ hay $POS_X(Y)$.

Một cách hình thức

$$POS_X(Y) = \bigcup_{[t]_Y \in R/Y} \underline{X}[t]_Y.$$

Khi đó, ta đặt

$$k = \frac{|POS_X(Y)|}{|R|}$$

và gọi hệ số k là *độ phụ thuộc* của Y vào X . Rõ ràng $0 \leq k \leq 1$. Lúc này, ta cũng nói Y *phụ thuộc cấp k vào X* (trên R) và ký hiệu $X \xrightarrow{k} Y$. Dễ kiểm chứng được khi $k = 1$ thì $X \xrightarrow{k} Y$ chính là PTH $X \rightarrow Y$ đúng trên R , còn khi $k = 0$ thì $X \xrightarrow{k} Y$ chính là PTH $X \rightarrow Y$ không đúng trên R . Điều này có nghĩa, các phụ thuộc cấp k là một dạng mở rộng của các PTH. Ở đây để đơn giản, phụ thuộc cấp $X \xrightarrow{1} Y$ được viết lại $X \rightarrow Y$ và gọi Y *phụ thuộc hoàn toàn vào X* . Do đó, trong các trường hợp $k < 1$ thì ta nói Y *phụ thuộc bộ phận cấp k vào X* .

Dễ thấy $[t]_X, [t]_Y, POS_X(Y) \in Rel(U)$ và $POS_X(Y)(X \rightarrow Y)$. Khi đó, miền $POS_X(Y)$ có thể biểu diễn lại

$$POS_X(Y) = \bigcup \{[t]_X \in R/X : \exists [t]_Y \in R/Y, [t]_X \subseteq [t]_Y\}.$$

Như vậy, để nghiên cứu cấu trúc logic của PTH chúng ta thấy hoàn toàn có thể sử dụng công cụ tập thô.

Ví dụ 5.10. Xét quan hệ $R = \{t_1, t_2, t_3\} \in Rel(U)$, với $U = \{a, b, c\}$, như sau:

a	b	c
0	0	1
1	0	1
2	0	0

Ta có $R/\{a\} = \{\{t_1\}, \{t_2\}, \{t_3\}\}$ và $R/\{c\} = \{\{t_1, t_2\}, \{t_3\}\}$.

Khi đó, với $X = \{a\}$ và $Y = \{c\}$ thì $\underline{X}\{t_1, t_2\} = \{t_1, t_2\}$ và $\underline{X}\{t_3\} = \{t_3\}$. Suy ra $POS_X(Y) = R$. Do đó $X \xrightarrow{1} Y$.

Trường hợp nếu $X = \{c\}$ và $Y = \{a\}$, thì $\underline{X}\{t_1\} = \emptyset$, $\underline{X}\{t_2\} = \emptyset$, $\underline{X}\{t_3\} = \{t_3\}$.

Suy ra $POS_X(Y) = \{t_3\}$. Vậy $X \xrightarrow{1/3} Y$.

Mệnh đề 5.12. Các mệnh đề sau là tương đương:

- 1) $X \rightarrow Y$.
- 2) $IND(X \cup Y) = IND(X)$.
- 3) $POS_X(Y) = R$.
- 4) $\underline{X}[t]_Y = [t]_Y$ với mọi $[t]_Y \in R/Y$.

Chứng minh.

1) $\Rightarrow$ 2) Biết rằng với mỗi $[t_1]_{X \cup Y} \in R/X \cup Y$, tồn tại $[t_2]_X \in R/X$ sao cho $[t_1]_{X \cup Y} \subseteq [t_2]_X$. Ta sẽ chứng minh $[t_1]_{X \cup Y} = [t_2]_X$. Thật vậy, giả sử tồn tại bộ $u \in [t_2]_X$ sao cho $u \notin [t_1]_{X \cup Y}$. Lấy bộ $v \in [t_1]_{X \cup Y}$. Suy ra $v \in [t_2]_X$. Nên $u(X) = v(X)$, và do đó $u(Y) = v(Y)$. Như vậy $u(X \cup Y) = v(X \cup Y)$. Mâu thuẫn với giả thiết. Vậy, $[t_1]_{X \cup Y} = [t_2]_X$ hay $IND(X \cup Y) = IND(X)$.

2) $\Rightarrow$ 3). Lấy bất kỳ một bộ $t \in R$. Rõ ràng, tồn tại một lớp tương đương $[t]_Y \in R/Y$ sao cho $[t]_Y \supseteq [t]_{X \cup Y} = [t]_X$. Suy ra $[t]_X \subseteq \underline{X}[t]_Y \subseteq POS_X(Y)$. Như vậy $t \in POS_X(Y)$. Hơn nữa chúng ta luôn có $POS_X(Y) \subseteq R$.

3) $\Rightarrow$ 4). Giả sử $u \in [t]_Y$. Vì $[t]_Y \subseteq R$ nên $u \in POS_X(Y)$. Suy ra tồn tại một $[v]_Y \in R/Y$ sao cho $u \in \underline{X}[v]_Y$. Do đó $[t]_Y = [u]_Y = [v]_Y$. Như vậy $u \in \underline{X}[t]_Y$. Ngoài ra chúng ta luôn có $X[t]_Y \subseteq [t]_Y$ với mọi $[t]_Y \in R/Y$.

4) $\Rightarrow$ 1). Lấy bất kỳ hai bộ $t_1, t_2 \in R$ sao cho $t_1(X) = t_2(X)$. Suy ra $\underline{X}[t_1]_Y = [t_1]_Y$ và $\underline{X}[t_2]_Y = [t_2]_Y$. Vì $t_1 \in \underline{X}[t_1]_Y$, $t_2 \in \underline{X}[t_2]_Y$, nên $[t_1]_X \subseteq [t_1]_Y$ và $[t_2]_X \subseteq [t_2]_Y$. Do đó $[t_1]_Y = [t_2]_Y$, tức là $t_1(Y) = t_2(Y)$. $\square$

Như vậy, tập thuộc tính Y phụ thuộc hoàn toàn (tức ứng bộ phận) vào tập thuộc tính X nếu tất cả (tương ứng một số) các bộ của quan hệ R có thể phân lớp một cách duy nhất vào các lớp tương đương trong R/Y bằng cách dùng X .

5.4.2.2. Một số tính chất cơ bản của phụ thuộc cấp k

Dựa vào tính chất của miền khẳng định $POS_X(Y)$, chúng ta suy ra được một số tính chất cơ bản sau của phụ thuộc cấp $X \xrightarrow{k} Y$.

Mệnh đề 5.13. $POS_X(Y) \subseteq POS_{X \cup Z}(Y \cup Z)$ với mọi $X, Y, Z \subseteq U$.

Chứng minh. Xét một bộ bất kỳ $t \in POS_X(Y)$. Khi đó, tồn tại $[t]_X \in R/X$ và $[t]_Y \in R/Y$ sao cho $t \in [t]_X$ và $[t]_X \subseteq [t]_Y$. Do đó $t \in [t]_Y$. Lúc này với mọi $u \in [t]_X, v \notin [t]_X$, ta có

$$\begin{cases} u(X) &= t(X) \\ u(Y) &= t(Y) \\ v(X) &\neq t(X) \end{cases}$$

Bây giờ xét tập thương $R/(X \cup Z)$, ta có $[t]_X = \bigcup [t]_{X \cup Z}$ với một số $[t]_{X \cup Z} \in R/(X \cup Z)$ và $w(X) = t(X)$ với mọi $w \in [t]_{X \cup Z}$. Khi đó phải tồn tại một lớp tương đương nào đó $[t]_{X \cup Z} \in R/(X \cup Z)$ sao cho $t \in [t]_{X \cup Z}$, và với mọi $u \in [t]_{X \cup Z}$ thì $u(X \cup Z) = t(X \cup Z)$. Suy ra

$$\begin{cases} u(Z) &= t(Z) \\ u(Y) &= t(Y) \end{cases}$$

Hay

$$u(Y \cup Z) = t(Y \cup Z).$$

Nghĩa là với mọi $u \in [t]_{X \cup Z}$, ta có $u \in [t]_{Y \cup Z}$ với $[t]_{Y \cup Z} \in R/(Y \cup Z)$. Như vậy, $t \in [t]_{X \cup Z}$ và $[t]_{X \cup Z} \subseteq [t]_{Y \cup Z}$. Do đó, theo định nghĩa $t \in POS_{X \cup Z}(Y \cup Z)$. $\square$

Hệ quả 5.3. (Tính gia tăng) Nếu $X \xrightarrow{k} Y$ thì $X \cup Z \xrightarrow{k'} Y \cup Z$ với $k' \geq k$.

Chứng minh. Theo Mệnh đề 5.13 ta có $|POS_X(Y)| \leq |POS_{X \cup Z}(Y \cup Z)|$. Suy ra $k' \geq k$. $\square$

Mệnh đề 5.14. Nếu $POS_X(Y) = R$ thì $POS_Y(Z) \subseteq POS_X(Z)$ với mọi $X, Y, Z \subseteq U$.

Chứng minh. Lấy một bộ bất kỳ $t \in POS_Y(Z)$. Theo định nghĩa, tồn tại $[t]_Y \in R/Y$ và $[t]_Z \in R/Z$ sao cho $t \in [t]_Y$ và $[t]_Y \subseteq [t]_Z$. Suy ra $t \in R$, và do đó $t \in POS_X(Y)$. Điều này có nghĩa, tồn tại $[t]_X \in R/X$ sao cho $t \in [t]_X$ và $[t]_X \subseteq [t]_Y$. Như vậy, $t \in [t]_X$ và $[t]_X \subseteq [t]_Z$. Nên $t \in POS_X(Z)$. $\square$

Hệ quả 5.4. (Tính bắc cầu yếu 1) Nếu $X \rightarrow Y$ và $Y \xrightarrow{k} Z$ thì $X \xrightarrow{k'} Z$ với $k' \geq k$.

Mệnh đề 5.15. Nếu $POS_X(Y) = R$ thì với mọi $[t]_x \in R/X$, tồn tại $[t]_y \in R/Y$ sao cho $[t]_x \subseteq [t]_y$.

Chứng minh. Giả sử $[t]_X \in R/X$ và $t \in [t]_X$. Suy ra $t \in R$, và do đó $t \in POS_X(Y)$. Vậy nên, tồn tại $[t]_Y \in R/Y$ sao cho $[t]_X \subseteq [t]_Y$. $\square$

Mệnh đề 5.16. Nếu $POS_Y(Z) = R$ thì $POS_X(Y) \subseteq POS_X(Z)$ với mọi $X, Y, Z \subseteq U$.

Chứng minh. Giả sử $t \in POS_X(Y)$. Suy ra, tồn tại $[t]_X \in R/X$ và $[t]_Y \in R/Y$ sao cho $t \in [t]_X$ và $[t]_X \subseteq [t]_Y$. Vì $POS_Y(Z) = R$, nên theo Mệnh đề 5.15, tồn tại $[t]_Z \in R/Z$ sao cho $[t]_Y \subseteq [t]_Z$. Như vậy, $t \in [t]_X$ và $[t]_X \subseteq [t]_Z$. Theo định nghĩa $t \in POS_X(Z)$. $\square$

Hệ quả 5.5. (Tính bắc cầu yếu 2) Nếu $X \xrightarrow{k} Y$ và $Y \rightarrow Z$ thì $X \xrightarrow{k'} Z$ với $k' \geq k$.

Hệ quả 5.6. (Tính thu hẹp phải) Nếu $X \xrightarrow{k} Y \cup Z$ thì $X \xrightarrow{k'} Y$ với $k' \geq k$, và $X \xrightarrow{k''} Z$ với $k'' \geq k$.

Chứng minh. Suy ra từ Hệ quả 5.5 và tính phản xạ của hệ tiên đề Armstrong. $\square$

Ngoài ra chúng ta dễ thấy (tính phản xạ) nếu $Y \subseteq X$ thì $X \xrightarrow{k} Y$ với $k = 1$.

5.4.2.3. Rút gọn thuộc tính

Xét hai tập thuộc tính $X, Y \subseteq U$. Có thể xảy ra trường hợp Y không những phụ thuộc vào X mà còn phụ thuộc (cùng độ phụ thuộc) vào tập con $Z \subset X$. Do đó, việc đi tìm tập con Z này thật sự có ý nghĩa và thú vị. Điều này dẫn chúng ta đến khái niệm rút gọn thuộc tính sau đây.

Thuộc tính $a \in X$ được gọi là Y -không cần thiết trong X nếu $POS_X(Y) = POS_{X \setminus \{a\}}(Y)$, ngược lại a được gọi là Y -cần thiết trong X . Trường hợp nếu mọi thuộc tính của X đều Y -cần thiết trong X thì X được gọi là Y -độc lập, ngược lại X được gọi là Y -phụ thuộc. Tập con thuộc tính $Z \subseteq X$ được gọi là một Y -rút gọn của X nếu Z là Y -độc lập và $POS_X(Y) = POS_Z(Y)$.

Dễ thấy, X có thể có nhiều Y -rút gọn. Ký hiệu $Red_Y(X)$ là tập tất cả các Y -rút gọn của X . Như vậy, một Y -rút gọn của X là một tập con thuộc tính tối thiểu của X cho phép phân loại các bộ của quan hệ R vào đúng các lớp tương đương trong R/Y .

Tập tất cả các thuộc tính Y -cần thiết trong X được gọi là Y -lõi của X và ký hiệu $Core_Y(X)$.

Sau đây là mối tương quan của các rút gọn và lõi.

Mệnh đề 5.17. $Core_Y(X) = \bigcap Red_Y(X)$.

Chứng minh. Giả sử Z là một Y -rút gọn bất kỳ của X và $a \in X \setminus Z$. Suy ra $POS_X(Y) = POS_Z(Y)$ và $Z \subseteq X \setminus \{a\} \subseteq X$. Do đó $POS_X(Y) = POS_{X \setminus \{a\}}(Y)$. Điều này có nghĩa, thuộc tính a là Y -không cần thiết trong X hay $a \notin Core_Y(X)$. Như vậy, $Core_Y(X) \subseteq Z$ với mọi $Z \in Red_Y(X)$, hay $Core_Y(X) \subseteq \bigcap Red_Y(X)$.

Trường hợp nếu thuộc tính a là Y -không cần thiết trong X thì $POS_X(Y) = POS_{X \setminus \{a\}}(Y)$. Suy ra, tồn tại một tập con thuộc tính Y -độc lập $Z \subseteq X \setminus \{a\}$ sao cho $POS_Z(Y) = POS_X(Y)$. Do đó $Z \in Red_Y(X)$ và $a \notin Z$. Vậy nên, $Z \subseteq Core_Y(X)$ hay $\bigcap Red_Y(X) \subseteq Core_Y(X)$. $\square$

Như vậy, mỗi thuộc tính của lỗi là thuộc một tập rút gọn nào đó. Do đó, theo một nghĩa nào đó thì lỗi là tập thuộc tính quan trọng nhất, không có bất kỳ thuộc tính nào của nó có thể bỏ đi mà không ảnh hưởng đến sự phân lớp của các thuộc tính.

Ví dụ 5.11. Xét quan hệ $R = \{t_1, t_2, \dots, t_6\} \in Rel(U)$ với $U = \{a, b, c, d\}$ như sau:

a	b	c	d
0	1	3	1
1	0	3	1
1	1	4	1
0	1	2	0
1	0	3	0
0	1	4	1

Xét $X = \{a, b, c\}$ và $Y = \{d\}$. Ta có $R/X = \{\{t_1\}, \{t_2, t_5\}, \{t_3\}, \{t_4\}, \{t_6\}\}$, $R/Y = \{\{t_1, t_2, t_3, t_6\}, \{t_4, t_5\}\}$. Suy ra $\underline{X}\{t_1, t_2, t_3, t_6\} = \{t_1, t_3, t_6\}$, $\underline{X}\{t_4, t_5\} = \{t_4\}$. Do đó $POS_X(Y) = \{t_1, t_3, t_4, t_6\}$.

Với $Z_1 = X \setminus \{a\}$ thì $R/Z_1 = \{\{t_1\}, \{t_2, t_5\}, \{t_3, t_6\}, \{t_4\}\}$. Lúc đó, $\underline{Z_1}\{t_1, t_2, t_3, t_6\} = \{t_1, t_3, t_6\}$, $\underline{Z_1}\{t_4, t_5\} = \{t_4\}$. Nên $POS_{Z_1}(Y) = \{t_1, t_3, t_4, t_6\}$. Như vậy, thuộc tính a là Y -không cần thiết trong X . Dễ dàng kiểm tra được các thuộc tính còn lại b, c đều Y -cần thiết trong X . Vậy $Z_1 \in Red_Y(X)$.

Trường hợp với $Z_2 = X \setminus \{b\}$ thì $R/Z_2 = \{\{t_1\}, \{t_2, t_5\}, \{t_3\}, \{t_4\}, \{t_6\}\}$. Rõ ràng thuộc tính b là Y -không cần thiết trong X và các thuộc tính còn lại a, c đều Y -cần thiết trong X . Nên $Z_2 \in Red_Y(X)$.

Có thể kiểm chứng được các tập con thuộc tính còn lại của X không phải Y -rút gọn của X . Như vậy, $Red_Y(X) = \{Z_1, Z_2\}$. Tức là thuộc tính c là thuộc tính lỗi (tối thiểu) cần thiết của X trong việc khẳng định Y .

Lưu ý, bài toán tìm tất cả các Y -rút gọn của X là bài toán khó, có độ phức tạp thời gian là hàm mũ theo số thuộc tính. Trong khuôn giáo trình CSDL chúng ta sẽ không đi sâu vào các vấn đề này.

5.4.3. Phụ thuộc hàm xấp xỉ

5.4.3.1. Định nghĩa

Cho quan hệ $R \in Rel(U)$ và $X \rightarrow Y$ là một PTH trên U . Xét $R_1 \subseteq R$ là quan hệ có số bộ ít nhất cần loại bỏ khỏi R để PTH $X \rightarrow Y$ đúng trên quan hệ $R \setminus R_1$. Khi đó, tỷ số của $|R_1|$ và $|R|$ được gọi là *độ đo lỗi* của PTH $X \rightarrow Y$ trên R và ký hiệu $e(X \rightarrow Y, R)$.

Tức là

$$e(X \rightarrow Y, R) = \frac{\min \{|R_1| : R_1 \subseteq R, (R \setminus R_1)(X \rightarrow Y)\}}{|R|}.$$

Một cách tương đương

$$e(X \rightarrow Y, R) = 1 - \frac{\max \{|R_2| : R_2 \subseteq R, R_2(X \rightarrow Y)\}}{|R|}$$

trong đó quan hệ $R_2 \subseteq R$ và R_2 có số bộ nhiều nhất thỏa PTH $X \rightarrow Y$. Độ đo lỗi định nghĩa ở trên còn được gọi là *độ đo lỗi e* . Như vậy, chúng ta có thể hiểu giữa hai tập thuộc tính bất kỳ X, Y trên quan hệ R luôn có một độ đo lỗi e , độ đo này đo sự phụ thuộc giữa X và Y . Trong trường hợp PTH $X \rightarrow Y$ đúng trên R thì rõ ràng độ đo lỗi $e(X \rightarrow Y, R) = 0$.

Cho ngưỡng lỗi ϵ , với $\epsilon \in [0, 1]$, ta nói PTH $X \rightarrow Y$ là một *PTH xấp xỉ đúng* trên R ứng với ngưỡng lỗi ϵ nếu $e(X \rightarrow Y, R) \leq \epsilon$. Lúc này ta còn nói R *thỏa* PTH xấp xỉ $X \rightarrow Y$ ứng với ngưỡng lỗi ϵ và ký hiệu $R(X \rightarrow Y, \epsilon)$. Rõ ràng, nếu $R(X \rightarrow Y)$ thì $R(X \rightarrow Y, \epsilon)$ với mọi ngưỡng lỗi ϵ . Trường hợp nếu R không thỏa PTH xấp xỉ $X \rightarrow Y$ ứng với ngưỡng lỗi ϵ thì ta viết $R(\neg X \rightarrow Y, \epsilon)$, tức là ngưỡng lỗi $e(X \rightarrow Y, R) > \epsilon$. Ký hiệu $AFD(R, \epsilon)$ là tập tất cả các PTH xấp xỉ đúng trên R ứng với ngưỡng lỗi ϵ .

Ví dụ 5.12. Xét quan hệ $R = \{t_1, t_2, \dots, t_7\}$ trên tập $U = \{a, b, c, d\}$:

a	b	c	d
0	0	1	1
0	1	1	1
0	2	1	2
1	2	0	0
2	0	0	0
2	0	2	0
1	1	2	1

Xét PTH $ab \rightarrow c$. Rõ ràng, bỏ bộ (hàng) 6 hoặc 5 ra khỏi R thì ta có $R(ab \rightarrow c)$. Như vậy, số bộ ít nhất cần loại khỏi quan hệ R để PTH $ab \rightarrow c$ đúng trên các bộ còn lại là một. Suy ra, $e(ab \rightarrow c, R) = 1/7$. Hoặc lập luận tương đương, số bộ lớn nhất trong quan hệ R để PTH là đúng trên các bộ đó là sáu. Do đó $e(ab \rightarrow c, R) = 1 - 6/7 = 1/7$. Trường hợp với PTH $b \rightarrow a$, thì lập luận tương tự ta có $e(b \rightarrow a, R) = 3/7$.

Bây giờ, ta giả sử ngưỡng lỗi $\epsilon = 1/7$. Khi đó, $e(ab \rightarrow c, R) \leq \epsilon$ và $e(b \rightarrow a, R) > \epsilon$. Như vậy, $R(ab \rightarrow c, \epsilon)$ và $R(\neg b \rightarrow a, \epsilon)$. Nghĩa là, $ab \rightarrow c$ là PTH xấp xỉ đúng trên R ứng với ngưỡng lỗi ϵ , còn $b \rightarrow a$ là PTH xấp xỉ không đúng trên R ứng với ngưỡng lỗi ϵ .

Các tính chất sau là rõ ràng.

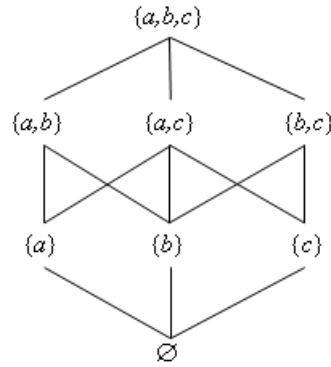
Bổ đề 5.7. *Giả sử $R(X \setminus \{b\} \rightarrow b)$, với $b \in X$. Khi đó*

- 1) *Nếu X là một khóa của R thì $X \setminus \{b\}$ cũng vậy.*
- 2) *Nếu $R(X \rightarrow a, \epsilon)$ thì $R(X \setminus \{b\} \rightarrow a, \epsilon)$.*

Lúc này có nghĩa PTH xấp xỉ $X \rightarrow a$ không tối tiểu. Lưu ý, khái niệm tối tiểu không thay đổi khi xét cho PTH xấp xỉ. Ta biết rằng $X \rightarrow Y \in F^+$ khi và chỉ khi $X \rightarrow a$ với mọi $a \in Y$. Do đó không mất tính tổng quát, trong mục này chúng ta chỉ xét các PTH có dạng $X \rightarrow a$.

Khai phá (hay suy diễn, phát hiện) các PTH xấp xỉ là vấn đề được các nhà khoa học máy tính tập trung nghiên cứu trong những năm gần đây, thuộc một trong những mục tiêu của lĩnh vực khai phá dữ liệu.

Bài toán *khai phá PTH xấp xỉ* được phát biểu như sau: “Cho quan hệ $R \in Rel(U)$ và ngưỡng lỗi ϵ . Tìm tất cả các PTH xấp xỉ không tầm thường tối tiểu $X \rightarrow a$ đúng trên R ứng với ngưỡng lỗi ϵ ”. Dễ dàng thấy bài toán này có thể giải bằng thuật toán cơ bản với ý tưởng như sau: chúng ta bắt đầu từ tập một thuộc tính ở mức L_1 thông qua dần tìm kiếm $\mathcal{P}(U)$, ở đây mức L_k là tập tất cả các tập thuộc tính có kích thước bằng k . Chẳng hạn, sau đây là dần tìm kiếm $\mathcal{P}(U)$ với tập thuộc tính $U = \{a, b, c\}$:



Trong dần này một cung giữa X và $X \cup \{a\}$ biểu diễn một PTH không tầm thường (trên U) dạng $X \rightarrow a$. Lúc này giả sử nếu chúng ta đang xử lý tập thuộc tính X thì thuật toán sẽ kiểm tra các phụ thuộc dạng $X \setminus \{a\} \rightarrow a$ với $a \in X$, xem có phải là các PTH xấp xỉ đúng trên R ứng với ngưỡng lỗi ϵ cho trước. Tuy nhiên, để giảm chi phí tính toán thì chúng ta có thể sử dụng các công cụ tập thương và tập dự tuyển về phía sau đây. Lúc này những tính toán tại mỗi mức bằng cách sử dụng những kết quả ở mức trước và cũng như cắt tỉa được không gian tìm kiếm của thuật toán.

5.4.3.2. Tập thương của quan hệ

Xét quan hệ không phân biệt $IND(X)$ và lớp tương đương $[t]_X \in R/X$. Ta gọi các bộ trong lớp tương đương $[t]_X$ là *tương đương* với nhau trên X . Nói cách khác, với hai bộ bất $t_1, t_2 \in R$ là tương đương trên X nếu

và chỉ nếu $t_1(X) = t_2(X)$. Như vậy, nếu hai bộ tương đương trên tập thuộc tính X , thì chúng sẽ không phân biệt được trên X . Lúc này, *tập thương* của quan hệ R theo quan hệ $IND(X)$ xác định bởi:

$$R/X = \{[t]_X : t \in R\}.$$

Người ta còn gọi tập thương R/X là một *phân hoạch* của quan hệ R sinh bởi X . Rõ ràng, nếu X là một khóa hay khóa tối thiểu của quan hệ R thì R/X chỉ bao gồm các lớp tương đương đơn tử (gồm một phần tử). Chẳng hạn, xét quan hệ R trong Ví dụ 5.12 ta có:

$$\begin{aligned} R/\{a\} &= \{\{t_1, t_2, t_3\}, \{t_4, t_7\}, \{t_5, t_6\}\} \\ R/\{b\} &= \{\{t_1, t_5, t_6\}, \{t_2, t_7\}, \{t_3, t_4\}\} \\ R/\{c\} &= \{\{t_1, t_2, t_3\}, \{t_4, t_5\}, \{t_6, t_7\}\} \\ R/\{a, b\} &= \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5, t_6\}, \{t_7\}\} \\ R/\{a, c\} &= \{\{t_1, t_2, t_3\}, \{t_4\}, \{t_5\}, \{t_6\}, \{t_7\}\} \\ R/\{b, c\} &= \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5\}, \{t_6\}, \{t_7\}\} \\ R/\{a, b, c\} &= \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5\}, \{t_6\}, \{t_7\}\}. \end{aligned}$$

Dễ thấy $\{b, c\} \in Key(R)$ và $\{a, b, c\}$ là một khóa của R .

Xét hai tập thương R/X và R/Y . Ta nói tập thương R/X *mịn* hơn tập thương R/Y nếu mỗi lớp tương đương của R/X là chứa trong một lớp tương đương nào đó của R/Y . Nghĩa là, với mỗi $[t]_X \in R/X$, tồn tại một $[t]_Y \in R/Y$ sao cho $[t]_X \subseteq [t]_Y$. Như vậy, nếu R/X mịn hơn R/Y thì rõ ràng $|R/X| \geq |R/Y|$. Chẳng hạn, với quan hệ R trong Ví dụ 5.12 ta có tập thương $R/\{b, c\}$ mịn hơn tập thương $R/\{a\}$, nhưng tập thương $R/\{b\}$ không mịn hơn tập thương $R/\{a\}$. Để ý nếu R/X mịn hơn R/Y và ngược lại, thì rõ ràng $R/X = R/Y$.

Mệnh đề 5.18. $R(X \rightarrow a)$ khi và chỉ khi R/X mịn hơn $R/\{a\}$.

Chứng minh. Nếu $R(X \rightarrow a)$ thì rõ ràng R/X mịn hơn $R/\{a\}$ theo định nghĩa.

Ngược lại, giả sử R/X mịn hơn $R/\{a\}$. Lúc đó, ta lấy bất kỳ một lớp tương đương $[t]_X \in R/X$. Vì các bộ tương đương với t trên X là phải ở trong $[t]_X$, nên theo giả thiết suy ra các bộ tương đương này cũng phải ở trong $[t]_{\{a\}}$. Do đó, các bộ này có giá trị giống nhau trên $\{a\}$. Suy ra, các bộ này thỏa PTH $X \rightarrow a$, và do đó $R(X \rightarrow a)$. $\square$

Như vậy, rõ ràng $R(X \rightarrow Y)$ nếu và chỉ nếu R/X mịn hơn R/Y , và do đó nếu $Y \subseteq X$ thì R/X mịn hơn R/Y .

Mệnh đề 5.19. $R(X \rightarrow a)$ khi và chỉ khi $|R/X| = |R/(X \cup \{a\})|$.

Chứng minh. Biết rằng nếu $R(X \rightarrow a)$ thì $R(X \rightarrow X \cup \{a\})$. Suy ra, R/X mịn hơn $R/(X \cup \{a\})$. Hơn nữa, chúng ta luôn có $R/(X \cup \{a\})$ mịn hơn R/X . Như vậy, rõ ràng $R/X = R/(X \cup \{a\})$, và do đó $|R/X| = |R/(X \cup \{a\})|$.

Ngược lại, giả sử $|R/X| = |R/(X \cup \{a\})|$. Hơn nữa biết rằng, $R/(X \cup \{a\})$ luôn mịn hơn R/X , nên ta phải có $R/X = R/(X \cup \{a\})$. Suy ra, PTH $X \rightarrow a$ phải đúng trên quan hệ R . $\square$

5.4.3.3. Biểu diễn độ đo lỗi $e(X \rightarrow a, R)$ qua tập thương

Xét PTH $X \rightarrow a$ trên quan hệ R . Độ đo lỗi $e(X \rightarrow a, R)$ có thể được đặc trưng qua tập thương của quan hệ R như sau.

Mệnh đề 5.20.

$$e(X \rightarrow a, R) = 1 -$$

$$\frac{\sum_{[t]_X \in R/X} \max \left\{ |[t]_{X \cup \{a\}}| : [t]_{X \cup \{a\}} \in R/(X \cup \{a\}), [t]_{X \cup \{a\}} \subseteq [t]_X \right\}}{|R|}.$$

Chứng minh. Như chúng ta đã biết $R/(X \cup \{a\})$ mịn hơn R/X , do đó bất kỳ lớp tương đương $[t]_X \in R/X$ cũng là hợp của một số lớp tương đương trong $R/(X \cup \{a\})$. Lúc đó, để $R(X \rightarrow a)$ thì với mỗi $[t]_X \in R/X$, ta phải loại bỏ các bộ tương ứng với các lớp tương đương trong họ

$$\mathcal{E} = \{[t]_{X \cup \{a\}} \in R/(X \cup \{a\}) : [t]_{X \cup \{a\}} \subseteq [t]_X\},$$

và chỉ giữ lại các bộ của lớp tương đương có số phần tử lớn nhất trong $\mathcal{E}$. Suy ra, số bộ ít nhất cần loại bỏ khỏi R để PTH $R(X \rightarrow a)$ là

$$|R| - \sum_{[t]_X \in R/X} \max \left\{ |[t]_{X \cup \{a\}}| : [t]_{X \cup \{a\}} \in R/(X \cup \{a\}), [t]_{X \cup \{a\}} \subseteq [t]_X \right\}$$

Như vậy

$$e(X \rightarrow a, R) = 1 - \frac{\sum_{[t]_X \in R/X} \max \left\{ |[t]_{X \cup \{a\}}| : [t]_{X \cup \{a\}} \in R/(X \cup \{a\}), [t]_{X \cup \{a\}} \subseteq [t]_X \right\}}{|R|}$$

□

Chẳng hạn, xét quan hệ R trong Ví dụ 5.12. Khi đó, độ đo lỗi $e(b \rightarrow a, R)$ được đặc trưng qua tập thương như sau:

$$R/\{b\} = \{\{t_1, t_5, t_6\}, \{t_2, t_7\}, \{t_3, t_4\}\}$$

$$R/\{a, b\} = \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5, t_6\}, \{t_7\}\}.$$

Với các lớp tương đương trong $R/\{b\}$:

- $\{t_1, t_5, t_6\}$ thì

$$\begin{aligned} \max \left\{ |[t]_{\{a, b\}}| : [t]_{\{a, b\}} \in R/\{a, b\}, [t]_{\{a, b\}} \subseteq \{t_1, t_5, t_6\} \right\} &= |\{t_5, t_6\}| \\ &= 2. \end{aligned}$$

- $\{t_2, t_7\}$ thì

$$\begin{aligned} \max \left\{ |[t]_{\{a, b\}}| : [t]_{\{a, b\}} \in R/\{a, b\}, [t]_{\{a, b\}} \subseteq \{t_2, t_7\} \right\} &= |\{t_2\}| \\ &= 1. \end{aligned}$$

- $\{t_3, t_4\}$ thì

$$\begin{aligned} \max \left\{ | [t]_{\{a,b\}} | : [t]_{\{a,b\}} \in R / \{a,b\}, [t]_{\{a,b\}} \subseteq \{t_3, t_4\} \right\} &= | \{t_3\} | \\ &= 1. \end{aligned}$$

Suy ra

$$e(b \rightarrow a, R) = 1 - \frac{4}{7} = \frac{3}{7}.$$

Từ Mệnh đề 5.20, chúng ta thấy ngay được mối liên hệ giữa độ đo lỗi $e(X \rightarrow a, R)$ và độ phụ thuộc k .

Hệ quả 5.7.

$$e(X \rightarrow a, R) + k = 1 - \frac{\sum_{[t]_X \in R/X} \max \left\{ | [t]_{X \cup \{a\}} | : [t]_{X \cup \{a\}} \in R / (X \cup \{a\}), [t]_{X \cup \{a\}} \subset [t]_X \right\}}{|R|}.$$

Chứng minh.

Theo Mệnh đề 5.20 ta có

$$\begin{aligned} e(X \rightarrow a, R) &= 1 - \\ &\frac{\sum_{[t]_X \in R/X} \max \left\{ | [t]_{X \cup \{a\}} | : [t]_{X \cup \{a\}} \in R / (X \cup \{a\}), [t]_{X \cup \{a\}} \subseteq [t]_X \right\}}{|R|} \\ &= 1 - \\ &\frac{\sum_{[t]_X \in R/X} \max \left\{ | [t]_{X \cup \{a\}} | : [t]_{X \cup \{a\}} \in R / (X \cup \{a\}), [t]_{X \cup \{a\}} \subset [t]_X \right\}}{|R|} \\ &\quad - \frac{\sum_{[t]_X \in R/X} \max \left\{ | [t]_{X \cup \{a\}} | : [t]_{X \cup \{a\}} \in R / (X \cup \{a\}), [t]_{X \cup \{a\}} = [t]_X \right\}}{|R|} \end{aligned}$$

$$= 1 - \frac{\sum_{[t]_X \in R/X} \max \left\{ |[t]_{X \cup \{a\}}| : [t]_{X \cup \{a\}} \in R/(X \cup \{a\}), [t]_{X \cup \{a\}} \subset [t]_X \right\}}{|R|} - \frac{|\bigcup \{[t]_X \in R/X : [t]_X (X \rightarrow a)\}|}{|R|}$$

$$= 1 - \frac{\sum_{[t]_X \in R/X} \max \left\{ |[t]_{X \cup \{a\}}| : [t]_{X \cup \{a\}} \in R/(X \cup \{a\}), [t]_{X \cup \{a\}} \subset [t]_X \right\}}{|R|} - \frac{POS_X(\{a\})}{|R|}$$

$$= 1 - k - \sum_{[t]_X \in R/X} \max \left\{ |[t]_{X \cup \{a\}}| : [t]_{X \cup \{a\}} \in R/(X \cup \{a\}), [t]_{X \cup \{a\}} \subset [t]_X \right\}.$$

Như vậy

$$e(X \rightarrow a, R) + k = 1 -$$

$$\frac{\sum_{[t]_X \in R/X} \max \left\{ |[t]_{X \cup \{a\}}| : [t]_{X \cup \{a\}} \in R/(X \cup \{a\}), [t]_{X \cup \{a\}} \subset [t]_X \right\}}{|R|}$$

□

Chẳng hạn, xét quan hệ R ở Ví dụ 5.12. Ta đã biết $e(b \rightarrow a, R) = \frac{3}{7}$ và các tập thương

$$R/\{a\} = \{\{t_1, t_2, t_3\}, \{t_4, t_7\}, \{t_5, t_6\}\}$$

$$R/\{b\} = \{\{t_1, t_5, t_6\}, \{t_2, t_7\}, \{t_3, t_4\}\}$$

$$R/\{a, b\} = \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5, t_6\}, \{t_7\}\}.$$

Suy ra $POS_{\{b\}}(\{a\}) = \emptyset$ và do đó $k = 0$.

Ngoài ra

$$\frac{\sum_{[t]_{\{b\}} \in R/\{b\}} \max \left\{ |[t]_{\{a,b\}}| : [t]_{\{a,b\}} \in R/\{a,b\}, [t]_{\{a,b\}} \subset [t]_{\{b\}} \right\}}{|R|} =$$

$$= \frac{2+1+1}{7} = \frac{4}{7}.$$

Như vậy $e(b \rightarrow a, R) + k = 1 - \frac{4}{7} = \frac{3}{7}$.

5.4.3.4. Tập thương thu gọn và cận của độ đo lỗi $g_3(X \rightarrow a, R)$

Để giảm tính toán khi làm việc với các tập thương, chúng ta có thể sử dụng tập thương thu gọn sau đây. Cho quan hệ $R \in Rel(U)$ và tập con thuộc tính $X \subseteq U$. Xét quan hệ con $R_1 \subseteq R$ là quan hệ có số bộ ít nhất cần loại bỏ khỏi R để X là một khóa của $R \setminus R_1$ (tức là $(R \setminus R_1)(X \rightarrow U)$). Lúc đó, tỷ số của $|R_1|$ và $|R|$ được gọi là *độ đo lỗi của X trên R* và ký hiệu là $e(X, R)$.

Như vậy

$$e(X, R) = e(X \rightarrow U, R).$$

Khi $e(X, R) \leq \epsilon$ (nhỏ) thì ta nói X là *khóa xấp xỉ* và $e(X \rightarrow U, R)$ là *độ đo lỗi khóa e* . Vì $1 \leq |R/X| \leq |R|$, nên $e(X, R) \in [0, 1]$. Dễ thấy, nếu X là một khóa của R thì $e(X, R) = 0$.

Xét Ví dụ 5.12. Chẳng hạn, ta có $e(\{a, b\}, R) = \frac{1}{7}$, $e(\{b, c\}, R) = 0$. Dễ dàng biểu diễn được độ đo lỗi $e(X, R)$ qua tập thương như sau.

Hệ quả 5.8.

$$e(X, R) = 1 - \frac{|R/X|}{|R|}.$$

Chứng minh. Vì U là một khóa của R , nên R/U gồm các lớp tương

đương đơn tử. Khi đó, theo Mệnh đề 5.20 suy ra

$$\begin{aligned}
 e(X, R) &= e(X \rightarrow U, R) \\
 &= 1 - \frac{\sum_{[t]_X \in R/X} \max \{ |[t]_U| : [t]_U \in R/U, [t]_U \subseteq [t]_X \}}{|R|} \\
 &= 1 - \frac{|R/X|}{|R|}.
 \end{aligned}$$

□

Hệ quả 5.9.

$$e(X, R) = e(Y, R) \Leftrightarrow |R/X| = |R/Y|.$$

Bây giờ xét tập thương R/X . Tập hợp tất cả các lớp tương đương có nhiều hơn một phần tử trong R/X được gọi là *tập thương thu gọn* của R/X , và ký hiệu là $\widehat{R/X}$. Tức là

$$\widehat{R/X} = \{[t]_X \in R/X : |[t]_X| > 1\}.$$

Như vậy, các lớp tương đương có đúng một phần tử trong tập thương bị loại bỏ trong tập thương thu gọn của nó. Với tập thương thu gọn, Mệnh đề 5.18 vẫn còn đúng. Tuy nhiên, Mệnh đề 5.19 và Mệnh đề 5.20 không còn đúng.

Ví dụ 5.13. Xét quan hệ $R = \{t_1, t_2, t_3\}$ trên tập thuộc tính $U = \{a, b, c, d\}$ như sau:

a	b	c	d
0	1	1	0
0	2	2	0
0	1	3	1

Ta có $R/\{a\} = \{\{t_1, t_2, t_3\}\}$, $R/\{a, b\} = \{\{t_1, t_3\}, \{t_2\}\}$.

Suy ra $\widehat{R/\{a\}} = \{\{t_1, t_2, t_3\}\}$, $\widehat{R/\{a, b\}} = \{\{t_1, t_3\}\}$. Vậy $|\widehat{R/\{a\}}| = |\widehat{R/\{a, b\}}|$ nhưng $R(\neg a \rightarrow b)$.

Tuy vậy, Mệnh đề 5.19 có thể thay bằng kết quả sau.

Mệnh đề 5.21. $R(X \rightarrow a)$ khi và chỉ khi $e(X, R) = e(X \cup \{a\}, R)$ với

$$e(X, R) = \frac{\|\widehat{R/X}\| - |\widehat{R/X}|}{|R|}$$

trong đó $\|\widehat{R/X}\| = \sum_{[t]_X \in \widehat{R/X}} |[t]_X|$.

Chứng minh. Vì $R(X \rightarrow a)$ khi và chỉ khi $|R/X| = |R/(X \cup \{a\})|$. Khi đó

$$\begin{aligned} e(X, R) &= 1 - \frac{|R/X|}{|R|} \\ &= 1 - \frac{|R/(X \cup \{a\})|}{|R|} \\ &= e(X \cup \{a\}, R). \end{aligned}$$

Ở đây

$$\begin{aligned} e(X, R) &= 1 - \frac{|R/X|}{|R|} \\ &= \frac{|R| - (|R| - \|\widehat{R/X}\| + |\widehat{R/X}|)}{|R|} \\ &= \frac{\|\widehat{R/X}\| - |\widehat{R/X}|}{|R|}. \end{aligned}$$

□

Xét quan hệ R trong Ví dụ 5.12. Tính đúng của PTH $ab \rightarrow c$ trên R được khảo sát qua tập thương thu gọn như sau. Ta đã biết

$$R/\{a, b\} = \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5, t_6\}, \{t_7\}\}$$

$$R/\{a, b, c\} = \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5\}, \{t_6\}, \{t_7\}\}.$$

Suy ra $R/\widehat{\{a, b\}} = \{\{t_5, t_6\}\}$, $R/\widehat{\{a, b, c\}} = \emptyset$. Nên $\|R/\widehat{\{a, b\}}\| = 2$ và $\|R/\widehat{\{a, b, c\}}\| = 0$. Do đó $e(\{a, b\}, R) = \frac{1}{7}$, $e(\{a, b, c\}, R) = 0$. Vậy, $R(\neg ab \rightarrow c)$.

Tích của hai tập thương R/X và R/Y , ký hiệu $(R/X).(R/Y)$, là một họ các tập con các bộ của R xác định bởi:

$$(R/X).(R/Y) = \bigcup_{[t]_X \in R/X} \mathcal{Q}([t]_X)$$

với

$$\mathcal{Q}([t]_X) = \bigcup_{[t]_Y \in R/Y} \{[t]_X \cap [t]_Y : [t]_X \cap [t]_Y \neq \emptyset\}.$$

Mệnh đề 5.22.

$$(R/X).(R/Y) = R/(X \cup Y).$$

Chứng minh. Giả sử $E \in (R/X).(R/Y)$. Suy ra, tồn tại $[t]_X \in R/X$, $[t]_Y \in R/Y$ sao cho $[t]_X \cap [t]_Y = E$. Khi đó, với mọi $t_1, t_2 \in E$ thì $t_1, t_2 \in [t]_X$ và $t_1, t_2 \in [t]_Y$. Suy ra $t_1(X) = t_2(X)$, $t_1(Y) = t_2(Y)$ và do đó $t_1(X \cup Y) = t_2(X \cup Y)$. Điều này có nghĩa, tồn tại một $[t]_{X \cup Y} \in R/(X \cup Y)$ sao cho $t_1, t_2 \in [t]_{X \cup Y}$. Như vậy, ta có $E \subseteq [t]_{X \cup Y}$.

Trường hợp nếu $[t]_{X \cup Y} \in R/(X \cup Y)$, thì với mọi $t_1, t_2 \in [t]_{X \cup Y}$ ta có $t_1(X \cup Y) = t_2(X \cup Y)$. Suy ra $t_1(X) = t_2(X)$, $t_1(Y) = t_2(Y)$. Điều này có nghĩa, phải tồn tại hai lớp tương đương $[t]_X \in R/X$ và $[t]_Y \in R/Y$ sao cho $t_1, t_2 \in [t]_X$ và $t_1, t_2 \in [t]_Y$. Rõ ràng $t_1, t_2 \in [t]_X \cap [t]_Y$, và do đó $[t]_{X \cup Y} \subseteq [t]_X \cap [t]_Y$. Hơn nữa, $[t]_X \cap [t]_Y \in (R/X).(R/Y)$.

Tóm lại, chúng ta có $(R/X).(R/Y) = R/(X \cup Y)$. □

Như vậy, tích hai tập thương cũng là một tập thương. Xét quan hệ R trong Ví dụ 5.12. Ta đã biết

$$R/\{a\} = \{\{t_1, t_2, t_3\}, \{t_4, t_7\}, \{t_5, t_6\}\}.$$

$$R/\{b\} = \{\{t_1, t_5, t_6\}, \{t_2, t_7\}, \{t_3, t_4\}\}.$$

$$R/\{a, b\} = \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5, t_6\}, \{t_7\}\}.$$

Khi đó

$$(\mathcal{Q}\{t_1, t_2, t_3\}) = \{\{t_1\}, \{t_2\}, \{t_3\}\}.$$

$$\mathcal{Q}(\{t_4, t_7\}) = \{\{t_7\}, \{t_4\}\}.$$

$$\mathcal{Q}(\{t_5, t_6\}) = \{\{t_5, t_6\}\}.$$

Suy ra

$$\begin{aligned} (R/\{a\}).(R/\{b\}) &= \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}, \{t_5, t_6\}, \{t_7\}\} \\ &= R/\{a, b\}. \end{aligned}$$

5.4.3.5. Tập dự tuyển và phụ thuộc tối thiểu

Để giảm không gian tìm kiếm các PTH xấp xỉ không tầm thường và tối thiểu, chúng ta có thể sử dụng tập dự tuyển về phải sau đây. Xét tập con thuộc tính $X \subseteq U$ và ngưỡng lỗi cho trước ϵ . Tập dự tuyển về phải của X (trong R) ứng với ngưỡng lỗi ϵ , ký hiệu $C(X, \epsilon)$, là tập các thuộc tính $a \notin X$ hoặc $a \in X$ sao cho $R(\neg X \setminus \{a\} \rightarrow a, \epsilon)$. Một cách hình thức

$$C(X, \epsilon) = \{a \in X : R(\neg X \setminus \{a\} \rightarrow a, \epsilon)\} \cup (U \setminus X).$$

Hay còn biểu diễn lại

$$C(X, \epsilon) = U \setminus \{a \in X : R(X \setminus \{a\} \rightarrow a, \epsilon)\}.$$

Chẳng hạn, xét Ví dụ 5.13 và ngưỡng lỗi $\epsilon = 1/2$, ta có $C(\{a, b\}, \epsilon) = \{b, c, d\}$, $C(\{b, c\}, \epsilon) = \{a, c, d\}$. Kết quả sau cho chúng ta mối quan hệ giữa tập dự tuyển về phải và PTH tối thiểu.

Định lý 5.23. Cho $a \in X$ và $R(X \setminus \{a\} \rightarrow a, \epsilon)$. Khi đó, phụ thuộc $X \setminus \{a\} \rightarrow a$ là tối thiểu khi và chỉ khi với mọi $b \in X$ sao cho $a \in C(X \setminus \{b\}, \epsilon)$.

Chứng minh. Chứng minh phản chứng. Giả sử phụ thuộc $X \setminus \{a\} \rightarrow a$

không tối tiểu. Suy ra, tồn tại thuộc tính $b \in X \setminus \{a\}$ sao cho $R((X \setminus \{a\}) \setminus \{b\} \rightarrow a, \epsilon)$. Hay $a \in X \setminus \{b\}$ và $R((X \setminus \{b\}) \setminus \{a\} \rightarrow a, \epsilon)$. Theo định nghĩa ta có $a \notin C(X \setminus \{b\}, \epsilon)$. Điều này mâu thuẫn với giả thiết.

Ngược lại, giả sử tồn tại một thuộc tính $b \in X$ sao cho $a \notin C(X \setminus \{b\}, \epsilon)$. Suy ra $R((X \setminus \{b\}) \setminus \{a\} \rightarrow a, \epsilon)$. Mâu thuẫn với giả thiết phụ thuộc $X \setminus \{a\} \rightarrow a$ là tối tiểu. $\square$

Chẳng hạn, xét quan hệ R ở Ví dụ 5.13 và $X = \{a, b, c\}$. Ta có $R(bc \rightarrow a)$. Vì $C(\{a, c\}, 0) = \{b, d, c\}$ nên $a \notin C(\{a, c\}, 0) = C(X \setminus \{b\}, 0)$. Suy ra, PTH $bc \rightarrow a$ là không tối tiểu. Trường hợp $X = \{b, c, d\}$, thì với $R(bd \rightarrow c)$ và $C(\{c, d\}, 0) = \{a, b, c\}$, $C(\{b, c\}, 0) = \{a, d, c\}$, $C(\{b, d\}, 0) = \{a, c, b, d\}$ nên PTH $bd \rightarrow c$ là tối tiểu. Trường hợp $X = \{a, b, d\}$, thì với $R(ad \rightarrow b, 1/3)$ ta có $b \notin C(\{a, b\}, 1/3) = C(X \setminus \{d\}, 1/3) = \{a, c, d\}$. Suy ra, PTH xấp xỉ $ad \rightarrow b$ không tối tiểu.

Định lý 5.24. *Giả sử X là một khóa của R và $a \in X$. Khi đó PTH $X \setminus \{a\} \rightarrow a$ là đúng và tối tiểu trên R khi và chỉ khi $X \setminus \{a\} \in \text{Key}(R)$ và với mọi $b \in X$ thì $a \in C(X \setminus \{b\})$.*

Chứng minh. Giả sử PTH $X \setminus \{a\} \rightarrow a$ đúng và tối tiểu trên R với $a \in X$. Vì X là một khóa của R nên $X \setminus \{a\}$ cũng như vậy. Hơn nữa, PTH $X \setminus \{a\} \rightarrow a$ là tối tiểu, suy ra $X \setminus \{a\} \in \text{Key}(R)$. Mặt khác, cũng từ giả sử PTH $X \setminus \{a\} \rightarrow a$ tối tiểu nên theo Định lý 5.23, ta có $a \in C(X \setminus \{b\}, \epsilon)$ với mọi $b \in X$.

Ngược lại, giả sử $X \setminus \{a\} \in \text{Key}(R)$ và với mọi $b \in X$ thì $a \in C(X \setminus \{b\})$. Suy ra $R(X \setminus \{a\} \rightarrow a)$. Bởi Định lý 5.23, ta có PTH $X \setminus \{a\} \rightarrow a$ là tối tiểu. $\square$

BÀI TẬP CHƯƠNG 5

5.1. Chứng minh các Bổ đề 5.3-5.6.

5.2. Xét một tập hữu hạn V và hai TTBD $L_1, L_2 \in \text{Closure}(V)$. Chứng minh tính đóng kín của lớp các TTBD đối với các phép toán sau:

a) (*Hạn chế*) Với $M \subseteq V$, ánh xạ hạn chế $L_1|_M(X) : \mathcal{P}(M) \rightarrow \mathcal{P}(M)$ xác định bởi $L_1|_M(X) = L_1(X) \cap M$ với mỗi $X \subseteq M$.

b) (*Hội*) Ánh xạ hội (của L_1 và L_2) $L_1 \wedge L_2 : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ xác định bởi $L_1 \wedge L_2(X) = L_1(X) \cap L_2(X)$ với mỗi $X \subseteq V$.

5.3. Xét hai tập hữu hạn phân biệt V_1, V_2 và tương ứng hai TTBD $L_1 \in \text{Closure}(V_1), L_2 \in \text{Closure}(V_2)$. Chứng minh *tích trực tiếp* của hai TTBD L_1 và L_2 , ký hiệu $L_1 \times L_2$, được định nghĩa như sau cũng là một TTBD:

$$L_1 \times L_2(X) = L_1(X \cap V_1) \cup L_2(X \cap V_2), \forall X \subseteq V_1 \cup V_2.$$

5.4. Cho một tập hữu hạn V và hai TTBD $L_1, L_2 \in \text{Closure}(V)$. Chứng minh lớp các TTBD không đóng kín đối với các phép toán sau:

a) (*Tuyển*) Ánh xạ tuyển (của L_1 và L_2) $L_1 \vee L_2 : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ xác định bởi $L_1 \vee L_2(X) = L_1(X) \cup L_2(X)$ với mỗi $X \subseteq V$.

b) (*Hợp thành*) Ánh xạ hợp thành (của L_1 và L_2) $L_1 L_2 : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ xác định bởi $L_1 L_2(X) = L_1(L_2(X))$ với mỗi $X \subseteq V$.

5.5. Cho tập hữu hạn V và TTBD $L \in \text{Closure}(V)$. Tập $X \subseteq V$ được gọi là *tập đóng* của L nếu $L(X) = X$. Ký hiệu $\text{Closed}(L)$ là tập tất cả tập đóng của L . Chứng minh rằng

a) $V \in L(X)$ và $L(X)$ là tập đóng nhỏ nhất chứa X .

b) Nếu X và Y là các tập đóng của L thì $L(X \cap Y) = L(X) \cap L(Y)$.

c) $\text{Closed}(L)$ đóng đối với phép toán giao: $X, Y \in \text{Closed}(L) \Rightarrow X \cap Y \in \text{Closed}(L)$.

5.6. Xét một tập hữu hạn V và một TTBD $L \in Closure(V)$. Ta định nghĩa hàm $H^L : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ như sau:

$$H^L(X) = X \cap L(V \setminus X) \quad (5.6)$$

với mỗi $X \subseteq V$.

a) Chứng minh H^L là một HC.

b) Chứng minh Công thức (5.6) xác định tương ứng 1-1 giữa TTBD và HC thỏa các điều kiện sau: với mọi $X, Y \subseteq V$

(i) Nếu $H^L(X) \subseteq Y \subseteq X$ thì $H^L(X) = H^L(Y)$.

(ii) Nếu $X \subseteq Y$ thì $H^L(Y) \cap X \subseteq H^L(X)$.

5.7. Xét một tập hữu hạn V và hai HC đặc biệt $H_1, H_2 \in Choice(V)$. Chứng minh *tuyển* của hai HC đặc biệt H_1 và H_2 , ký hiệu $H_1 \vee H_2$, được định nghĩa như sau cũng là một HC đặc biệt:

$$H_1 \vee H_2(X) = H_1(X) \cup H_2(X), \forall X \subseteq U.$$

5.8. Xét hai tập hữu hạn phân biệt V_1, V_2 và tương ứng hai HC đặc biệt $H_1 \in Choice(V_1), H_2 \in Choice(V_2)$. Chứng minh *tích trực tiếp* của hai HC đặc biệt H_1 và H_2 , ký hiệu $H_1 \times H_2$, được định nghĩa như sau cũng là một HC đặc biệt:

$$H_1 \times H_2(X) = H_1(X \cap V_1) \cup H_2(X \cap V_2), \forall X \subseteq V_1 \cup V_2.$$

5.9. Cho một tập hữu hạn V và hai HC đặc biệt $H_1, H_2 \in Choice(V)$. Chứng minh lớp các HC đặc biệt không đóng kín đối với phép toán sau:

a) (*Hội*) Ánh xạ hội (của H_1 và H_2) $H_1 \wedge H_2 : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ xác định bởi $H_1 \wedge H_2(X) = H_1(X) \cap H_2(X)$ với mỗi $X \subseteq V$.

b) (*Hợp thành*) Ánh xạ hợp thành (của H_1 và H_2) $H_1 H_2 : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ xác định bởi $H_1 H_2(X) = H_1(H_2(X))$ với mỗi $X \subseteq V$.

5.10. Cho LĐQH $S = (U, F)$. Chứng minh rằng

$$Gen(F) = \bigcup_{a \in U} M(F, a).$$

5.11. Cho LĐQH $S = (U, F)$ và quan hệ $R \in Rel(U)$. Chứng minh R là quan hệ Armstrong của S khi và chỉ khi

$$Gen(F) \subseteq E(R) \subseteq Closed(F).$$

5.12. Xét quan hệ $R \in Rel(U)$, $X \subseteq U$ và $O \subseteq R$. Chứng minh rằng

a) O là X -xác định (tương ứng X -không xác định toàn phần) khi và chỉ khi tập $R \setminus O$ là X -xác định (tương ứng X -không xác định toàn phần).

b) O là X -không xác định trong khi và chỉ khi $R \setminus O$ là X -không xác định ngoài.

5.13. Chứng minh Mệnh đề 5.20 không đúng cho tập thương thu gọn.

5.14. Chứng minh *tính cộng tính phải* không đúng đối với các phụ thuộc cấp k , tức là nếu $X \xrightarrow{k} Y$ và $X \xrightarrow{k} Z$ thì không suy ra được $X \xrightarrow{k'} Y \cup Z$ với $k' \geq k$.

5.15. Chứng minh *tính bắc cầu* không đúng đối với các phụ thuộc cấp k , tức là nếu $X \xrightarrow{k} Y$ và $Y \xrightarrow{k} Z$ thì không suy ra được $X \xrightarrow{k'} Z$ với $k \leq k' < 1$.

TÀI LIỆU THAM KHẢO

- [1] W. W. Armstrong, Dependency structures of data base relationships, *Information Processing*, 1974, 580-583.
- [2] C. Beeri, R. Fagin, J. H. Howard , A complete axiomatization for functional and multivalued dependencies in database Relations, *Proceedings of the 1977 ACM SIGMOD International Conference on Management of Data*, 1977, 47-61.
- [3] E. F. Codd, A Relational model of data for large shared data banks, *CACM*, 1970, 377-387.
- [4] J. Demetrovics, G. Hencsey, L. Libkin, I. Muchnik, On the interaction between closure operations and choice functions with applications to relational databases, *Acta Cybernetica*, 1992, 129-139.
- [5] J. Demetrovics, Vu Duc Thi, Some results about normal forms for functional dependency in the Relational datamodel, *Discrete Applied Mathematics*, 1996, 61-74.
- [6] J. Demetrovics, Vu Duc Thi, Nguyen Long Giang, On finding all reducts of consistent decision tables, *Cybernetics and Infomation Technologies*, 2014, 3-10.
- [7] J. Demetrovics, Vu Duc Thi, Nguyen Long Giang, Tran Huy Duong, On the time complexity of the problem related to reducts of consistent decision tables, *Serdica Journal of Computing*, 2015, 167-176.
- [8] K. Engel, *Sperner theory*, Cambridge University Press, 1997.
- [9] G. Gottlob, L. Libkin, Investigations on Armstrong relations,

- dependency inference and excluded functional dependencies, *Acta Cybernetica*, 1990, 385-402.
- [10] Nguyễn Xuân Huy, *Các phụ thuộc logic trong cơ sở dữ liệu*, NXB Thống kê, 2006.
 - [11] D. Maier, *The theory of relational databases*, Computer Science Press, 1983.
 - [12] H. Mannila, K. J. R  ih  , Design by example: an application of Armstrong relations, *J. Comput. System Sci.*, 1986, 126-141.
 - [13] Vu Duc Nghia, Relationships between closure operations and choice functions equivalent descriptions of a family of functional dependencies, *Acta Cybernetica*, 2004, 485-506.
 - [14] Z. Pawlak., *Rough sets-Theoretical aspects of reasoning about data*, Kluwer Academic Publishers, The Netherlands, 1991.
 - [15] Nguyễn Hoàng Sơn, Nguyễn Việt Hùng, Một số kết quả về khóa trong cơ sở dữ liệu quan hệ, *Tạp chí Tin học và Điều khiển học*, 2002, 285-289.
 - [16] Nguyễn Hoàng Sơn, *Một số vấn đề liên quan đến ràng buộc dữ liệu trong cơ sở dữ liệu quan hệ*, Luận án tiến sĩ toán học, Hà nội, 2006.
 - [17] Nguyen Hoang Son, Investigation on antikeys and minimal keys of relation schemes by hypergraphs, *Annales Univ. Sci. Budapest Sectio Computatoria*, 2006, 79-89.
 - [18] Nguyễn Hoàng Sơn, Một số kết quả về hàm chọn, *Tạp chí Khoa học và Công nghệ, Đại học Khoa học Huế*, 2018, 13-19.
 - [19] Nguyen Hoang Son, Vu Duc Thi, Some the combinatorial characteristics of closure operations, *Algebra and Discrete Mathematics*, 2019, 144-156.
 - [20] Nguyễn Hoàng Sơn, Nguyễn Ngọc Thủy, Nguyễn Hùng Sơn, Nguyễn Long Giang, Some observations on representation of dependency degree k , *Proceedings of the 9th International Conference on Knowledge and Systems Engineering (KSE 2017, Publisher: IEEE)*, 2017, 13-17.

- [21] Vu Duc Thi, Nguyen Hoang Son, Some problems related to keys and the Boyce-Codd normal form, *Acta Cybernetica*, 2004, 473-483.
- [22] Vu Duc Thi, Nguyen Hoang Son, On the dense families in the relational datamodel, *Asean Journal on Science & Technology for Development*, 2005, 241-249.
- [23] Vu Duc Thi, Nguyen Hoang Son, Some results related to dense families of database relations, *Acta Cybernetica*, 2005, 173-182.
- [24] Vu Duc Thi, Nguyen Hoang Son, On Armstrong relations for strong dependencies, *Acta Cybernetica*, 2006, 521-531.
- [25] Vu Duc Thi, Nguyen Hoang Son, On the time complexity of the problem of constructing a relation scheme, *Advances in Natural Sciences*, 2008, 1-8.
- [26] Vu Duc Thi, Nguyen Hoang Son, On dense families of relation schemes and its application, *Tạp chí Tin học và Điều khiển học*, 2009, 125-134.
- [27] Vu Duc Thi, Nguyen Long Giang, A method to construct a decision table from a relation scheme, *Cybernetics and Information Technologies*, 2011, 32-41.
- [28] Vũ Đức Thi, *Cơ sở dữ liệu: kiến thức và thực hành*, NXB Thống kê, 1997.
- [29] Ho Thuan, *Contribution to the theory of Relational databases*, Tanulmanyok, Studies 84, 1986.
- [30] Hồ Thuần, Hồ Cẩm Hà, *Các hệ cơ sở dữ liệu: lý thuyết và thực hành*, NXB Giáo dục, 2005.

DANH MỤC CHỮ VIẾT TẮT, KÝ HIỆU

CSDL	Cơ sở dữ liệu
NSD	Người sử dụng
MHDL	Mô hình dữ liệu
LĐQH	Lược đồ quan hệ
PTH	Phụ thuộc hàm
PTĐT	Phụ thuộc đa trị
TTBĐ	Toán tử bao đóng
HBĐ	Hệ bao đóng
HC	Hàm chọn
$\subseteq, \subset$	Bao hàm, bao hàm thật sự
$\mathcal{P}(V)$	Tập tất cả các tập con của V
$Rel(U)$	Tập các quan hệ R trên tập thuộc tính U
$S = (U, F)$	Lược đồ quan hệ S
$Key(R)$	Tập các khóa tối thiểu của quan hệ R
$Key(S)$	Tập các khóa tối thiểu của lược đồ quan hệ S
$E(R)$	Hệ bằng nhau của quan hệ R
$\max(R)$	Hệ bằng nhau cực đại của quan hệ R
$Tran(\mathcal{S})$	Tập tất cả đại diện của hệ Sperner $\mathcal{S}$
$FD(R)$	Tập các phụ thuộc hàm đúng trên R
$R(X \rightarrow Y)$	Quan hệ R thỏa phụ thuộc hàm $X \rightarrow Y$
$R(\neg X \rightarrow Y)$	Quan hệ R không thỏa phụ thuộc hàm $X \rightarrow Y$
$R(F), R(\neg F)$	Quan hệ R thỏa, không thỏa tập phụ thuộc hàm F
$Sat(F)$	Tập tất cả các quan hệ thỏa tập phụ thuộc hàm F
$\models$	Suy diễn theo quan hệ
$\models$	Suy diễn theo hệ tiên đề
$\overset{A}{F}^*$	Tập các phụ thuộc hàm được suy diễn theo quan hệ từ tập phụ thuộc hàm F
F^+	Bao đóng tập phụ thuộc hàm F
X^+, X_R^+	Bao đóng tập thuộc tính X trên lược đồ quan hệ S , quan hệ R
$Closed(F)$	Tập các tập đóng ứng với $S = (U, F)$
$AntiSperner(\mathcal{S})$	Phản hệ Sperner $\mathcal{S}$
$Antikey(S)$	Tập các phản khóa của lược đồ quan hệ S

U_N	Tập các thuộc tính không cơ bản
U_L, U_R	Tập thuộc tính bên trái, bên phải phụ thuộc hàm
U_I	Tập giao các khóa tối thiểu
$\tilde{S} = (\tilde{U}, \tilde{F})$	Lược đồ quan hệ chuyển dịch
$Closure(V)$	Tập các toán tử bao đóng trên V
$K(a)$	Họ các tập tối thiểu của thuộc tính a
$Gen(F)$	Hệ sinh của tập phụ thuộc hàm F
$IND(X)$	Quan hệ không phân biệt trên X
R/X	Tập thương của quan hệ R theo $IND(X)$
$\underline{X}(O), \overline{X}(O)$	X -xấp xỉ dưới, X -xấp xỉ trên của O
$B_X(O)$	X -miền biên của O
$POS(X, Y)$	Miền khẳng định của X ứng với Y
$\rightarrow, \xrightarrow{k}$	Phụ thuộc hàm, phụ thuộc hàm cấp k
$e(X \rightarrow Y, R)$	Độ đo lỗi của $X \rightarrow Y$ trên quan hệ R
$AFD(R, \epsilon)$	Tập các phụ thuộc hàm xấp xỉ đúng trên quan hệ R ứng với ϵ
$e(X)$	Độ đo lỗi của X
$\widehat{R/X}$	Tập thương thu gọn của R/X
$\rightarrow \rightarrow$	Phụ thuộc đa trị
FM	
$\models$	Suy diễn theo quan hệ (phụ thuộc đa trị)
$\overset{A}{FM}$	
$\models$	Suy diễn theo hệ tiên đề (phụ thuộc đa trị)
$\overset{A}{D} = F \cup M$	Tập phụ thuộc hàm F và phụ thuộc đa trị M
D^+	Bao đóng tập phụ thuộc D
$mdsb(\mathcal{P})$	Cơ sở phân tách tối thiểu của họ $\mathcal{P}$
$Dep(X)$	Cơ sở phụ thuộc của X

CHỈ MỤC

A

ánh xạ chiếu-kết nối, 148

B

bài toán

 khai phá phụ thuộc hàm xấp
 xỉ, 237

 suy diễn phụ thuộc, 114

 thành viên, 114

 xây dựng quan hệ Armstrong,
 118

bảo toàn tập phụ thuộc hàm, 152

bất khả quy, 216

bao đóng, 110, 114, 122, 177

C

cơ sở

 dữ liệu, 9

 phân tách tối thiểu, 183

 phụ thuộc, 184

D

dạng chuẩn

 ba, 156

 bốn, 190

Boyce-Codd, 156

 hai, 155

 một, 154

H

hợp thành, 249, 250

hàm chọn, 211

 đặc biệt, 213

hạn chế, 249

hệ

 bằng nhau, 46

 bằng nhau cực đại, 46

 bao đóng, 209

 cơ sở dữ liệu, 10

 quản trị cơ sở dữ liệu, 10

 sinh, 216

 Sperner, 46

 tiên đề Armstrong, 111

 tiên đề Armstrong-BFH, 178

hội, 249, 250

K

 kết nối

 không mất thông tin, 147

 mất thông tin, 147

khóa, 43, 134
 chính, 44
 dự tuyển, 44
 tối tiểu, 44, 134
 xấp xỉ, 243

L

lược đồ
 cơ sở dữ liệu, 12
 cơ sở dữ liệu quan hệ, 42
 quan hệ, 41
 quan hệ chuyển dịch, 197

M

mịn, 238
mô hình dữ liệu, 15
 quan hệ, 39
 thực thể-mối quan hệ, 25
mối quan hệ isa, 32
mối quan hệ, 31
miền khẳng định, 228

P

phép tách, 147
phép toán
 chọn, 56
 chia, 55
 chiếu, 57
 dán bộ, 55
 gộp nhóm, 61
 giao, 53
 hợp, 53
 kết nối, 58

 kết nối bằng, 59
 kết nối θ , 58
 tích Descartes, 55

phạm trù, 225

 X-cơ sở, 225

phản

 hệ Sperner, 48
 khóa, 143

phụ thuộc

 đa trị, 169
 đầy đủ, 154
 bắc cầu, 155
 bộ phận, 154
 bộ phận cấp k, 229
 cấp k, 229
 hàm, 107
 hàm xấp xỉ, 235
 hoàn toàn, 229
 trực tiếp, 155

phủ, 124

 không dư, 126
 rút gọn, 128
 rút gọn phải, 128
 rút gọn trái, 128
 tối tiểu, 132

Q

quan hệ, 41
 Armstrong, 118
 không phân biệt, 220

S

sơ đồ thực thể-mối quan hệ, 34

suy diễn, 125
theo quan hệ, 110, 170
theo tiên đề, 111, 178

T

tương đương, 115, 124, 237
tích, 246

trực tiếp, 249, 250

tính

đơn điệu, 115
bù, 177
bắc cầu, 111, 178
cộng tính đầy đủ, 112
cộng tính phải, 112, 181
gia tăng, 111, 177
giả bắc cầu, 112, 181
lũy đẳng, 115
lặp lại, 178
liên kết, 178
mở rộng trái và thu hẹp phải,
113
phản xạ, 111, 115, 183
phản xạ chặt, 111
thu hẹp phải, 112, 181

tập

đóng, 115, 249
dự tuyển về phải, 247
tối tiểu của thuộc tính, 214

tập thương, 238

thể hiện

lược đồ quan hệ, 118

thỏa

phụ thuộc đa trị, 169
phụ thuộc hàm, 107
phụ thuộc hàm xấp xỉ, 235

thực thể, 28

thuật toán

kiểm tra tính bảo toàn, 152
kiểm tra tính không mất thông
tin, 148

kiểm tra tính không mất thông
tin ứng với PTH và PTĐT,
189

suy diễn phụ thuộc, 121

tách về dạng chuẩn ba, 165

tách về dạng chuẩn bốn, 191

tách về dạng chuẩn Boyce-Codd,
163

tìm bao đóng thuộc tính, 119,
123

tìm cơ sở phụ thuộc, 185

tìm họ các tập tối tiểu, 216

tìm một khóa tối tiểu, 46, 139

tìm một phủ không dư, 127

tìm một phủ rút gọn, 130

tìm một phủ rút gọn phải,
130

tìm một phủ rút gọn trái, 130

tìm một phủ tối tiểu, 132

tìm tất cả đại diện, 49

tìm tất cả khóa tối tiểu, 142

tìm tất cả khóa tối tiểu của
một quan hệ, 50

thành viên, 120, 187

xây dựng quan hệ Armstrong,
217
thuộc tính, 28
 cơ bản, 135
 không cơ bản, 135
 siêu cơ bản, 194
toán tử bao đóng, 205
tuyển, 249, 250

X

X-không xác định, 225
 ngoài, 226
 toàn phần, 226
 trong, 226
X-miền
 biên, 222
 ngoài, 222
X-rõ, 225
X-thô, 225
X-xác định, 225
 thô, 226
X-xấp xỉ
 dưới, 221
 trên, 221

ISBN 978-604-912-908-7



Gia: 60.000đ