Chuẩn hóa lược đồ quan hệ

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

Các dạng chuẩn trên khóa chính

Sau khi đã nghiên cứu các phụ thuộc hàm và một số tính chất của chúng, bây giờ chúng ta sẽ sử dụng chúng như thông tin về ngữ nghĩa của các lược đồ quan hệ. Ta giả sử rằng mỗi một quan hệ được cho trước một tập các phụ thuộc hàm và mỗi quan hệ có một khoá chính. Trong phần này chúng ta sẽ nghiên cứu các dạng chuẩn và quá trình chuẩn hoá các lược đồ quan hệ.

Nhập môn về chuẩn hoá

Quá trình chuẩn hoá (do Codd đề nghị 1972) lấy một lược đồ quan hệ và thực hiện một loạt các kiểm tra để xác nhận nó có thoả mãn một dạng chuẩn nào đó hay không. Quá trình này được thực hiện theo phương pháp trên xuống bằng việc đánh giá mỗi quan hệ với tiêu chuẩn của các dạng chuẩn và tách các quan hệ nếu cần. Quá trình này có thể xem như là việc thiết kế quan hệ bằng phân tích. Lúc đầu, Codd đề nghị ba dạng chuẩn gọi là dạng chuẩn 1, dạng chuẩn 2 và dạng chuẩn 3. Một định nghĩa mạnh hơn cho dạng chuẩn 3 gọi là dạng chuẩn Boyce-Codd do Boyce và Codd đề nghị muộn hơn. Tất cả các dạng chuẩn này dựa trên các phụ thuộc hàm giữa các thuộc tính của một quan hệ. Sau đó, dạng chuẩn 4 (4NF) và dạng chuẩn 5 (5NF) được đề nghị dựa trên các phụ thuộc hàm đa trị và các phụ thuộc hàm nối.

Chuẩn hoá dữ liệu có thể được xem như một quá trình phân tích các lược đồ quan hệ cho trước dựa trên các phụ thuộc hàm và các khoá chính của chúng để đạt đến các tính chất mong muốn:

(1) Cực tiểu sự dư thừa và

(2) Cực tiểu các phép cập nhật bất thường.

Các lược đồ quan hệ không thoả mãn các kiểm tra dạng chuẩn sẽ được tách ra thành các lược đồ quan hệ nhỏ hơn thoả mãn các kiểm tra và có các tính chất mong muốn. Như vậy, thủ tục chuẩn hoá cung cấp cho những người thiết kế cơ sở dữ liệu:

- Một cơ cấu hình thức để phân tích các lược đồ quan hệ dựa trên các khoá của nó và các phụ thuộc hàm giữa các thuộc tính của nó.
- Một loạt các kiểm tra dạng chuẩn có thể thực hiện trên các lược đồ quan hệ riêng rẽ sao cho cơ sở dữ liệu quan hệ có thể được chuẩn hoá đến một mức cần thiết.

Dạng chuẩn của một quan hệ liên quan đến điều kiện dạng chuẩn cao nhất mà nó thoả mãn. Các dạng chuẩn khi được xem xét độc lập với các sự kiện khác không đảm bảo một thiết kế cơ sở dữ liệu tốt. Nói chung, việc xác minh riêng biệt từng lược đồ quan hệ ở dạng chuẩn này dạng chuẩn nọ là chưa đủ. Tốt hơn là quá trình chuẩn hoá thông qua phép tách phải khẳng định một vài tính chất hỗ trợ mà tất cả các lược đồ quan hệ phải có. Chúng gồm hai tính chất sau:

- Tính chất nối không mất mát (hoặc nối không phụ thêm), nó đảm bảo rằng vấn đề tạo ra các bộ giả không xuất hiện đối với các lược đồ quan hệ được tạo ra sau khi tách.
- Tính chất bảo toàn sự phụ thuộc, nó đảm bảo rằng từng phụ thuộc hàm sẽ được biểu hiện trong các quan hệ riêng rẽ nhận được sau khi tách.

Tính chất nối không mất mát là rất quan trọng, phải đạt được bằng mọi giá, còn tính chất bảo toàn phụ thuộc thì cũng rất mong muốn nhưng đôi khi có thể hy sinh.

Trước khi định nghĩa các dạng chuẩn chúng ta xem lại định nghĩa các khóa của một quan hệ. Một *siêu khóa* (superkey) của một lược đồ quan hệ $R = \{A_1, A_2, .., A_n\}$ là một tập con S các thuộc tính của R, $S \subseteq R$, có tính chất là không có hai bộ t_1, t_2 trong một trạng thái quan hệ hợp pháp r nào của R mà $t_1[S] = t_2[S]$. Một *khóa* K là một siêu khóa có tính chất là nếu bỏ đi bất kỳ thuộc tính nào ra khỏi K thì K không còn là siêu khóa nữa. Điều đó có nghĩa là khóa là một siêu khóa tối thiểu. Nếu một lược đồ quan hệ có nhiều hơn một khóa thì các khóa đó được gọi là các *khóa dự tuyển*. Một trong những khóa dự tuyển sẽ được chỉ định làm *khóa chính* và các khóa còn lại được gọi là các *khóa phụ* (secondary key). Một lược đồ quan hệ phải có một khóa chính. Một thuộc tính của một lược đồ quan hệ R được gọi là một *thuộc tính khóa* của R nếu nó là một thành phần của khóa chính của R. Một thuộc tính được gọi là thuộc tính *không khóa* nếu nó không phải là một thuộc tính khóa (*thuật toán tìm tất cả các khóa của lược đồ được trình bày trong phần*).

Dạng chuẩn 1

Một quan hệ được gọi là ở *dạng chuẩn 1* (1NF) nếu miền giá trị của một thuộc tính chỉ chứa các giá trị nguyên tử (đơn, không phân chia được) và giá trị của mỗi thuộc tính trong một bộ phải là một giá trị đơn lấy từ miền giá trị của thuộc tính đó. Như vậy, 1NF không cho phép quan hệ có các thuộc tính đa trị hoặc có các nhóm thuộc tính lặp (quan hệ trong quan hệ). Nó chỉ cho phép các giá trị của các thuộc tính là nguyên tử.

Ví dụ, xét các quan hệ ĐƠNVỊ và NHÂNVIÊN_DỰÁN như hình vẽ dưới đây. Các quan hệ này không thỏa mãn điều kiện 1NF. Quan hệ ĐƠNVỊ chứa thuộc tính đa trị Địađiểm, quan hệ NHÂNVIÊN_DỰÁN chứa nhóm các thuộc tính lặp (Tênnhânviên, Sốgiờ).

ĐƠNVỊ	MãsốĐV	TênĐV	Mã sốNQL	Địađiểm
	5	Nghiêncứu	NV002	Namđịnh, Hànội,Bắcninh
	4	Hànhchính	NV014	Hànôi
	1	Lãnhđạo	NV061	Hànội

Bảng ĐƠNVỊ

NHÂNVIÊN_DỰÁN	MãsốDA	TênDA	Tênnhânviên	Sốgiờ
	1	DA01	Vân, Nam	15, 20
	2	DA02	Nam, Thanh Bằng	10, 12, 28
	3	DA03	Thanh	20

Bảng NHÂNVIÊN_DỰÁN

Để đạt đến dạng chuẩn 1 đối với các quan hệ ở trên chúng ta dùng phương pháp sau:

Loại bỏ các thuộc tính vi phạm dạng chuẩn 1 và đặt chúng vào một bảng riêng cùng với khoá chính của quan hệ ban đầu. Khoá chính của bảng này là một tổ hợp của khoá chính của quan hệ ban đầu và thuộc tính đa trị hoặc khoá bộ phận của nhóm lặp.

Các thuộc tính còn lại lập thành một quan hệ với khóa chính là khóa chính ban đầu.

Áp dụng : Lược đồ quan hệ ĐƠNVỊ ở trên sẽ được tách thành hai:

ĐƠNVỊ (Mã sốĐV, TênĐV, Mã sốNQL)

ĐƠNVỊ_ĐỊAĐIỂM (MãsốĐV, Điạđiểm)

Lược đồ quan hệ NHÂNVIÊN_DỰÁN cũng được tách thành hai:

DỰÁN (MãsốDA, TênDA)

NHÂNVIÊN_DỰÁN(MãsốDA, Tênnhânviên, Sốgiờ)

Dạng chuẩn 2

Dạng chuẩn 2 (2NF) dựa trên khái niệm phụ thuộc hàm đầy đủ. Một phụ thuộc hàm X → Y là một phụ thuộc hàm đầy đủ nếu loại bỏ bất kỳ thuộc tính A nào ra khỏi X thì phụ thuộc hàm không còn đúng nữa. Điều đó có nghĩa là, với thuộc tính A bất kỳ, A ∈ X, (X – {A}) không xác định Y. Một phụ thuộc hàm X → Y là phụ thuộc bộ phận nếu có thể bỏ một thuộc tính A∈ X, ra khỏi X phụ thuộc hàm vẫn đúng, điều đó có nghĩa là với A∈ X, (X – {A}) → Y.

Ví dụ, xét lược đồ quan hệ

NHÂNVIÊN_DỰÁN(MãsốNV, MãsốDA, Sốgiờ, HọtênNV, TênDA, ĐịađiểmDA)

MãsốNV, MãsốDA → Sốgiờ là phụ thuộc hàm đầy đủ

MãsốNV, MãsốDA → HọtênNV là phụ thuộc hàm bộ phận, bởi vì có phụ thuộc hàm

MãsốNV →HọtênNV

Việc kiểm tra đối với 2NF bao gồm việc kiểm tra đối với các phụ thuộc hàm có các thuộc tính ở vế trái của nó là một bộ phận của khoá chính. Nếu khoá chính chứa một thuộc tính đơn thì không cần phải kiểm tra. Một lược đồ quan hệ R là ở *dạng chuẩn 2* nếu nó thỏa mãn dạng chuẩn 1 và mỗi thuộc tính không khoá A trong R là phụ thuộc hàm đầy đủ vào khoá chính của R.

Nếu một quan hệ không thoả mãn điều kiện 2NF ta có thể chuẩn hoá nó để có các quan hệ 2NF như sau: Loại bỏ các thuộc tính không khoá phụ thuộc vào một bộ phận khoá chính và tách thành ra một bảng riêng, khoá chính của bảng là bộ phận khoá mà chúng phụ thuộc vào. Các thuộc tính còn lại lập thành một quan hệ, khóa chính của nó là khóa chính ban đầu.

Ví dụ, xét lược đồ quan hệ:

NHÂNVIÊN_DỰÁN(MãsốNV, MãsốDA, Sốgiờ, HọtênNV, TênDA, ĐịađiểmDA)

với các phụ thuộc hàm:

MãsốNV, MãsốDA → Sốgiờ

MãsốNV →HọtênNV

MãsốDA→TênDA, ĐịađiểmDA

Ta thấy ở đây có những thuộc tính không khoá phụ thuộc vào một bộ phận của khoá chính, như vậy nó không thoả mãn điều kiên 2NF.

Áp dụng phương pháp chuẩn hoá trên, lược đồ được tách thành các lược đồ như sau:

N_D1(MãsốDA, TênDA, ĐịađiểmDA)

N_D2(MãsốNV , HọtênNV)

N_D3(MãsốNV, MãsốDA, Sốgiờ)

Dạng chuẩn 3

Dạng chuẩn 3 (3NF) dựa trên khái niệm phụ thuộc bắc cầu. Một phụ thuộc hàm X → Y trong một lược đồ quan hệ R là một phụ thuộc hàm bắc cầu nếu có một tập hợp thuộc tính Z không phải là một khoá dự tuyển cũng không phải là một tập con của một khoá nào và cả hai X → Z và Z →Y đều đúng. Theo định nghĩa nguyên thuỷ của Codd, một lược đồ quan hệ R là ở 3NF nếu nó thoả mãn 2NF và không có thuộc tính không khoá nào của R là phụ thuộc bắc cầu vào khoá chính.

Nếu một lược đồ quan hệ không thoả mãn điều kiện 3NF, ta có thể chuẩn hoá nó để có được các lược đồ 3NF như sau: Loại bỏ các thuộc tính phụ thuộc bắc cầu ra khỏi quan hệ và tách chúng thành một quan hệ riêng có khoá chính là thuộc tính bắc cầu. Các thuộc tính còn lại lập thành một quan hệ có khóa chính là quan hệ ban đầu.

Ví dụ: Xét lược đồ quan hệ

NHÂNVIÊN_ĐƠNVỊ(HọtênNV, MãsốNV, Ngàysinh, Địachỉ, MãsốĐV, TênĐV, MãsốNQL)

Với các phụ thuộc hàm:

MãsốNV→ HọtênNV, Ngày sinh, Địachỉ, MãsốĐV, TênĐV, MãsốNQL

MãsốDV→ TênĐV, Mã sốNQL

Các thuộc tính TênĐV, MãsốNQL phụ thuộc bắc cầu vào khoá chính, lược đồ quan hệ không thoả mãn điều kiện 3NF.

Áp dụng phương pháp chuẩn hoá ở trên, lược đồ được tách ra như sau:

NV_DV1(HọtênNV, MãsốNV, Ngàysinh, Địachỉ, MãsốĐV)

NV_DV2(MãsốĐV, TênĐV, MãsốNQL)

Dạng chuẩn Boyce-Codd

Một lược đồ quan hệ R được gọi là ở dạng chuẩn Boyce-Codd (BCNF) nếu nó là ở dạng chuẩn 3NF và không có các thuộc tính khóa phụ thuộc hàm và thuộc tính không khóa.

Ví dụ: Lược đồ

R ($\underline{A1}$,$\underline{A2}$,A_3,A_4,A_5)

Với các phụ thuộc hàm:

A1,$A_2 \rightarrow A_3$,A_4,A_5

$A_4 \rightarrow A_2$

Quan hệ này vi phạm dạng chuẩn BCNF bởi vì có thuộc tính khóa (A_2) phụ thuộc hàm vào thuộc tính không khóa (A_4).

Nếu một lược đồ quan hệ không thoả mãn điều kiện BCNF, ta có thể chuẩn hoá nó để có được các lược đồ BCNF như: Loại bỏ các thuộc tính khóa phụ thuộc hàm vào thuộc tính không khóa ra khỏi quan hệ và tách chúng thành một quan hệ riêng có khoá chính là thuộc tính không khóa gây ra phụ thuộc.

Áp dụng phương pháp chuẩn hóa ở trên, lược đồ được tách ra như sau:

R1($\underline{A4}$, A_2)

R2($\underline{A1}$, $\underline{A4}$, A_3, A_5)

Ví dụ áp dụng:

Cho lược đồ quan hệ R = {$\underline{A}$,$\underline{B}$,C,D,E,F,G,H,I,J} có khóa chính là A,B

Với tập các phụ thuộc hàm :

A,B $\rightarrow$ C,D,E,F,G,H,I,J

A$\rightarrow$ E,F,G,H,I,J

F → I, J

D →B

Do có có phụ thuộc hàm A→ E,F,G,H,I,J mà A là một bộ phận của khóa chính nên quan hệ R là vi phạm 2NF. Ta tách R thành R_1(A,E,F,G,H,I,J) và R_2(A,B,C,D). Trong R_1, do có phụ thuộc hàm F→ I, J, nên ta có I,J phụ thuộc bắc cầu vào khóa chính, R_1 là quan hệ vi phạm 3NF. Trong R_2 ta có phụ thuộc hàm D → B trong đó B là một thuộc tính khóa, R_2 vi phạm BCNF. Tách R_1 và R_2 ta có:

R_{11}(F,I,J) , R_{12}(A,E,F,G,H), R_{21}(D,B), R_{22}(A,D,C)

Định nghĩa tổng quát các dạng chuẩn và các thuật toán liên quan đến quá trình chuẩn hóa CSDL

Như chúng ta đã thảo luận trong bài 7, có hai cách chính để thiết kế cơ sở dữ liệu quan hệ. Cách thứ nhất là *thiết kế trên-xuống* (top-down design). Đây là cách hay được sử dụng nhất trong thiết kế ứng dụng cơ sở dữ liệu thương mại. Nó bao gồm việc thiết kế một lược đồ quan niệm trong một mô hình dữ liệu bậc cao, chẳng hạn như mô hình ER, sau đó ánh xạ lược đồ quan niệm vào một tập quan hệ sử dụng các thủ tục ánh xạ như đã nói đến trong bài 5. Sau đó, mỗi một quan hệ được phân tích dựa trên các phụ thuộc hàm và các khóa chính được chỉ định bằng cách áp dụng các thủ tục chuẩn hóa như đã nói đến trong phần 8.1 để loại bỏ các phụ thuộc hàm bộ phận và các phụ thuộc hàm bắc cầu. Việc phân tích các phụ thuộc không mong muốn cũng có thể được thực hiện trong quá trình thiết kế quan niệm bằng cách phân tích các phụ thuộc hàm giữa các thuộc tính bên trong các kiểu thực thể và các kiểu liên kết để ngăn ngừa sự cần thiết có sự chuẩn hóa phụ thêm sau khi việc ánh xạ được thực hiện.

Cách thứ hai là *thiết kế dưới-lên* (bottom-up design), một kỹ thuật tiếp cận và nhìn nhận việc thiết kế lược đồ cơ sở dữ liệu quan hệ một cách chặt chẽ trên cơ sở các phụ thuộc hàm được chỉ ra trên các thuộc tính của cơ sở dữ liệu. Sau khi người thiết kế chỉ ra các phụ thuộc, người ta áp dụng một thuật toán chuẩn hóa để tổng hợp các lược đồ quan hệ. Mỗi một lược đồ quan hệ riêng rẽ ở dạng chuẩn 3NF hoặc BCNF hoặc ở dạng chuẩn cao hơn.

Trong phần này chúng ta chủ yếu trình bày cách tiếp cận thứ hai. Trước tiên chúng ta sẽ định nghĩa lại các dạng chuẩn một cách tổng quát, sau đó trình bày các thuật toán chuẩn hóa và các kiểu phụ thuộc khác. Chúng ta cũng sẽ trình bày chi tiết hơn về hai tính chất cần có là nối không phụ thêm (mất mát) và bảo toàn phụ thuộc. Các thuật toán chuẩn hóa thường bắt đầu bằng việc tổng hợp một lược đồ quan hệ rất lớn, gọi là *quan hệ phổ quát* (universal relation), chứa tất cả các thuộc tính của cơ sở dữ liệu. Sau đó chúng ta thực hiện lặp đi lặp lại việc tách (decomposition) dựa trên các phụ thuộc hàm và các phụ

thuộc khác do người thiết kế cơ sở dữ liệu chỉ ra cho đến khi không còn tách được nữa hoặc không muốn tách nữa.

Định nghĩa tổng quát các dạng chuẩn

Nói chung, chúng ta muốn thiết kế các lược đồ của chúng ta sao cho chúng không còn các phụ thuộc bộ phận và các phụ thuộc bắc cầu bởi vì các kiểu phụ thuộc này gây ra các sửa đổi bất thường. Các bước chuẩn hóa thành 3NF, BCNF đã được trình bày trong phần trước loại bỏ các phụ thuộc bộ phận và bắc cầu dựa trên khóa chính. Các định nghĩa này không tính đến các khóa dự tuyển của quan hệ. Trong phần này chúng ta sẽ đưa ra các định nghĩa về các dạng chuẩn tổng quát hơn, có tính đến tất cả các khóa dự tuyển. Cụ thể, *thuộc tính khóa* được định nghĩa lại *là một bộ phận của một khóa dự tuyển*. Các phụ thuộc hàm bộ phận, đầy đủ, bắc cầu bây giờ sẽ được định nghĩa đối với tất cả các khóa dự tuyển của quan hệ.

Định nghĩa dạng chuẩn 1: Một lược đồ quan hệ R là ở dạng chuẩn 1 (1NF) nếu miền giá trị của các thuộc tính của nó chỉ chứa các *giá trị nguyên tử* (đơn, không phân chia được) và giá trị của một thuộc tính bất kỳ trong một bộ giá trị phải là một giá trị đơn thuộc miền giá trị của thuộc tính đó.

Định nghĩa dạng chuẩn 2: Một lược đồ quan hệ R là ở dạng chuẩn 2 (2NF) nếu mỗi thuộc tính không khóa A trong R không phụ thuộc bộ phận vào một khóa bất kỳ của R.

Ví dụ: Xét lược đồ quan hệ

R={A,B,C,D,E,F}

Với các phụ thuộc hàm A → B,C,D,E,F; B,C → A,D,E,F; B → F; D →E.

Lược đồ trên có hai khóa dự tuyển là A và {B,C}. Ta chọn A làm khóa chính. Do có phụ thuộc hàm B → F nên F phụ thuộc bộ phận vào khóa {B,C}, lược đồ vi phạm chuẩn 2NF (chú ý rằng, trong định nghĩa dạng chuẩn dựa trên khóa chính, lược đồ này không vi phạm 2NF).

Định nghĩa dạng chuẩn 3: Một lược đồ quan hệ R là ở dạng chuẩn 3 (3NF) nếu khi một phụ thuộc hàm X → A thỏa mãn trong R, thì:

1) Hoặc X là một siêu khóa của R.

2) Hoặc A là một thuộc tính khóa của R.

Ví dụ: Xét lược đồ quan hệ R ở ví dụ trên. Giả sử nó được tách thành hai lược đồ:

R1 = {A,B,C,D,E}

R2 = {B, F}.

Do có phụ thuộc hàm D → E trong đó D không phải thuộc tính khóa, E cũng không phải là thuộc tính khóa, nên R1 vi phạm chuẩn 3NF

Định nghĩa dạng chuẩn Boyce- Codd: Một lược đồ quan hệ là ở dạng chuẩn Boyce-Codd (BCNF) nếu khi một phụ thuộc hàm X → A thỏa mãn trong R thì X là một siêu khóa của R.

Ví dụ: Xét lược đồ R = {A, B, C, D} có A là khóa chính và {B,C} là khóa dự tuyển. Nếu có tồn tại một phụ thuộc hàm D → B thì lược đồ này vi phạm BCNF vì B là một thuộc tính khóa (chú ý rằng trong trường hợp định nghĩa dạng chuẩn dựa trên khóa chính, lược đồ này không vi phạm BCNF).

Các thuật toán liên quan đến quá trình chuẩn hóa CSDL

Tách quan hệ và tính không đầy đủ của các dạng chuẩn

Tách quan hệ: Các thuật toán thiết kế cơ sở dữ liệu quan hệ được trình bày trong phần này bắt đầu từ một lược đồ quan hệ vũ trụ đơn R = {A_1, A_2, …, A_n} chứa tất cả các thuộc tính của cơ sở dữ liệu. Với giả thiết quan hệ vũ trụ, tên của mỗi thuộc tính là duy nhất. Tập hợp F các phụ thuộc hàm thỏa mãn trên các thuộc tính của R do những người thiết kế cơ sở dữ liệu chỉ ra sẽ được các thuật toán sử dụng. Sử dụng các phụ thuộc hàm, các thuật toán sẽ tách lược đồ quan hệ vũ trụ R thành một tập hợp các lược đồ quan hệ D = {R_1, R_2, …, R_m}, tập hợp đó sẽ là lược đồ cơ sở dữ liệu quan hệ. D được gọi là một phép tách (decomposition) của R. Chúng ta phải đảm bảo rằng mỗi thuộc tính trong R sẽ xuất hiện trong ít nhất là một lược đồ quan hệ R_i trong phép tách để nó khỏi bị "mất ". Một cách hình thức, ta có *điều kiện bảo toàn thuộc tính sau đây*: **?**R_i = R

Tính không đầy đủ của các dạng chuẩn: Mục đích của chúng ta là mỗi quan hệ riêng rẽ R_i trong phép tách D là ở dạng chuẩn BCNF hoặc 3NF. Tuy nhiên, điều đó không đủ để đảm bảo một thiết kế cơ sở dữ liệu tốt. Bên cạnh việc xem xét từng quan hệ riêng rẽ, chúng ta cần xem xét toàn bộ phép tách. Ví dụ, xét hai quan hệ:

NV_ĐĐ(Tên, ĐiađiểmDA)

NV_DA1(Mã sốNV, Mã sốDA, Sốgiờ, TênDA, ĐịađiểmDA)

Ở phần 8.1.4, ta thấy rằng dù quan hệ NV_ĐĐ là một quan hệ ở dạng BCNF nhưng khi chúng ta đem nối tự nhiên với quan hệ NV_DA1 thì chúng ta nhận được một quan hệ có chứa các bộ giả. Điều đó xảy ra là do ngữ nghĩa không rõ ràng của quan hệ NV_ĐĐ. Đó là một lược đồ quan hệ được thiết kế tồi. Chúng ta cần phải có tiêu chuẩn khác để cùng

với các điều kiện 3NF và BCNF ngăn ngừa các thiết kế tồi như vậy. Trong các phần tiếp theo chúng ta sẽ nối đến các điều kiện phụ thêm phải thỏa mãn trên phép tách D.

Phép tách và sự bảo toàn phụ thuộc

Việc mỗi phụ thuộc hàm X → Y trong F hoặc được xuất hiện trực tiếp trong một trong các lược đồ quan hệ R_i trong phép tách D hoặc có thể được suy diễn từ các phụ thuộc hàm có trong R_i là rất có lợi. Ta gọi đó là *điều kiện bảo toàn phụ thuộc*. Chúng ta muốn bảo toàn phụ thuộc bởi vì mỗi phụ thuộc trong F biểu thị một ràng buộc trong cơ sở dữ liệu. Nếu như một trong các phụ thuộc không được thể hiện trong một quan hệ riêng rẽ R_i nào đó của phép tách, chúng ta không thể ép buộc ràng buộc này đối với quan hệ riêng rẽ, thay vào đó, chúng ta nối hai hoặc nhiều quan hệ trong phép tách và sau đó kiểm tra rằng phụ thuộc hàm thỏa mãn trong kết quả của phép nối. Rõ ràng đó là một thủ tục không hiệu quả và không thực tiễn.

Việc các phụ thuộc chính xác được chỉ ra ở trong F xuất hiện trong các quan hệ riêng rẽ của phép tách D là không cần thiết. Chỉ cần hợp của các phụ thuộc thỏa mãn trên các quan hệ riêng rẽ trong D là tương đương với F là đủ. Bây giờ chúng ta định nghĩa các khái niệm này một cách hình thức.

Cho trước một tập hợp các phụ thuộc F trên R, *phép chiếu của F trên Ri*, ký hiệu là $\pi_{Ri}(F)$ trong đó R_i là một tập con của R, là một tập hợp các phụ thuộc hàm X→Y trong F^+ sao cho các thuộc tính trong X ? Y đều được chứa trong R_i. Như vậy, phép chiếu của F trên mỗi lược đồ quan hệ R_i trong phép tách D là tập hợp các phụ thuộc hàm trong F^+, bao đóng của F, sao cho các thuộc tính ở vế trái và vế phải của chúng đều ở trong R_i. Ta nói rằng phép tách D = {R1, R2, …, Rm} của R bảo toàn phụ thuộc đối với F nếu hợp của các phép chiếu của F trên mỗi R_i trong D là tương đương với F. Điều đó có nghĩa là:

$$((\pi_{R1}(F)) ? (\pi_{R2}(F)) ? \ldots ? (\pi_{Rm}(F)))^+ = F^+$$

Nếu một phép tách là không bảo toàn phụ thuộc, một vài phụ thuộc sẽ bị mất trong phép tách. Để kiểm tra xem một phụ thuộc hàm X→ B, trong đó X là tập thuộc tính thuộc về R_i, B là một thuộc tính thuộc R_i có thỏa mãn trong R_i hay không ta làm như sau: Trước hết tính X^+ , sau đó với mỗi thuộc tính B sao cho

1. B là một thuộc tính của R_i

2. B là ở trong X^+

3. B không ở trong X

Khi đó phụ thuộc hàm X → B thỏa mãn trong R_i.

Một ví dụ về phép tách không bảo toàn phụ thuộc. Xét lược đồ quan hệ:

R = { A,B,C,D} với các phụ thuộc hàm:

A → BCD; BC → DA; D →B

Lược đồ này có hai khóa dự tuyển là A và BC. Lược đồ này vi phạm BCNF. Nó được tách thành:

R1 = {D,B}, lược đồ này chứa phụ thuộc hàm D → B

R2 = {A,C,D}, lược đồ này chứa phụ thuộc hàm A → CD

Rõ ràng sau khi tách, phụ thuộc hàm BC → DA bị mất.

Định lý 8.1: Luôn luôn tìm được một phép tách bảo toàn phụ thuộc D đối với F sao cho mỗi quan hệ R_i trong D là ở 3NF. Phép tách D đươc thực hiện theo thuật toán sau đây:

Thuật toán 8.1: Tạo một phép tách bảo toàn phụ thuộc D = $\{R_1,R_2, \ldots,R_m\}$ của một quan hệ vũ trụ R dựa trên một tập phụ thuộc hàm F sao cho mỗi R_i trong D là ở 3NF. Thuật toán này chỉ đảm bảo tính chất bảo toàn phụ thuộc, không đảm bảo tính chất nối không mất mát.

Input: Một quan hệ vũ trụ R và một tập phụ thuộc hàm F trên các thuộc tính của R.

1. Tìm phủ tối thiểu G của F.
2. Với mỗi vế trái X của một phụ thuộc hàm xuất hiện trong G, hãy tạo một lược đồ trong D với các thuộc tính $\{X ? \{A_1\} ? \{A_2\} ? \ldots ? \{A_k\}\}$ trong đó $X \rightarrow A_1$, $X \rightarrow A_2, \ldots, X \rightarrow A_k$ chỉ là các phụ thuộc hàm trong G với X là vế trái (X là khóa của quan hệ này).
3. Đặt các thuộc tính còn lại (những thuộc tính chưa được đặt vào quan hệ nào) vào một quan hệ đơn để đảm bảo tính chất bảo toàn thuộc tính.

Ví dụ áp dụng:

Xét lược đồ: R = { A,B,C,D} , với các phụ thuộc hàm:

F = {A → BCD; BC → DA; D →B}

Lược đồ này có hai khóa dự tuyển là A và BC.

Ta thực hiện thuật toán như sau: Trước tiên ta tìm G là phủ tối thiểu của F. Theo thuật toán tìm phủ tối thiểu, đầu tiên ta làm cho các vế phải trong G chỉ chứa một thuộc tính, ta có:

G = {A → B; A → C; A→ D; BC → D; BC → A; D → B}

Sau đó ta bỏ đi các phụ thuộc hàm thừa (là các phụ thuộc hàm có thể suy diễn được từ các phụ thuộc hàm khác). Ta thấy A →B là thừa vì có A →D, D →B. Vậy G còn lại là:

G = {A → C; A→ D; BC → D; BC → A; D → B}. Lược đồ R sẽ được tách thành:

R_1(A,C,D); R2(B,C,D,A); R3(D,B) với các khóa chính được gạch dưới.

Rõ ràng rằng tất cả các phụ thuộc hàm trong G đều được thuật toán bảo toàn bởi vì mỗi phụ thuộc xuất hiện trong một trong các quan hệ của phép tách D. Bởi vì G tương đương với F, tất cả các phụ thuộc của F cũng được bảo toàn hoặc trực tiếp bằng thuật toán hoặc được suy diễn từ những phụ thuộc hàm trong các quan hệ kết quả, như vậy tính chất bảo toàn phụ thuộc được đảm bảo.

Phép tách và kết nối không mất mát

Phép tách D phải có một tính chất nữa là nối không mất mát (hoặc tính chất nối không phụ thêm), nó đảm bảo rằng không có các bộ giả được tạo ra khi áp dụng một phép nối tự nhiên vào các quan hệ trong phép tách. Chúng ta đã đưa ra ví dụ về phép tách không có tính chất nối không mất thông tin ở phần I.4 chương này. Trong phép tách đó, khi ta thực hiện phép nối tự nhiên trên các quan hệ của phép tách, rất nhiều các bộ giả đã sinh ra.

Một cách hình thức, ta nói rằng một phép tách D = { R_1, R_2,…,R_m} của R có tính chất nối không mất mát (không phụ thêm) đối với một tập hợp phụ thuộc hàm F trên R nếu với mỗi trạng thái quan hệ r của R thỏa mãn F thì

* ($\pi_{R1}(r)$, $\pi_{R1}(r)$ …, $\pi_{R1}(r)$) = r

trong đó * là phép nối tự nhiên của các quan hệ trong D.

Nếu một phép tách không có tính chất nối không mất mát thông tin thì chúng ta có thể nhận được các bộ phụ thêm (các bộ giả) sau khi áp dụng các phép chiếu và nối tự nhiên. Nghĩa của từ mất mát ở đây là mất mát thông tin chưa không phải mất các bộ giá trị. Vì vậy, với tính chất này ta nên gọi chính xác hơn là tính chất nối không phụ thêm.

Chúng ta có thuật toán để kiểm tra một phép tách có tính chất nối không mất mát thông tin hay không như sau:

Thuật toán 8.2: Kiểm tra tính chất nối không mất mát

Input: Một quan hệ vũ trụ $R(A_1,A_2,\ldots A_n)$, một phép tách $D = \{R_1, R_2, \ldots, R_m\}$ của R và một tập F các phụ thuộc hàm.

1. Tạo một ma trận S có m hàng, n cột. Mỗi cột của ma trận ứng với một thuộc tính, mỗi hàng ứng với mỗi quan hệ R_i
2. Đặt $S(i,j) = 1$ nếu thuộc tính A_j thuộc về quan hệ R_i và bằng 0 trong trường hợp ngược lại.
3. Lặp lại vòng lặp sau đây cho đến khi nào việc thực hiện vòng lặp không làm thay đổi S: Với mỗi phụ thuộc hàm $X \rightarrow Y$ trong F, xác định các hàng trong S có các ký hiệu 1 như nhau trong các cột ứng với các thuộc tính trong X. Nếu có một hàng trong số đó chứa 1 trong các cột ứng với thuộc tính Y thì hãy làm cho các làm cho các cột tương ứng của các hàng khác cũng chứa 1.
4. Nếu có một hàng chứa toàn ký hiệu "1" thì phép tách có tính chất nối không mất mát, ngược lại, phép tách không có tính chất đó.

Cho trước một quan hệ R được tách thành một số quan hệ $R_1, R_2, ..,R_m$. Thuật toán 8.2 bắt đầu bằng việc tạo ra một trạng thái quan hệ r trong ma trận S. Hàng i trong S biểu diễn một bộ t_i (tương ứng với quan hệ R_i). Hàng này có các ký hiệu "1" trong các cột tương ứng với các thuộc tính của R_i và các ký hiệu "0" trong các cột còn lại. Sau đó thuật toán biến đổi các hàng của ma trận này (trong vòng lặp của bước 3) sao cho chúng biểu diễn các bộ thỏa mãn tất cả các phụ thuộc hàm trong F. Ở cuối vòng lặp áp dụng các phụ thuộc hàm, hai hàng bất kỳ trong S – chúng biểu diễn hai bộ trong r – có các giá trị giống nhau đối với các thuộc tính của X ở vế trái của phụ thuộc hàm $X \rightarrow Y$ trong F sẽ cũng có các giá trị giống nhau đối với các thuộc tính của vế phải Y. Có thể chỉ ra rằng sau khi áp dụng vòng lặp của bước 3, nếu một hàng bất kỳ trong S kết thúc với toàn ký hiệu "1" thì D có tính chất nối không mất mát đối với F. Mặt khác, nếu không có hàng nào kết thúc bằng tất cả ký hiệu "1" thì D không thỏa mãn tính chất nối không mất mát. Trong trường hợp sau, trạng thái quan hệ r được biểu diễn bằng S ở cuối thuật toán sẽ là một ví dụ về một trạng thái quan hệ r của R thỏa mãn các phụ thuộc trong F nhưng không thỏa mãn điều kiện nối không mất mát . Như vậy, quan hệ này được dùng như một phản ví dụ chứng minh rằng D không có tính chất nối không mất mát đối với F. Chú ý rằng các ký hiệu "1" và "0" không có ý nghĩa đặc biệt gì ở cuối thuật toán.

Ví dụ áp dụng 1:

R = (MãsốNV, TênNV, MãsốDA, TênDA, ĐịađiểmDA, Sốgiờ)

R1= (TênNV, ĐịađiểmDA)

R2 = (MãsốNV, MãsốDA, Sốgiờ, TênDA, ĐịađiểmDA)

F= { Mã sốNV→ TênNV, MãsốDA → {TênDA, ĐịađiểmDA}, {MãsốNV, Mã sốDA}→ Sốgiờ}

	MãsốNV	TênNV	Mã sốDA	TênDA	ĐịađiểmDA	Sốgiờ
R1	0	1	0	0	1	0
R2	1	0	1	1	1	1

Xét lần lượt phụ thuộc hàm MãsốNV → TênNV, MãsốDA → {TênDA, ĐịađiểmDA}, {MãsốNV, Mã sốDA} → Sốgiờ. Ta thấy không có trường hợp nào các thuộc tính tương ứng với các vế trái đều có giá trị bằng 1, vì vậy ta không thể làm gì để biến đổi ma trận. Ma trận không chứa một hàng gồm toàn ký hiệu "1". Phép tách là mất mát.

Ví dụ áp dụng 2:

R = (MãsốNV, TênNV, MãsốDA, TênDA, ĐịađiểmDA, Sốgiờ)

R1= (MãsốNV, TênNV)

R2 = (MãsốDA, TênDA, ĐịađiểmDA)

R3 = (MãsốNV, MãsốDA, Sốgiờ)

F= {Mã sốNV→ TênNV, MãsốDA → {TênDA, ĐịađiểmDA}, {MãsốNV, MãsốDA} → Sốgiờ}

	MãsốNV	TênNV	Mã sốDA	TênDA	ĐịađiểmDA	Sốgiờ
R1	1	1	0	0	0	0
R2	0	0	1	1	1	0
R3	1	0	1	0	0	1

(Giá trị ban đầu của ma trận S)

	MãsốNV	TênNV	Mã sốDA	TênDA	ĐịađiểmDA	Sốgiờ
R1	1	1	0	0	0	0
R2	0	0	1	1	1	0
R3	1	0 1	1	0 1	0 1	1

Sau khi áp dụng hai phụ thuộc hàm đầu tiên, ma trận S chứa một hàng gồm toàn ký hiệu 1. Phép tách này là không mất mát.

Hình 8-3. Thuật toán kiểm tra nối không mất mát

Thuật toán 8.2 cho phép chúng ta kiểm tra xem một phép tách D cụ thể có tuân theo tính chất nối không mất mát hay không. Câu hỏi tiếp theo là liệu có một thuật toán tách một lược đồ quan hệ vũ trụ R = {A_1, A_2, …, A_n} thành một phép tách D = {R_1, R_2, …,R_m} sao cho mỗi R_i là ở BCNF và phép tách D có tính chất nối không mất mát đối với F hay không? Câu trả lời là có. Trước khi trình bày thuật toán, ta xem một số tính chất của các phép tách nối không mất mát nói chung.

Tính chất 1: Một phép tách D = {R_1,R_2} của R có tính chất nối không mất mát đối với một tập phụ thuộc hàm F trên R khi và chỉ khi

- Hoặc phụ thuộc hàm ((R_1? R_2) → (R_1? R_2)) ở trong F^+.
- Hoặc phụ thuộc hàm ((R_1? R_2) → (R_2 ? R_1)) ở trong F^+.

Với tính chất này, chúng ta có thể kiểm tra lại các phép tách chuẩn hóa trong 4.3 và sẽ thấy rằng các phép tách đó là thỏa mãn tính chất nối không mất mát.

Tính chất 2: Nếu một phép tách D = {R_1, R_2, …, R_m} của R có tính chất nối không mất mát đối với một tập phụ thuộc hàm F trên R và nếu một phép tách D_1 = {Q_1, Q_2, …,Q_k} của R_i có tính chất nối không mất mát đối với phép chiếu của F trên R_i thì phép tách D2 = { R_1, R_2,…, R_{i-1}, Q_1, Q_2,…,Q_k, R_{i+1},…, R_m} của R có tính chất nối không mất mát đối với F.

Tính chất này nói rằng nếu một phép tách D đã có tính chất nối không mất mát đối với một tập F và chúng ta tiếp tục tách một trong các quan hệ R_i trong D thành phép tách khác D_1 (l = 1,2,..k) có tính chất nối không mất mát đối với π_{Ri}(F) thì việc thay R_i trong D bằng D_1 (l = 1,2,..k) cũng tạo ra một phép tách có tính chất nối không mất mát đối với F.

Thuật toán 8.3 sau đây sử dụng hai tính chất trên để tạo ra một phép tách D = {R_1, R_2, …, R_m} của một quan hệ vũ trụ R dựa trên một tập các phụ thuộc hàm F sao cho mỗi R_i là BCNF.

Thuật toán 8.3: Tách quan hệ thành các quan hệ BCNF với tính chất nối không mất mát.

Input: Một quan hệ vũ trụ R và một tập hợp các phụ thuộc hàm F trên các thuộc tính của R.

1. Đặt D := {R} ;
2. Khi có một lược đồ quan hệ Q trong D không phải ở BCNF, thực hiện vòng lặp: Với mỗi một lược đồ quan hệ Q trong D không ở BCNF hãy tìm một phụ thuộc hàm X→ Y trong Q vi phạm BCNF và thay thế Q trong D bằng hai lược đồ quan hệ (Q-Y) và (X?Y). Quá trình lặp dừng khi không còn quan hệ nào trong D vi phạm BCNF.

Mỗi lần đi vào vòng lặp trong thuật toán 8.3, chúng ta tách một quan hệ Q không phải BCNF thành hai lược đồ quan hệ. Theo các tính chất 1 và 2, phép tách D có tính chất nối không mất mát. Kết thúc thuật toán, tất cả các quan hệ trong D sẽ ở BCNF.

Trong bước 2 của thuật toán 8.3, cần xác định xem một lược đồ quan hệ Q có ở BCNF hay không. Một phương pháp để làm điều đó là kiểm tra. Với mỗi phụ thuộc hàm X → Y trong Q, ta tính X^+. Nếu X^+ không chứa tất cả các thuộc tính trong Q thì X → Y vi phạm BCNF bởi vì X không phải là một siêu khóa.

Một kỹ thuật nữa dựa trên quan sát rằng khi một lược đồ quan hệ Q vi phạm BCNF thì có tồn tại một cặp thuộc tính A,B trong Q sao cho {Q – {A,B}} → A. Bằng việc tính bao đóng $\{Q - \{A,B\}\}^+$ cho mỗi cặp thuộc tính {A,B} của Q và kiểm tra xem bao đóng có chứa A (hoặc B) hay không, chúng ta có thể xác định được Q có ở BCNF hay không.

Ví dụ áp dụng: Xét lược đồ quan hệ

R = { A, B, C, D, E, F)

Với các phụ thuộc hàm:

A → BCDEF, BC → ADEF, B→ F, D→ E, D→ B

Lược đồ quan hệ này có hai khóa dự tuyển là A và BC.

Ta có B → F vi phạm BCNF vì B không phải là siêu khóa, R được tách thành:

R1(B,F) với phụ thuộc hàm B→ F

R2(A,B,C,D,E) với các phụ thuộc hàm A→SCDE, BC→ADF, D→E, D→B

Do D→ E vi phạm BCNF (D là một thuộc tính không khóa), R2 được tách thành:

R21(D,E) với phụ thuộc hàm D → E

R22(ABCD) với các phụ thuộc hàm A → BCD, BC→ AD, D→ B

Do D B vi phạm BCNF (Dkhông phải là thuộc tính khóa), R22 được tách thành:

R221(D,B)

R222(A,B,D) với phụ thuộc hàm A → BD (phụ thuộc hàm BC → AD bị mất)

Tóm lại, ta có phép tách D = {R1, R21, R221, R222}. Phép tách này có tính chất nối không mất thông tin nhưng không bảo toàn phụ thuộc.

Nếu chúng ta muốn có một phép tách có tính chất nối không mất mát và bảo toàn phụ thuộc thì ta phải hài lòng với các lược đồ quan hệ ở dạng 3NF. Thuật toán sau đây là cải tiến của thuật toán 8.1, tạo ra một phép tách thỏa mãn :

- Bảo toàn phụ thuộc.

- Có tính chất nối không mất mát.

- Mỗi lược đồ quan hệ kết quả là ở dạng 3NF.

Thuật toán 8.4: Thuật toán tổng hợp quan hệ với tính chất bảo toàn phụ thuộc và nối không mất mát.

Input: Một quan hệ vũ trụ R và một tập các phụ thuộc hàm F trên các thuộc tính của R.

1. Tìm phủ tối thiểu G cho F.
2. Với mỗi vế trái X của một phụ thuộc hàm xuất hiện trong G hãy tạo ra một lược đồ quan hệ trong D với các thuộc tính {X?{A_1}?{A_2}?…? {A_k}}, trong đó X →A_1, X→A_2,…, X→ A_k chỉ là các phụ thuộc hàm ở trong G với X là vế trái (X là khóa của quan hệ này).
3. Nếu không có lược đồ quan hệ nào trong D chứa một khóa của R thì hãy tạo ra thêm một lược đồ quan hệ trong D chứa các thuộc tính tạo nên một khóa của R.

Bước 3 của thuật toán 8.4 đòi hỏi phải xác định một khóa K của R. Để xác định một khóa K của R, ta sử dụng thuật toán sau

Thuật toán 8.5: Tìm một khóa K của R dựa trên tập F các phụ thuộc hàm.

1. Đặt K := R;
2. Với mỗi thuộc tính A trong K

{tính $(K-A)^+$ đối với F;

Nếu $(K-A)^+$ chứa tất cả các thuộc tính trong R thì đặt K := K-{A}};

*Chú ý: Chúng ta có nhận xét sau: Nếu quan hệ có khóa thì các thuộc tính khóa của quan hệ phải là các tập con của tập hợp các thuộc tính ở vế phải các phụ thuộc hàm trong F. Vì vậy, để tìm được các khóa nhanh hơn, trước tiên chúng ta tính R_F là hợp của các thuộc tính ở các vế trái của các phụ thuộc hàm trong F, sau đó đi tính bao đóng của tất cả các tập con của R_F. Nếu bao đóng của tập con nào chứa tất cả các thuộc tính của R thì tập đó là một siêu khóa. Để kiểm tra nó là một khóa ta thực hiện như bước 2) của thuật toán trên.

Thuật toán 8.6: xác định tất cả các khóa của lược đồ:

Cho lược đồ quan hệ α = (U, F) hãy tìm tất cả các khoá của lược đồ quan hệ α.

B1) Xác định I_α

B2) Nếu $I_\alpha^+ = U$ thì kết luận lược đồ có một khoá duy nhất là I_α và kết thúc thuật toán. Ngược lại $I_\alpha^+ \neq U$ thì chuyển sang bước tiếp theo.

B3) Xác định N={ ? (R_i -L_i) sao cho $L_i \subseteq I_\alpha$ }

- đặt $N' = (I_\alpha N)^+ \setminus I_\alpha$ ($N' \subseteq N_\alpha$)

- đặt $B = U \setminus N' \setminus I_\alpha$

B4) Nếu | B |=2 (tập B chỉ gồm có hai thuộc tính), giả sử $B=B_1B_2$ khi đó lược đồ chỉ có hai khoá là $I_\alpha B_1$ và $I_\alpha B_2$, kết thúc thuật toán. Ngược lại nếu | B |>2 thì chuyển sang bước tiếp theo.

B5) Tìm tất cả các tập con khác rỗng của B, ký hiệu các tập con đó là $\{B_1, B_2, .. B_n\}$. Tính $(B_i?I_\alpha)^+$ (i=1..n), nếu $(B_i?I_\alpha)^+ = U$ thì đặt $M_i = B_i?I_\alpha$.

B6) Khi đó $M= \{M_1, M_2, \ldots, M_h\}$ là họ tất cả các siêu khoá của lược đồ α.

B7) Loại bỏ các siêu khoá không tối tiểu ra khỏi M, tức là nếu $\exists M_i \subseteq M_j$ (*) ($i \neq j$, i, j = 1,*h*) thì loại M_j ra khỏi M. Tập M thu được sau khi loại bỏ tất các phần tử thoả mãn biểu thức (*) chính là họ tất cả các khoá của lược đồ α.

Nhận xét: Từ một tập B gồm n phần tử ta có thể lập lên 2^n-1 tập con khác rỗng của B. Cho nên dù n là một số không lớn lắm thì 2^n-1 cũng là một số tương đối lớn. Do vậy bài toán tìm tất cả các khoá của lược đồ quan hệ có thể coi là một bài toán khó (vì có độ phức tạp thuật toán là hàm mũ).

Không phải lúc nào cũng có khả năng tìm được một phép tách thành các lược đồ quan hệ bảo toàn phụ thuộc và mỗi lược đồ trong phép tách là ở BCNF. Các lược đồ quan hệ trong phép tách theo thuật toán ở trên thường là 3NF. Để có các lược đồ BCNF, chúng ta có thể kiểm tra các lược đồ quan hệ 3NF trong phép tách một cách riêng rẽ để xem nó có thỏa mãn BCNF không. Nếu có lược đồ quan hệ R_i không ở BCNF thì ta có thể tách tiếp hoặc để nguyên nó là 3NF.