ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
----------o0o----------

Giáo trình

THIẾT KẾ CƠ SỞ DỮ LIỆU

Biên soạn: Trịnh Minh Tuấn

2005

LỜI NÓI ĐẦU

Trong giáo trình **Cơ Sở Dữ Liệu** đã trình bày các khái niệm cơ sở cũng như các ứng dụng trên mô hình cơ sở dữ liệu quan hệ.

Giáo trình Thiết kế Cơ Sở Dữ Liệu là một giáo trình chuyên sâu, đưa ra các phương pháp nhằm giúp học viên thiết kế một cơ sở dữ liệu tốt. Với phương châm xây dựng một giáo trình vừa đáp ứng yêu cầu chuẩn mực của sách giáo khoa, vừa có giá trị thực tiễn và phù hợp với phương thức học tập qua mạng, trong đó chủ yếu là khả năng tự học, tự nghiên cứu của học viên, chúng tôi đã tham khảo nhiều tài liệu có giá trị của các tác giả trong và ngoài nước. Giáo trình nầy được dùng kèm với giáo trình điện tử trên dĩa CD.

Tuy có rất nhiều cố gắng trong công tác biên soạn, nhưng do lần đầu tiên xuất bản nên chắc chắn sẽ có ít nhiều khiếm khuyết. Chúng tôi rất trân trọng tiếp thu những ý kiến đóng góp của bạn đọc để hoàn thiện giáo trình trong các lần tái bản kế tiếp.

MỤC LỤC

Chương I: MÔ HÌNH QUAN HỆ

I. MÔ HÌNH QUAN HỆ

I.1. Các khái niệm cơ bản

Khái niệm toán học của mô hình quan hệ là quan hệ hiểu theo nghĩa lý thuyết tập hợp : là tập của con của tích Đề - Các của các miền; *miền (domain)* là một tập các giá trị. Ví dụ tập các số nguyên là một miền; tập các xâu ký tự tạo thành tên người trong tiếng Anh có độ dài không quá 30 ký tự là một miền; tập hai số {0,1} cũng là một miền v.v…

Gọi $D_1, D_2, \ldots, D_n$ là n miền. Tích Đề - Các của n miền ký hiệu : $D_1 x D_2 x \ldots x D_n$ là tập tất cả *n -bộ (n- tuples)* $(v_1, v_2, \ldots v_n)$ sao cho $v_i \in D_i$, với $i = 1 \ldots n$

Thí dụ:
Với n= 2, $D_1 = \{0,1\}$, $D_2 = \{a,b,c\}$, khi đó
$D_1 \times D_2 = \{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}$.

Quan hệ : Quan hệ là một tập con của tích Đề - Các của một hoặc nhiều miền. Như vậy mỗi quan hệ có thể là vô hạn. Ở đây luôn luôn giả thiết rằng, quan hệ là một tập hữu hạn.

Mỗi hàng của quan hệ gọi là *bộ (tuples)*, quan hệ là tập con của tích Đề - Các $D_1 \times D_2 \times \ldots \times D_n$ gọi là quan hệ n ngôi. Khi đó mỗi bộ của của quan hệ có n thành phần (n cột). Các cột của quan hệ gọi là *thuộc tính (attributes)*. Định nghĩa quan hệ một cách hình thức như sau:

I.1.1. Định nghĩa I.1

Gọi R = {A_1, A_2, …, A_n } là tập hữu hạn các thuộc tính, mỗi thuộc tính A_i với i = 1,2,…, n có miền giá trị tương ứng là dom(A_i). Quan hệ r được định nghĩa trên tập thuộc tính R là tập con của tích Đề - Các của các miền.

$r \subseteq dom(A_1) \times dom(A_2) \times \ldots \times dom(A_n)$

Khi đó ký hiệu là r(R) hoặc r (A_1,…A_n).

Thí dụ:

Hình I.1 cho thấy quan hệ NHANVIEN bao gồm các thuộc tính HOTEN, NAMSINH, NOILAMVIEC là một quan hệ 3 ngôi.

NHANVIEN	(HoTen,	NamSinh,	NoiLamViec)
t_1	Lê Văn A	1960	Trường ĐHVL
t_2	Hoàng Thị B	1970	Trường ĐHBK

- Hình I.1 - quan hệ NHANVIEN -

t_1=(Lê Văn A ,1960, Trường DHVL) là một bộ của quan hệ NHANVIEN

Lược đồ quan hệ là sự trừu tượng hóa của quan hệ, một sự trừu tượng hóa ở mức độ *cấu trúc* của một bảng 2 chiều. Khi nói đến lược đồ quan hệ tức là đề cập đến cấu trúc tổng quát của một quan hệ, đó là các thuộc tính và mối liên hệ ngữ nghĩa giữa chúng.

Ký hiệu : lược đồ quan hệ R

Thể hiện (còn gọi là tình trạng) của quan hệ là tập hợp các bộ giá trị của quan hệ vào một thời điểm. Tại những thời điểm khác nhau quan hệ sẽ có những thể hiện khác nhau.

Ký hiệu : thể hiện r

Lưu ý: Khi cho quan hệ r, ta muốn nói đến một thể hiện cụ thể của quan hệ đó. Nghĩa là r là tập hợp gồm các bộ cụ thể.

Thí dụ:

Cho *lược đồ quan hệ* R = $\{A_1, A_2, \ldots, A_n\}$, với A_i là các *thuộc tính*, gọi r là một *quan hệ (thể hiện)* của *lược đồ quan hệ* R. *Quan hệ* r gồm có các *bộ* sau :

$t_1=(a_{11},a_{21}, \ldots,a_{n1})$

$t_2=(a_{12},a_{22}, \ldots,a_{n2})$

Ta có: quan hệ r $\in$ lược đồ quan hệ R; bộ t_1 $\in$ quan hệ r

I.2. Khoá

I.2.1. Định nghĩa I.2

Cho lược đồ quan hệ R định nghĩa trên tập các thuộc tính $U=\{A_1,\ldots A_n\}$. $K \subseteq U$ *là khoá (key) của lược đồ quan hệ R nếu thoả 2 điều kiện sau đây :*

(i) K xác định đươc mọi giá trị của A_j, *với j=1, 2, ..., n*

(ii) Không tồn tại $K' \subset K$ $(K' <> K)$ *mà K' cũng thỏa (i)*

Điều kiện (i) nghĩa là: với một quan hệ bất kỳ r $\in$ lược đồ R, và với bất kỳ hai bộ t1, t2 $\in$ quan hệ r đều tồn tại một thuộc tính A $\in$ K sao cho $t_1[A] \neq t_2[A]$. Nói cách khác, không tồn tại hai bộ mà có giá trị bằng nhau trên mọi thuộc tính của K. Điều kiện này có thể viết $t_1[K] \neq t_2[K]$. Do vậy mỗi giá trị của K là xác định duy nhất.

Giả sử K là khóa thì mọi tập K" ⊆ U mà K" ⊇ K thì K" cũng thoả (i). Các tập K" thoả điều kiện (i) được gọi là *siêu khoá* (super key) còn gọi là *khoá bao hàm*. Điều kiện (ii) xác định khoá là tập *nhỏ nhất* trong một họ các siêu khoá.

Trong lược đồ quan hệ có thể có nhiều khoá. Khi cài đặt trên một hệ quản trị cơ sở dữ liệu ta phải chọn một để làm *khóa chính* (primary key).

HANGHOA	(MSMH	TENHANG	SOLUONG)
	10101	sắt phi 6	1000
	10102	sắt phi 8	2000
	20001	xi măng	1000

- Hình I.2 - quan hệ HANGHOA -

Trong hình I.2 biểu diễn quan hệ HANGHOA trong đó mã số mặt hàng (MSMH) là khoá. Mỗi giá trị MSMH đều xác định duy nhất một loại mặt hàng trong quan hệ HANGHOA.

I.3. Các phép tính trên CSDL quan hệ

Các phép tính cơ bản mà nhờ đó một cơ sở dữ liệu được thay đổi là chèn (INSERT). loại bỏ (DELETE) và cập nhật (UPDATE).

Trong mô hình CSDL quan hệ được nêu trên, các phép tính này được áp dụng cho từng bộ phận của các quan hệ lưu trữ trong máy - việc tổ chức các quan hệ và các bộ của nó có thể được xem như biểu diễn tương ứng một - một qua các tệp (file) và các bản ghi (record).

I.3.1. Phép chèn (INSERT)

Phép chèn thêm một bộ t vào quan hệ r của lược đồ R= $\{A_1,\ldots,A_n\}$ có dạng r= r ∪ {t}

INSERT (r ; $A_1=d_1$, $A_2=d_2,\ldots, A_n=d_n$).

Trong đó A_i với i= 1,…, n là tên các thuộc tính và di ∈ dom (A_i) là các giá trị thuộc miền trị tương ứng của thuộc tính A_i.

Thí dụ:

Thêm một bộ t_3 = (Vũ Văn Tần, 1960, trường ĐHBK) vào quan hệ NHANVIEN trong hình I.1:

INSERT (NHANVIEN ; HOTEN = Vũ Văn Tần, NAMSINH = 1960, NOILAMVIEC = trường ĐHBK)

Nếu xem thứ tự trường là cố định, khi đó có thể biểu diễn phép chèn dưới dạng không tường minh như sau:

INSERT (r ; d_1, d_2,…., d_n).

Mục đích của phép chèn là thêm một bộ mới vào một quan hệ nhất định. Kết quả của phép tính này có thể gây ra một số sai sót với những lý do sau đây:

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

Do vậy, tuỳ từng trường hợp cụ thể sẽ có những cách khắc phục riêng.

I.3.2. Phép loại bỏ (DELETE)

Phép loại bỏ là phép xoá một bộ ra khỏi một quan hệ cho trước. Giống như phép chèn, phép loại bỏ có dạng: r = r – {t}

DELETE (r ; A_1=d_1, A_2=d_2,…., A_n=d_n) hoặc

DELETE (r ; d_1, d_2,…., d_n).

Thí dụ:

Cần loại bỏ bộ t_1 từ quan hệ NHANVIEN trong hình I.1:

DELETE (NHANVIEN ; Lê Văn A, 1960, Trường DHVL)

Tất nhiên không phải lúc nào phép loại bỏ cũng cần đầy đủ thông tin về cả bộ. Nếu có giá trị về bộ đó tại các thuộc tính khoá $K = \{B_1, \ldots B_i\}$ khi đó phép loại bỏ chỉ cần viết:

DELETE (r; $B_1 = e_1$, $B_2 = e_2$, $\ldots B_i = e_i$)

Thí dụ:

Cần loại bỏ sắt phi sáu ra khỏi quan hệ HANGHOA trong hình I.2, khi đó chỉ cần viết:

DELETE (HANGHOA; MSMH = 10101)

I.3.3. Phép cập nhật (UPDATE)

Trong thực tế không phải lúc nào cũng chỉ dùng phép chèn hoặc loại bỏ đi một bộ mà nhiều khi chỉ cần sửa đổi một số giá trị nào đó tại một số thuộc tính, lúc đó cần thiết phải sử dụng phép cập nhật.

Gọi tập $\{C_1, \ldots, C_p\} \subseteq \{A_1, \ldots, A_n\}$ là tập các thuộc tính mà tại đó các giá trị của bộ cần thay đổi, khi đó phép cập nhật có dạng : $r = (r \setminus \{t\}) \cup \{t'\}$.

UPDATE (r; $A_1=d_1$, $A_2=d_2, \ldots, A_n=d_n$;$C_1=e_1$, $C_2=e_2$,…, $C_p=e_p$)

Nếu $K= \{B_1, \ldots, B_m\}$ là khoá của quan hệ, khi đó chỉ cần viết:

UPDATE (r; $B_1=d_1$, $B_2=d_2, \ldots, B_m=d_m$;$C_1=e_1$, $C_2=e_2$,…, $C_p=e_p$)

Thí dụ:

Cần thay đổi số lượng của sắt phi 8 trong quan hệ HANGHOA trong hình I.2, còn 150 tấn

UPDATE (HANGHOA; MSMH=10102; SOLUONG =150)

Phép cập nhật là phép tính rất thuận lợi, hay dùng. Cũng có thể không dùng phép cập nhật mà dùng tổ hợp của phép loại bỏ và phép chèn một bộ mới. Do vậy những sai sót của phép cập nhật cũng sẽ xảy ra tương tự như phép chèn và phép loại bỏ.

II. ĐẠI SỐ QUAN HỆ

Trong chương này trình bày nguyên tắc tiếp cận để thiết kế các ngôn ngữ biểu diễn câu hỏi về các quan hệ. Đối tượng của ngôn ngữ thao tác dữ liệu quan hệ hay còn gọi là "ngôn ngữ hỏi" (query language), thường liên quan chặt chẽ với các phép tính chèn , loại bỏ, cập nhật các bộ của quan hệ. Mặt khác các câu hỏi có thể xem trong trường hợp tổng quát là những hàm số áp dụng lên các quan hệ. Ngôn ngữ hỏi cho mô hình quan hệ được chia hai lớp :

- Ngôn ngữ đại số, trong đó câu hỏi được biểu diễn nhờ áp dụng các phép tính đặc biệt đối với quan hệ và
- Ngôn ngữ tính toán tân từ, trong đó câu hỏi được biểu diễn là một tập hợp các bộ thoả mãn các tân từ xác định.

Dưới đây sẽ trình bày chi tiết ngôn ngữ đại số quan hệ như là cơ sở của một ngôn ngữ bậc cao để thao tác trên quan hệ.

Gọi r là quan hệ trên tập thuộc tính $R = \{A_1,\ldots, A_n\}$.

Ở đây luôn giả thiết rằng quan hệ r là tập hữu hạn các bộ. Đối với các phép hợp, giao và trừ, hai quan hệ tham gia phải là *khả hợp*.

Hai quan hệ được gọi là khả hợp nếu chúng giống nhau đôi một các thuộc tính. Các thuộc tính có thể khác tên gọi nhưng phải cùng miền giá trị.

II.1. Phép hợp

Hợp của hai quan hệ r và s khả hợp, ký hiệu là $r \cup s$ là tập các bộ t thuộc quan hệ r hay thuộc quan hệ s.

Biểu diễn hình thức phép hợp có dạng:

$r \cup s = \{ t / t \in r \text{ hay } t \in s \}$

Thí dụ:

r (A B C)

A	B	C
a_1	b_1	c_1
a_2	b_1	c_2
a_2	b_2	c_1

s (A B C)

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$r \cup s$ = (A B C)

A	B	C
a_1	b_1	c_1
a_2	b_1	c_2
a_2	b_2	c_1
a_2	b_2	c_2

II.2. Phép giao

Giao của hai quan hệ r và s khả hợp, ký hiệu là $r \cap s$ là tập hợp các bộ t thuộc cả quan hệ r và s.

Biểu diễn hình thức phép giao có dạng:

$r \cap s = \{ t / t \in r \text{ và } t \in s \}$

Thí dụ:

Với r và s là hai quan hệ ở ví dụ trên, giao của chúng là:

$r \cap s$ = (A B C)

A	B	C
a_1	b_1	c_1

II.3. Phép trừ

Hiệu của hai quan hệ r và s khả hợp, ký hiệu là r-s, là tập các bộ t thuộc r nhưng không thuộc s.

Biểu diễn hình thức phép có dạng:

$r - s = \{ t / t \in r \text{ và } t \notin s \}$

Thí dụ:

Cũng với r, s ở ví dụ trên:

r - s = (A	B	C)
a_2	b_1	c_2
a_2	b_2	c_1

Chú ý: Phép giao của 2 quan hệ r ∩ s có thể biểu diễn qua phép trừ: r ∩ s = r-(r - s)

II.4. Tích Đề-Các

Gọi r là quan hệ xác định trên tập thuộc tính $\{A_1,A_2,...,A_n\}$ và s là quan hệ xác định trên tập thuộc tính $\{B_1,B_2,...,B_m\}$. Tích Đề-Các của quan hệ r và s, ký hiệu r x s, là tập (n+m) –bộ với n thành phần đầu có dạng một bộ thuộc r và n thành phần sau có dạng một bộ thuộc s.

Biểu diễn hình thức có dạng

r x s = {t / t có dạng (a1,a2,...,an,b1,b2,...,bm), trong đó (a1,a2,...,an) ∈ r và (b1,b2,...,bm) ∈ s }

Thí dụ:

r (A	B	C)
a_1	b_1	c_1
a_2	b_2	c_2

s (D	E	F)
d	e	f
d’	e’	f’

r x s = (A	B	C	D	E	F)
a_1	b_1	c_1	d	e	f
a_1	b_1	c_1	d’	e’	f’
a_2	b_2	c_2	d	e	f
a_2	b_2	c_2	d’	e’	f’

II.5. Phép chiếu

Phép chiếu trên một quan hệ thực chất là loại bỏ đi một số thuộc tính và giữ lại những thuộc tính khác của quan hệ đó. Giả sử r là một quan hệ n ngôi (gồm n thuộc tính): R = $\{A_1,\ldots, A_n\}$, Viết $\prod_{Ai1Ai2\ldots Aim}(r)$,

$\{A_{i1},\ldots, A_{im}\} \subseteq R$. Khi đó hiểu rằng phép chiếu trên các thuộc tính $A_{i1},\ldots, A_{im}$ của quan hệ r sẽ được tập các bộ có dạng ai1 …aim. Để thuận tiện cho việc biểu diễn hình thức phép chiếu, từ đây quy định một số ký hiệu như sau:

Gọi t là một bộ thuộc r, $A \in R$. t[A] là giá trị của bộ t tại thuộc tính A. $X \subseteq R$. Với $X=\{B_1,\ldots, B_m\}$thì $t[X] = (t[B_1], t[B_2], \ldots, t[B_n])$.

Gọi X là tập con của thuộc tính R. Phép chiếu trên tập X của quan hệ r, ký hiệu là $\prod_X(r)$ (hoặc ký hiệu r[X]) được định nghĩa như sau:

$\prod_X(r) = \{t[X] / t \in r\}$

Thí dụ:

R = {A, B, C, D} , X= {A, B} ; Y = {A, C}

r (A, B, C, D) $\prod_X(r)$ = (A, B) ; $\prod_Y(r)$ = (A, C)

r (A, B, C, D)	$\prod_X(r)$ = (A, B)	$\prod_Y(r)$ = (A, C)
a_1 b_1 c_1 d_1	a_1 b_1	a_1 c_1
a_1 b_1 c_1 d_2	a_2 b_2	a_2 c_2
a_2 b_2 c_2 d_2		a_2 c_3
a_2 b_2 c_3 d_3		

II.6. Phép chọn

Phép chọn là phép tính để xây dựng một tập con các bộ của quan hệ đã cho, thoả mãn biểu thức F xác định. Biểu thức F được diễn tả bằng một tổ hợp Boolean của các toán hạng, mỗi toán hạng là một phép so sánh đơn giản giữa hai biến là hai thuộc tính hoặc giữa một biến là một thuộc tính và một hằng, cho giá trị "đúng" hoặc "sai" đối với mỗi bộ đã cho khi kiểm tra riêng bộ ấy.

Các phép so sánh trong biểu thức F là <, =, >=. <= và ≠; Các phép logic là ∧ (và) ∨ (hoặc) ¬ (không).

Gọi F(t) là một biểu thức logic có biến là các bộ t. Phép chọn các bộ của quan hệ r dựa theo điều kiện F, ký hiệu là $\sigma_F(r)$ *(hoặc ký hiệu r:(F)) được định nghĩa như sau:*

$\sigma_F(r) = \{ t \in r \mid F(t) = \text{đúng} \}$

F(t) được biểu thị là giá trị của các thuộc tính xuất hiện trong biểu thức F tại bộ t thoả các điều kiện của F.

Thí dụ:

Cho quan hệ ở hình sau. Các phép chọn

r (A, B, C, D)

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_2	c_3	d_3

$\sigma_{A = a1}(r)$ =

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_1	d_2

$\sigma_{A = a1 \wedge D=d2}(r)$ =

A	B	C	D
a_1	b_1	c_1	d_2

II.7. Phép kết nối

Để định nghĩa phép kết nối của các quan hệ, trước hết làm quen với khái niệm *xếp cạnh nhau*. Giả sử cho bộ

$t = (t_1, t_2, \ldots, t_n)$ và bộ $u = (u_1, u_2, \ldots u_m)$,

phép *xếp cạnh nhau* của d và e định nghĩa qua

$(t,u) = (t_1, t_2, \ldots, t_n, u_1, u_2, \ldots u_m)$

Gọi θ là một trong các phép so sánh { =,>, >=,<, <=, ≠}.

Phép kết nối của quan hệ r đối với thuộc tính A với quan hệ s đối với thuộc tính B được định nghĩa qua

$$r \underset{A\theta B}{><} s = \{ (t,u) \ / \ t \in r;\ u \in s \text{ và } t[A]\ \theta\ u[B] \}.$$

Dĩ nhiên, ở đây cần giả thiết rằng mỗi giá trị của cột r[A] đều có thể sánh được (qua phép θ) với mỗi giá trị của cột s[B].

Trong trường hợp phép so sánh θ là "=" gọi là kết nối bằng. Trường hợp kết nối bằng tại thuộc tính cùng tên của hai quan hệ và một trong hai thuộc tính đó được loại bỏ qua phép chiếu, thì phép kết nối được gọi là "kết nối tự nhiên" và sử dụng ký hiệu "*". Khi đó phép kết nối tự nhiên của hai quan hệ r(ABC) và s(CDE) biểu diễn qua:

$$r(ABC)*s(CDE)=\{t[ABCDE]/t[ABC] \in r \text{ và } t[CDE]\in s\}$$

Thí dụ:

r(A B C)	s(C D E)	$r \underset{B\geq C}{><} s$ = (A B C C D E)
a_1 1 1	1 d_1 e_1	a_1 1 1 1 d_1 e_1
a_1 2 1	2 d_2 e_2	a_2 2 1 1 d_1 e_1
a_1 2 2	3 d_3 e_3	a_2 2 1 2 d_2 e_2
		a_1 2 2 1 d_1 e_1
		a_1 2 2 2 d_2 e_2

Kết quả nối tự nhiên:

r(ABC) *s(CDE) =

A	B	C	D	E
a_1	1	1	d_1	e_1
a_2	2	1	d_1	e_1
a_2	2	2	d_2	e_2

II.8. Phép chia

Gọi r là quan hệ n – ngôi và s quan hệ m – ngôi (n>m, s ≠ ∅). Phép chia r ÷ s là tập của tất cả (n-m) bộ t sao cho với mọi bộ u ∈ s thì bộ (t,u) ∈ r.

Thí dụ:

r(A B C D)

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

s(C D)

C	D
c	d
e	f

r ÷ s = (A B)

A	B
a	b
e	d

II.9. Các ví dụ về tìm kiếm bằng đại số quan hệ

Thí dụ:

Cho ba quan hệ:

S (S#, SNAME, STATUS, CITY): các hãng cung ứng

P (P#, PNAME, COLOR, WEIGHT, CITY):các mặt hàng

SP (S#, P#, QTY): các mặt hàng đã cung cấp.

- Tìm số hiệu của những hãng đã cung ứng mặt hàng P2.

$\prod_{S\#} (\sigma_{P\# = 'P2'} (SP))$

- Tìm số hiệu của những hãng cung ứng ít nhất là một mặt hàng màu đỏ

$\prod_{S\#} (\sigma_{COLOR = 'RED'} (P*SP))$

hoặc

$\prod_{S\#} (\sigma_{COLOR = 'RED'} (P))*SP)$

Chương II: CÁC PHỤ THUỘC DỮ LIỆU TRONG MÔ HÌNH QUAN HỆ

I. MỞ ĐẦU

I.1. Thế nào là một thiết kế cơ sở dữ liệu kém

Trước khi bàn về cách thiết kế một lược đồ cơ sở dữ liệu tốt, chúng ta phải phân tích xem tại sao trong một số lược đồ lại tồn tại những vấn đề rắc rối.

Ví dụ cho lược đồ về *thông tin cung cấp* như sau:

TT_CUNGCAP(NHACC, DIACHI, MATHANG, GIA)

XMHT	, 123	, P400	,50000
XMHT	, 123	, P500	,52000
XMSM	, 456	, P400	,49000

- Hình II.1 - quan hệ TT_CUNGCAP -

Lược đồ này chứa tất cả các thông tin về nhà cung cấp như: tên nhà cung cấp, địa chỉ nhà cung cấp, mặt hàng mà họ có thể cung cấp và giá. Chúng ta có thể nhận thấy nhiều vấn đề trong đó:

1. Dư thừa (redundancy). Địa chỉ của nhà cung cấp được lập lại mỗi lần cho mỗi mặt hàng được cung cấp.
2. Mâu thuẫn tiềm ẩn (potential inconsistancy) hay bất thường khi cập nhật. Do hậu quả của dư thừa, chúng ta có thể thay đổi địa chỉ của một nhà cung cấp trong một bộ nhưng vẫn để lại địa chỉ cũ trong một bộ khác. Vì vậy chúng ta có thể không có một địa chỉ duy nhất đối với mỗi nhà cung cấp như chúng ta tưởng.

3. Bất thường khi chèn (insertion anomaly). Chúng ta không thể biết địa chỉ của nhà cung cấp nếu hiện tại họ không cung cấp ít nhất một mặt hàng. Chúng ta có thể đặt những giá trị null trong các thành phần MATHANG và GIA của một bộ cho người đó, nhưng khi chúng ta nhập một mặt hàng cho nhà cung cấp đó, chúng ta có nhớ xoá đi bộ mang giá trị null hay không? Điều tệ hại là MATHANG và NHACC cùng tạo ra một khoá cho quan hệ đó, và có lẽ không thể tìm ra các bộ nhờ chỉ mục sơ cấp được nếu có những giá trị null trong trường khoá MATHANG.
4. Bất thường khi xoá (deletion anomaly). Ngược lại với vấn đề (3) là vấn đề chúng ta có thể xoá tất cả các mặt hàng được cung cấp bởi một người, vô ý làm mất dấu vết để tìm ra địa chỉ của nhà cung cấp này.

Bạn đọc cần nhớ rằng vấn đề dư thừa và mâu thuẩn tiềm ẩn là những vấn đề chúng ta đã từng phân tích và giải quyết trong các mô hình khác. Trong mô hình mạng, các trường ảo được đưa ra nhằm mục đích loại bỏ các dư thừa và mâu thuẩn. Trong mô hình phân cấp, chúng ta đã dùng kiểu mẫu tinh ảo với mục đích tương tự. Mô hình đối tượng được tạo ra bằng các con trỏ chứ không phải bằng việc sao chép các đối tượng.

Trong thí dụ hiện tại, tất cả các vấn đề nêu trên sẽ biến mất khi chúng ta thay TT_CUNGCAP bằng hai lược đồ quan hệ sau:

NHACUNGCAP (NHACC, DIACHI)
XMHT , 123
XMSM , 456
CUNGCAP(NHACC, MATHANG, GIA)
XMHT , P400 ,50000
XMHT , P500 ,52000
XMSM , P400 ,49000

- Hình II.2 - quan hệ NHACUNGCAP và CUNGCAP -

Giống như trong hình II.1. Ở đây, quan hệ NHACUNGCAP cung cấp địa chỉ của mỗi nhà cung cấp đúng một lần; do vậy không có dư thừa. Ngoài ra chúng ta cũng có thể nhập địa chỉ của nhà cung cấp dù hiện tại họ không cung cấp một mặt hàng nào.

Tuy vậy vẫn còn một số câu hỏi còn bỏ ngõ. Chẳng hạn phân rã ở trên vẫn còn một khiếm khuyết: đó là để tìm địa chỉ của nhà cung cấp mặt hàng P400, bây giờ chúng ta phải thực hiện một phép nối có chi phí cao, còn với một quan hệ duy nhất TT_CUNGCAP chúng ta có thể dễ dàng trả lời bằng cách thực hiện một phép chọn rồi chiếu. Hoặc giả làm sao để xác định được rằng cách thay thế ở trên có nhiều ưu điểm hơn? Liệu có tồn tại trong hai lược đồ quan hệ mới những vấn đề thuộc bốn loại đã nêu ở trên không? Làm thế nào để có được một cách thay thế tốt cho một lược đồ quan hệ chưa tốt?

I.2. Phụ thuộc và dư thừa

Chúng ta còn nhấn mạnh đến mối quan hệ giữa các phụ thuộc và dư thừa. Tổng quát, một phụ thuộc là một khẳng định rằng chỉ một tập con của các quan hệ khả hữu là "hợp lệ", nghĩa là chỉ có một số quan hệ phản ảnh đúng tình trạng khả hữu của thế giới thực. Nếu không phải tất cả các quan hệ đều hợp lý, chúng ta có thể suy ra rằng tồn tại một số loại dư thừa trong những quan hệ hợp lệ. Nghĩa là cho trước khẳng định R là một quan hệ hợp lệ, nghĩa là thoả một số phụ thuộc nào đó, và cho trước các thông tin về gía trị hiện tại của R, chúng ta sẽ có thể suy ra được một số thông tin khác về giá trị hiện tại của R.

Trong trường hợp phụ thuộc hàm, hình thái của dư thừa khá rõ ràng. Trong quan hệ TT_CUNGCAP nếu chúng ta gặp hai bộ:

TT_CUNGCAP(NHACC,	DIACHI,	MATHANG,	GIA)
XMHT	, 123	, P400	,50000
XMHT	, ???	, P500	,52000

Chúng ta có thể cho rằng DIACHI phụ thuộc hàm vào NHACC và suy ra rằng ba chấm hỏi ??? biểu thị cho chuỗi « 123 ». Vì vậy đối với nhà cung cấp đã cho, phụ thuộc hàm làm cho tất cả các giá trị của DIACHI trở nên dư thừa ngoại trừ giá trị của DIACHI ở hàng đầu tiên : ta biết được điều đó mà không cần phải thấy được nó. Ngược lại, nếu chúng ta không cho rằng phụ thuộc hàm của DIACHI vào NHACC là đúng, thế thì không có lý do gì để tin rằng ??? biểu thị một giá trị cụ thể nào, và trường hợp đó sẽ không phải là dư thừa.

Khi phân tích một số loại phụ thuộc tổng quát hơn, hình thái của dư thừa không rõ ràng như trong phụ thuộc hàm. Tuy nhiên trong tất cả mọi trường hợp, dường như là căn nguyên và cách điều trị dư thừa luôn đi song hành. Nghĩa là phụ thuộc không chỉ gây ra dư thừa, chẳng hạn như phụ thuộc của DIACHI vào NHACC, nhưng nó cũng cho phép phân rã quan hệ TT_CUNGCAP thành các quan hệ NHACC và CUNGCAP bằng một cách để có thể khôi phục lại quan hệ gốc TT_CUNGCAP từ các quan hệ NHACC và CUNGCAP. Chúng ta sẽ thảo luận những vấn đề này một cách đầy đủ hơn trong phần sau.

I.3. Phụ thuộc hàm là gì

Gọi R $(A_1,...,A_n)$ là một lược đồ quan hệ,X,Y là các tập con của $\{A_1,...,A_n\}$. Ta nói $X \rightarrow Y$, đọc là « X xác định Y theo kiểu hàm » hoặc « Y phụ thuộc hàm vào X » nếu, với bất kỳ mọi quan hệ r nào đó là giá trị hiện hành (thể hiện) của R, không thể tồn tại hai bộ giống nhau ở các thành phần cho tất cả các thuộc tính trong tập X nhưng lại khác nhau ở một hay nhiều thành phần cho các thuộc tính trong tập Y. Vì vậy phụ thuộc hàm của thuộc tính địa chỉ nhà cung cấp (DIACHI) vào tên nhà cung cấp (NHACC) có thể đựoc diễn tả bằng

(NHACC) $\rightarrow$ (DIACHI)

I.4. Quy ước về ký hiệu

Để nhắc bạn đọc về ý nghĩa các ký hiệu được sử dụng, chúng ta thừa nhận các quy ước dưới đây :

1. Các chữ hoa ở đầu bộ chữ cái A,B, C,....biểu thị một thuộc tính đơn.
2. Các chữ hoa ở cuối bộ chữ cái U, V, ...Z, biểu thị cho tập các thuộc tính, có thể là tập chỉ một thuộc tính.
3. R đựoc dùng để biểu thị một lược đồ quan hệ. Chúng ta cùng đặt tên các quan hệ bằng lược đồ của chúng, chẳng hạn một quan hệ có các thuộc tính A,B,C có thể được viết là ABC.
4. Chúng ta sử dụng r cho một quan hệ, là *thể hiện hiện hành* của lược đồ R.
5. Ký hiệu nối kết chuỗi được dùng biểu thị phép hợp. Do vậy, $A_{1....}A_n$ được dùng để biểu diễn tập các thuộc tính $\{A_{1....}A_n\}$ và XY viết tắt của $X \cup Y$. trường hợp XA hay AX cũng được viết thay cho $X \cup \{A\}$, với X là tập các thuộc tính và A là một thuộc tính đơn.

II. PHỤ THUỘC HÀM

II.1. Ý nghĩa của phụ thuộc hàm

Các phụ thuộc hàm nảy sinh tự nhiên trong nhiều tình huống. Chẳng hạn nếu R biểu diễn cho một thực thể có các thuộc tính là $A_{1....}A_n$ và X là tập các thuộc tính tạo ra một khoá cho tập thực thể này thì chúng ta có thể khẳng định rằng $X \rightarrow Y$ với mọi tập con thuộc tính Y, kể cả khi tập Y có các thuộc tính chung với X. Lý do là các bộ phận của mỗi quan hệ khả hữu r biểu diễn các thực thể, và các thực thể được nhận dạng bằng giá trị của các thuộc tính khoá. Vì vậy hai bộ có giá trị giống nhau ở các thuộc tính trong X phải biểu diễn cho cùng một thực thể và do đó chỉ là một bộ.

Cũng cần nhấn mạnh rằng phụ thuộc hàm là những khẳng định về tất cả quan hệ khả hữu của lược đồ quan hệ R. chúng ta không xem xét một quan hệ r cụ thể của lược đồ R nhằm suy ra các phụ thuộc đúng trong R. Thí dụ, nếu r là một tập rỗng thì tất cả các phụ thuộc đều đúng, nhưng nó không đúng trong trường hợp tổng quát, khi gía trị của quan hệ biểu thị bởi R thay đổi. Tuy nhiên, chúng ta có thể xem xét một quan hệ cụ thể của R để khám phá ra một sự phụ thuộc không đúng.

Cách duy nhất để xác định đúng các phụ thuộc thích hợp cho một lược đồ R là xem xét cẩn thận xem các thuộc tính mang ý nghĩa gì. Theo cách này, các phụ thuộc thực sự là những khẳng định về thế giới thực, dù không thể chứng minh được, nhưng chúng ta hy vọng rằng chúng sẽ được quản lý, kiểm tra bởi DBMS nếu các nhà thiết kế CSDL đưa ra yêu cầu cho DBMS.

Trong quan hệ TT_CUNGCAP (hình II.1) có các phụ thuộc hàm sau:

NHACC →DIACHI

và

NHACC, MATHANG→GIA

Chúng ta có thể nhận xét rằng có nhiều phụ thuộc tầm thường như:

NHACC →NHACC

Và một số ít tầm thường hơn như

NHACC, MATHANG→DIACHI, GIA

Lý do khiến chúng ta tin rằng các phụ thuộc này là hợp lý vì nếu cho trước tên nhà cung cấp và mặt hàng, chúng ta có thể xác định một địa chỉ duy nhất; chúng ta bỏ qua mặt hàng và lấy địa chỉ này, chúng ta cũng có thể xác định được một giá duy nhất, là giá bán sỉ mặt hàng của nhà cung cấp này.

Tuy nhiên bạn đọc nên biết rằng phụ thuộc ở trên, không giống như các phụ thuộc khác chúng ta đã nói trong thí dụ này là, nó không đi kèm với một quan hệ cụ thể nào, chúng được nhận ra do chúng ta hiểu rõ ngữ nghĩa của “nhà cung cấp”, “mặt hàng”, “địa chỉ”, và “giá”. Chúng ta hy vọng rằng phụ thuộc này sẽ có ảnh hưởng trong các quan hệ có liên qua đến các thuộc tính đó, nhưng bản chất của ảnh hưởng này thường rất mơ hồ.

Chúng ta cũng có thể tự hỏi rằng không biết một phụ thuộc như

DIACHI→ NHACC

Có đúng hay không? Xem xét dữ liệu mẫu của hình II.1 chúng ta không tìm thấy có hai bộ nào giống nhau ở địa chỉ lại không giống nhau ở tên, chỉ đơn giản vì không có hai bộ nào có cùng địa chỉ. Tuy nhiên về nguyên tắc, không có gì để loại trừ khả năng hai nhà cung cấp có cùng địa chỉ, vì vậy chúng ta không dám khẳng định phụ thuộc này đúng, dù rằng nó dường như hợp lý trong quan hệ mẫu chúng ta vừa xem.

II.2. Một số định nghĩa

II.2.1. Phụ thuộc hàm :

Cho lược đồ quan hệ xác định trên tập các thuộc tính R $= \{A_1, A_2, .., A_n\}$. *Một phụ thuộc hàm trên R là công thức có dạng :*

f: $X \rightarrow Y$; *với* $X, Y \subseteq R$

Nếu f: X→Y là một phụ thuộc hàm trên R thì ta nói tập thuộc tính Y *phụ thuộc* vào tập thuộc tính X, hoặc tập thuộc tính X *xác định hàm* tập thuộc tính Y.

II.2.2. Thể hiện thỏa phụ thuộc :

Với một lược đồ R có thể có nhiều thể hiện (*quan hệ*) r khác nhau. Đó là một tập gồm các bộ là *tình trạng* của dữ liệu tại một thời điểm nào đó.

Chúng ta nói rằng một quan hệ r thoả phụ thuộc hàm X→Y nếu với mỗi hai bộ μ và v trong r sao cho: nếu μ[X] = v[X] thì μ[Y] = v[Y].

Chú ý rằng giống như mọi câu lệnh "if….then", khẳng định trên có thể được thoả bởi μ[X] khác với v[X] hoặc bởi μ[Y] giống với v[Y].

Nếu r không thoả phụ thuộc X→Y thì ta nói r *vi phạm* phụ thuộc đó.

Cho F là một tập các phụ thuộc hàm, ta nói quan hệ r thỏa F, nghĩa là r thỏa tất cả các phụ thuộc hàm trong F.

II.2.3. Phụ thuộc hàm định nghĩa trên lược đồ :

Nếu với *mọi quan hệ r* của lược đồ R, *r thoả X→Y* thì ta nói X→Y được *định nghĩa* trên lược đồ R.

Giả sử chúng ta khai báo rằng X→Y được *định nghĩa* trên lược đồ R thì chúng ta biết rằng mọi quan hệ r của lược đồ R sẽ thoả X→Y. Tuy nhiên nếu X→Y không được *định nghĩa* trên lược đồ R thì một quan hệ r nào đó vẫn có thể ngẫu nhiên thoả X→Y hay có thể vi phạm X→Y.

Cho F là một tập các phụ thuộc hàm, ta nói F được định nghĩa trên lược đồ R, nghĩa là tất cả các phụ thuộc hàm trong F được định nghĩa trên R.

II.3. Phụ thuộc hàm hệ quả

Giả sử R là một lược đồ quan hệ và A, B, C là một số thuộc tính của nó. Cũng giả sử rằng các phụ thuộc hàm A→B và B→C được định nghĩa trong R. Chúng ta mong muốn rằng phụ thuộc hàm A→C cũng được định nghĩa trong R. Thật vậy, giả sử r là một quan hệ thoả A→B và B→C , nhưng có hai bộ μ và v trong r giống nhau ở thành phần A nhưng không giống nhau ở thành phần C. Thế thì chúng ta phải đặt câu hỏi liệu μ và v có giống nhau ở thuộc tính B hay không. Nếu không, r vi phạm phụ thuộc hàm A→B. Nếu có, thì bởi vì chúng giống nhau ở B nhưng không giống nhau ở thành phần C, nên r vi phạm B→C . Do vậy, r buộc phải thoả A→C.

Tổng quát, gọi F là tập các phụ thuộc hàm cho lược đồ quan hệ R và gọi X→Y là một phụ thuộc hàm. Ta nói X→Y là hệ quả của F, và viết là F $\models$ X→Y, nếu mỗi quan hệ r của R thoả các phụ thuộc trong F thì cũng thoả X→Y.

Ở trên chúng ta thấy rằng nếu F chứa A→B và B→C thì A→C là hệ quả của F. Nghĩa là

$\{A \rightarrow B, B \rightarrow C \models A \rightarrow C\}$

II.4. Bao đóng của các tập phụ thuộc

Chúng ta định nghĩa F^+, bao đóng (closure), là tập các phụ thuộc hàm hệ quả của F, nghĩa là

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

Thí dụ II.1:

Gọi R=ABC và F= {A $\rightarrow$ B, B $\rightarrow$ C} thế thì bao đóng F^+ chứa tất cả các phụ thuộc dạng X$\rightarrow$Y sao cho :

1. Hoặc X chứa A, chẳng hạn, ABC $\rightarrow$ AB,AB $\rightarrow$ BC, hay A $\rightarrow$ C.
2. Hoặc X chứa B nhưng không chứa A, và Y không chứa A, chẳng hạn BC $\rightarrow$ B, B $\rightarrow$ C hay B $\rightarrow \varnothing$,
3. Hoặc X $\rightarrow$ Y là mọt trong ba phụ thuộc C $\rightarrow$ C, C $\rightarrow$ $\varnothing$ hay $\varnothing \rightarrow \varnothing$.

Lưu ý : $\forall F,\ F \subseteq F^+$

II.5. Phụ thuộc hàm suy dẫn

Ta gọi f là một phụ thuộc hàm được *suy dẫn* từ tập các phụ thuộc hàm cho trước F nhờ vào một tập các *luật dẫn*, ký hiệu F $\vdash$ f , nếu tồn tại một chuỗi các phụ thuộc hàm :

$f_1, f_2, \ldots, f_n$ sao cho $f_{n\,=}\,f$ và mỗi f_i là một phụ thuộc hàm trong F hay được suy dẫn từ những phụ thuộc hàm j = 1,2, …,i-1 trước đó nhờ vào luật dẫn.

Ký hiệu F^*, tập các phụ thuộc hàm được suy dẫn từ F nhờ vào các luật dẫn.

Điều ta mong muốn là $F^+ = F^*$

Nói một cách khác, tập các luật dẫn là *đúng đắn (hợp lệ)* và *đầy đủ :*

- Tập các luật dẫn là đúng đắn nếu và chỉ nếu :

$F^* \subseteq F^+$, nghĩa là : $\forall$f, nếu F $\vdash$ f thì F $\models$ f

Tức là: nếu ta *dùng các luật dẫn* để suy dẫn ra một phụ thuộc f nào đó từ tập các phụ thuộc hàm F cho trước, thì f cũng là phụ thuộc hàm *hệ quả* của F.

- Tập các luật dẫn là đầy đủ nếu và chỉ nếu :

$F^+ \subseteq F^*$, nghĩa là : $\forall$f, nếu F $\models$ f thì F $\vdash$ f

Tức là: nếu f cũng là phụ thuộc hàm *hệ quả* của F, thì ta có thể *dùng các luật dẫn* để suy dẫn ra f từ tập các phụ thuộc hàm F.

II.6. Các tiên đề cho phụ thuộc hàm

Để xác định khoá và hiểu được các phép suy dẫn cho các phụ thuộc hàm nói chung, chúng ta cần tính được F^+ từ F, hoặc ít nhất khẳng định được $X \rightarrow Y$ có thuộc F^+ hay không nếu biết tập phụ thuộc hàm F và phụ thuộc hàm $X \rightarrow Y$. Muốn vậy, chúng ta phải có những qui tắc suy dẫn cho biết làm sao có thể suy ra một hay nhiều phụ thuộc từ các phụ thuộc khác. Thực tế chúng ta còn có nhiều hơn thế: chúng ta có một tập các quy tắc suy dẫn *đầy đủ*, theo nghĩa là từ một tập các phụ thuộc F đã biết, những qui tắc này cho phép chúng ta suy ra được tất cả các phụ thuộc thực sự, nghĩa là những phụ thuộc trong F^+. Hơn nữa, những qui tắc này là *đúng đắn*, theo nghĩa là khi sử dụng chúng, chúng ta không thể suy ra từ F một phụ thuộc sai, nghĩa là một phụ thuộc không thuộc F^+.

Tập các qui tắc (luật dẫn) này thường được gọi là hệ tiên đề Armstrong (Armstrong's axioms), do Armstrong đưa ra lần đầu vào năm 1874.

Trong những phần dưới đây, giả sử rằng chúng ta đã có một lược đồ quan hệ với tập các thuộc tính U, là tập gồm đầy đủ các thuộc tính của R, và tập các phụ thuộc hàm F chỉ chứa các thuộc tính trong U.

Chúng ta có các qui tắc suy dẫn sau:

A1: Tính phản xạ (Relexivity)

Nếu $Y \subseteq X \subseteq U$, thì $X \rightarrow Y$.

Qui tắc này đưa ra những phụ thuộc tầm thường (trivial dependency) là những phụ thuộc mà vế trái được hàm chứa trong vế phải. Những phụ thuộc tầm thường đều đúng trong mọi quan hệ, chúng nói lên rằng việc sử dụng qui tắc này chỉ phụ thuộc vào U, không phải vào F.

A2: Tính tăng trưởng (Augmentation).

Nếu $X \rightarrow Y$, và $Z \subseteq U$ thì $XZ \rightarrow YZ$

Để ý rằng X,Y và Z là các tập thuộc tính, và XZ là dạng viết tắt của $X \cup Z$. Một điều quan trọng khác cũng cần phải nhớ rằng phụ thuộc $X \rightarrow Y$ đã cho có thể thuộc F hoặc nó có thể được suy dẫn từ các phụ thuộc trong F bằng cách sử dụng các tiên đề đang mô tả ở đây.

A3: Tính bắc cầu (Transitivity)

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

Thí dụ II.2:

Xét lược đồ quan hệ ABCD với các phụ thuộc hàm $A \rightarrow C$ và $B \rightarrow D$. Chúng ta có thể chứng minh rằng AB là khoá của ABCD (sự thực nó là khoá duy nhất). Chúng ta có thể chứng minh rằng AB là một khoá bao hàm bằng các bước sau đây:

1. $A \rightarrow C$ (đã cho)
2. $AB \rightarrow ABC$ [tính tăng trưởng của (1) từ AB]
3. $B \rightarrow D$ (đã cho)
4. $ABC \rightarrow ABCD$ [tính tăng trưởng của (3) từ ABC]
5. $AB \rightarrow ABCD$ [tính bắc cầu được áp dụng cho (2) và (4)]

Để chứng minh AB là một khoá, chúng ta cũng phải chứng minh rằng bản thân mỗi thuộc tính A và B không thể suy diễn logic ra tất cả các thuộc tính. Chúng ta có thể chứng minh A không phải là khoá bao hàm bằng cách đưa ra một quan hệ thoả các phụ thuộc đã biết (1) và (3) ở trên nhưng không thoả $A \rightarrow ABCD$ và có thể tiến hành tương tự với B. Tuy nhiên chúng ta cũng sẽ xây dựng một thuật toán nhằm kiểm tra xem một tập thuộc tính có phải là khoá hay không, vì thế chúng ta tạm bỏ qua bước này.

II.7. Tính đúng đắn của hệ tiên đề Armstrong

Tương đối dễ chứng minh rằng hệ tiên đề Armstrong là đúng đắn; nghĩa là chúng chỉ dẫn đến những kết luận đúng. Còn tính đầy đủ khó chứng minh hơn. Chúng có thể được dùng để thực hiện các phép suy diễn có giá trị về các phụ thuộc. Trước tiên chúng ta chứng minh tính đúng đắn của hệ tiên đề Armstrong.

II.7.1. Bổ đề II.1:

Hệ tiên đề Armstrong là đúng. Nghĩa là nếu $X \rightarrow Y$ được suy dẫn ra từ F nhờ hệ tiên đề này thì phụ thuộc $X \rightarrow Y$ đúng trong mọi quan hệ mà các phụ thuộc của F đúng. Nghĩa là nếu r là một quan hệ thỏa F thì r cũng thỏa $X \rightarrow Y$.

Chứng minh:

A1, tiên đề về tính phản xạ rõ ràng là đúng đắn. chúng ta không thể có một quan hệ r với hai bộ giống nhau ở các thành phần trong X nhưng lại không giống nhau ở một tập con nào đó của X.

A2, tính tăng trưởng, giả sử rằng chúng ta có quan hệ r thoả $X \rightarrow Y$, thế thì có hai bộ μ và v giống nhau ở các thuộc tính của XZ nhưng khác nhau ở các thuộc tính của YZ. Bởi vì chúng không thể khác nhau ở các thuộc tính của Z, μ và v phải khác nhau ở một số thuộc tính của Y. Nhưng vì μ và v giống nhau ở phần X nhưng không giống nhau ở phần Y nên mâu thuẫn với giả thiết rằng $X \rightarrow Y$ đúng đối với r.

A3, tiên đề về tính bắc cầu, là sự mở rộng của lập luận trước đây về $A \rightarrow B$ và $B \rightarrow C$ suy ra $A \rightarrow C$. Chúng tôi để phần này lại cho bạn đọc tự chứng minh.

II.8. Các qui tắc suy dẫn bổ sung

Có nhiều qui tắc được suy ra từ hệ tiên đề Arsmtrong. Trong bổ đề kế tiếp, chúng ta trình bày ba trong số các qui tắc này. Bởi vì chúng ta đã chứng minh tính đúng đắn của A1,A2,và A3 nên chúng ta được quyền sử dụng chúng trong các phép chứng minh sau.

II.8.1. Bổ đề II.2 :

a) Qui tắc hợp (The union rule)
Nếu $X \rightarrow Y$ và $X \rightarrow Z$ thì $X \rightarrow YZ$.

b) Qui tắc giả bắc cầu (The pseudotransitivity rule)
Nếu $X \rightarrow Y$ và $WY \rightarrow Z$ thì $WX \rightarrow Z$.

c) Qui tắc phân rã (The decomposition rule)
Nếu $X \rightarrow Y$ và $Z \subseteq Y$ thì $X \rightarrow Z$.

Chứng minh :

a) Chúng ta đã có $X \rightarrow Y$ nên dùng qui tắc tăng trưởng với X chúng ta có $X \rightarrow XY$. Chúng ta cũng có $X \rightarrow Z$ nên suy ra $XY \rightarrow YZ$. Nhờ tính chất bắc cầu, từ $X \rightarrow XY$ và $XY \rightarrow YZ$ suy ra $X \rightarrow YZ$.
b) Sử dụng tính tăng trưởng $X \rightarrow Y$ thành $WX \rightarrow WY$. Bởi vì chúng ta đã có $WY \rightarrow Z$, tính bắc cầu cho phép suy ra $WX \rightarrow Z$.
c) Từ tính phản xạ ta có $Y \rightarrow Z$, nên nhờ tính bắc cầu ta có $X \rightarrow Z$. □

Một hệ quả quan trọng của các qui tắc hợp và phân rã là nếu $A_1,\ldots A_n$ là các thuộc tính thì $X \rightarrow A_1,\ldots A_n$ đúng nếu và chỉ nếu $X \rightarrow A_1$ đúng với mọi i. Vì vậy chúng ta chỉ cần sử dụng các phụ thuộc mà vế phải chỉ có một thuộc tính duy nhất. Chúng ta sẽ thảo luận vấn đề này chi tiết hơn khi phân tích về "phủ cực tiểu" của các phụ thuộc hàm.

II.9. Bao đóng của tập thuộc tính

Trước khi chứng minh tính đầy đủ, chúng ta cần định nghĩa bao đóng (closure) của một tập các thuộc tính ứng với một tập phụ thuộc hàm.

Gọi F là tập phụ thuộc hàm trên tập thuộc tính U. Cho X là một tập con của U. Ta gọi bao đóng của X (ứng với F), ký hiệu X^+_F (hay X^+), là tập các thuộc tính A sao cho $X \rightarrow A$ có thể suy dẫn ra từ F nhờ hệ tiên đề Arsmtrong.

$X^+_F = \{ A / X \rightarrow A \in F^* \}$

Điểm cốt lõi của bao đóng của tập thuộc tính là nó cho phép chúng ta khẳng định rằng một phụ thuộc $X \rightarrow Y$ có thể suy ra từ F bằng hệ tiên đề Arsmtrong được hay không . Bổ đề sau đây khẳng định điều này.

II.9.1. Bổ đề II.3 :

$X \rightarrow Y$ suy ra được từ một tập phụ thuộc F đã cho bằng cách sử dụng hệ tiên đề Arsmtrong nếu và chỉ nếu $Y \subseteq X^+$ (ở đây bao đóng của X được lấy ứng với F).

Chứng minh:

Đặt $Y=A_1,\ldots A_n$ cho tập các thuộc tính $A_1,\ldots A_n$ và giả sử $Y \subseteq X^+$; . Theo định nghĩa của X^+, $X \rightarrow A_i$, được suy ra từ hệ tiên đề Arsmtrong với mọi i. Bằng qui tắc hợp ta có $X \rightarrow Y$ đúng.

Ngược lại, giả sử rằng $X \rightarrow Y$ được suy ra từ hệ tiên đề này. Đối với mỗi i, $X \rightarrow A_i$ đúng theo qui tắc phân rã, vì vậy $Y \subseteq X^+$. □

II.10. Tính đầy đủ của hệ tiên đề Armstrong

Bây giờ chúng ta đã sẵn sàng để có thể chứng minh tính đầy đủ (completeness) của hệ tiên đề Arsmtrong. Chúng ta sẽ chứng minh rằng nếu F là tập các phụ thuộc đã cho, và không thể suy ra $X \rightarrow Y$ đúng bằng hệ tiên đề Arsmtrong thì phải có một quan hệ trong đó tất cả các phụ thuộc của F đều đúng nhưng $X \rightarrow Y$ không đúng; nghĩa là không khẳng định logic $X \rightarrow Y$.

II.10.1. Định lý II.1 : hệ tiên đề Arsmtrong là đúng đắn và đầy đủ.

Chứng minh:

Tính đúng đắn đã được chứng minh qua Bổ đề II.1, vì vậy chúng ta chỉ còn phải chứng minh tính đầy đủ.

Gọi F là tập các tập các phụ thuộc trên tập thuộc tính U và giả sử $X \rightarrow Y$ không thể suy ra được từ hệ tiên đề này. Xét một quan hệ r có hai bộ được trình bày trong Hình II.3. Trước tiên chúng ta chứng minh rằng tất cả các phụ thuộc trong F đều được thoả bởi r. Về trực quan, một phụ thuộc $V \rightarrow W$ bị r phạm cho phép chúng ta "đẩy" X^+ vượt quá giá trị mà nó có khi được cho tập các phụ thuộc F.

Giả sử $V \rightarrow W$ thuộc F nhưng không được thoả bởi r. Thế thì $V \subseteq X^+$, nếu không thì hai bộ của r không giống nhau ở các thuộc tính của V, vì vậy không vi phạm $V \rightarrow W$. Cũng vậy W không thể là một tập con của X^+. Bởi vì $V \subseteq X^+$, nên $X \rightarrow V$ được suy ra từ hệ tiên đề Armstrong nhờ Bổ đề II.3. Phụ thuộc $V \rightarrow W$ thuộc F, vì thế do tính bắc cầu chúng ta có $X \rightarrow W$. Do tính phản xạ, $W \rightarrow A$, vậy do tính bắc cầu $X \rightarrow A$ được suy ra từ các tiên đề này. Nhưng nếu vậy thì theo định nghĩa của bao đóng. A thuộc X^+, mâu thuẫn với giả thiết. Do vậy có thể kết luận rằng r thoả mọi phụ thuộc $V \rightarrow W$ trong F.

Các thuộc tính của X^+			Các thuộc tính khác		
1	1	1….. 1	1	1	1….. 1
1	1	1….. 1	0	0	0 …. 0

- Hình II.3 - một quan hệ chứng minh F không khẳng định logic $X \rightarrow Y$ -

Bây giờ chúng ta chứng minh rằng $X \rightarrow Y$ không được thoả bởi r. Giả sử rằng nó được thoả. Hiển nhiên là $X \subseteq X^+$ nên suy ra rằng $Y \subseteq X^+$, nếu không hai bộ của r sẽ có các giá trị trong X giống nhau nhưng các giá trị trong Y lại không giống nhau. Thế nhưng bổ đề II.3 lại khẳng định rằng $X \rightarrow Y$ có thể được suy ra từ các tiêu đề, là điều trái với giả thiết là $X \rightarrow Y$ không thể suy ra từ các tiên đề. Vì thế $X \rightarrow Y$ không được thoả bởi r, mặc dù mỗi phụ thuộc F đều thoả. Chúng ta kết luận rằng khi $X \rightarrow Y$ không suy ra được từ tập phụ thuộc F bằng hệ tiên đề Armstrong, $X \rightarrow Y$ cũng không là *hệ quả* của F. Điều này khẳng định rằng hệ tiên đề này là đầy đủ. □

Định lý II.1 có một số hệ quả sau:

Như đã đề cập trong phần II.5 và kết quả của định lý II.1 ta có $F^+ = F^*$.

Chúng ta đã định nghĩa X^+ là tập các thuộc tính sao cho $X \rightarrow A$ được *suy dẫn* từ tập các phụ thuộc F đã cho bằng cách dùng các tiên đề này. Bây giờ chúng ta có một định nghĩa tương đương cho X^+, đó là một tập các thuộc tính A sao cho $X \rightarrow A$ là *hệ quả* của F.

$$X^+_F = \{ A / X \rightarrow A \in F^* \}$$
$$= \{ A / X \rightarrow A \in F^+ \}$$

Một hệ quả khác là mặc dù chúng ta định nghĩa F^+ là tập các phụ thuộc là *hệ quả* của F, chúng ta có thể xem F^+ là tập các phụ thuộc được *suy dẫn* từ F bằng hệ tiên đề Armstrong.

$$F^+ = \{X \to Y \;/\; F \models X \to Y\}$$
$$= \{X \to Y \;/\; F \vdash X \to Y\}$$

II.10.2. Bài toán thành viên

Cho trước tập phụ thuộc hàm F và f : X→Y là một phụ thuộc hàm mới được nhận diện. Bài tóan đặt ra là f có phải là *thành viên* của F, nghĩa là f có thuộc bao đóng của F không ?

f là thành viên của F
⇔ $f \in F^+$
⇔ $f \in F^*$ *do* $F^+ = F^*$ *(định lý II.1)*
⇔ $Y \subseteq X^+$ *do bổ đề II.3*

Cho nên để giải quyết bài tóan thành viên, ta chỉ cần tính X^+ đối với F, sau đó xét xem $Y \subseteq X^+$?

II.11. Tính các bao đóng

Rõ ràng là việc tính F^+ cho một tập phụ thuộc F nói chung tốn rất nhiều thời gian, đơn giản là vì tập các phụ thuộc của F^+ có thể rất lớn dù rằng tập phụ thuộc F nhỏ. Chẳng hạn xét tập phụ thuộc:

$F = (A \to B_1, A \to B_2, ..., A \to B_n)$

Thế thì F^+ bao gồm tất cả các phụ thuộc $A \to Y$, trong đó Y là một tập con của ($B_1, B_2, ..., B_n$). Bởi vì có có đến 2^n tập Y như thế, chúng ta không hy vọng liệt kê hết được tập F^+ ngay cả với kích thước n vừa phải.

Ngược lại, việc tính X^+ cho tập các thuộc tính X thì không khó; chi phí thuật toán này tỷ lệ với chiều dài của tất cả các phụ thuộc trong F. Nhờ Bổ đề II.3 và định lý II.1 về tính đúng đắn và đầy đủ của hệ tiên đề Armstrong, chúng ta có thể biết được $X \to Y$ có thuộc F^+ hay không bằng cách tính X^+ ứng với F. Cách tính X^+ đơn giản như sau.

II.11.1. Thuật toán II.1: tính các bao đóng một tập thuộc tính ứng với môt tập phụ thuộc hàm.

NHẬP:

Tập thuộc tính hữu hạn U, tập phụ thuộc hàm F trên U, và tập $X \subseteq U$.

XUẤT:

X^+, bao đóng của X ứng với F.

PHƯƠNG PHÁP:

chúng ta tính chuỗi các tập thuộc tính $X^{(0)}$, $X^{(1)}$, …. bằng các qui tắc:

1. $X^{(0)}$ chính là X.
2. $X^{(i+1)}$ là hợp của $X^{(i)}$ với tập các thuộc tính A sao cho có một phụ thuộc $Y \rightarrow Z$ thuộc F, A thuộc Z và $Y \subseteq X^{(i)}$.

Bởi vì $X = X^{(0)} \subseteq X^{(1)} \subseteq \ldots \subseteq X^{(i)} \subseteq U$, và U là hữu hạn, cuối cùng chúng ta phải đạt đến một trị số i sao cho $X^{(i)} = X^{(i+1)}$. Bởi vì mỗi $X^{(j+1)}$ chỉ được tính theo $X^{(j)}$, suy ra rằng $X^{(i)} = X^{(i+1)} = X^{(i+2)}$ …Chúng ta không cần phải tính tiếp khi đã phát hiện ra rằng$X^{(i)} = X^{(i+1)}$, và có thể chứng minh rằng X^+ chính là $X^{(i)}$ với trị số i này. □

Thí dụ II.3:

Gọi F là tập gồm 8 phụ thuộc sau:

$AB \rightarrow C$	$D \rightarrow EG$
$C \rightarrow A$	$BE \rightarrow C$
$BC \rightarrow D$	$CG \rightarrow BD$
$ACD \rightarrow B$	$CE \rightarrow AG$

Và cho X= BD. Hãy tính X^+

Để áp dụng Thuật toán II.1, khởi đầu chúng ta đặt $X^{(0)}$ = BD.

Muốn tính $X^{(1)}$, chúng ta tìm các phụ thuộc có vế phải là B,D hoặc BD. Chỉ có một phụ thuộc $D \rightarrow EG$, vì thế chúng ta nối E và G với $X^{(1)}$, kết quả là $X^{(1)}$ = BDEG.

Đối với $X^{(2)}$ chúng ta tìm các vế trái được chứa trong $X^{(1)}$ và tìm được hai phụ thuộc $D \rightarrow EG$ và $BE \rightarrow C$. Vì vậy $X^{(2)}$ =BCDEG.

Tiếp tục với $X^{(3)}$, chúng ta tìm các vế trái được chứa trong BCDEG và tìm ra các phụ thuộc mới $C \rightarrow A$, $BC \rightarrow D$, $CG \rightarrow BD$, và $CE \rightarrow AG$. Vì vậy $X^{(3)}$ = ABCDEG, là tập hợp gồm tất cả mọi thuộc tính đã cho.

Do đó không có gì ngạc nhiên là $X^{(3)} = X^{(4)} = \ldots$. Do vậy $(BD)^+$ = ABCDEG. □

Bây giờ chúng ta phải chứng minh tính đúng đắn của Thuật toán II.1. Chúng ta có thể dễ dàng chứng minh được rằng mỗi thuộc tính được đặt vào một tập $X^{(j)}$ đều thuộc X^+ , nhưng chứng minh rằng mỗi thuộc tính trong X^+ đều được đặt trong một $X^{(j)}$ nào đó thì khó hơn.

II.11.2. Định lý II.2: Thuật toán II.1 tính đúng X^+.

Chứng minh:

Trước tiên chúng ta chứng minh bằng qui nạp trên j rằng nếu A được đặt trong $X^{(j)}$ trong khi "chạy" thuật toán II.1 thì A thuộc X^+; nghĩa là nếu A thuộc tập $X^{(j)}$ được trả về bởi thuật toán II.1 thì A thuộc X^+.

Bước cơ sở: j = 0. Thế thì A thuộc X, vì vậy do tính phản xạ, $X \rightarrow A$.

Qui nạp: Với $j > 0$ và giả sử rằng $X^{(j-1)}$ chỉ chứa các thuộc tính X^+. Giả sử A được đặt trong $X^{(j)}$ do A thuộc Z, $Y \rightarrow Z$ thuộc F, và $Y \subseteq X^{(j-1)}$. Bởi vì $Y \subseteq X^{(j-1)}$ chúng ta biết rằng $Y \subseteq X^+$ theo giả thiết qui nạp . Vì vậy $X \rightarrow Y$ theo Bổ đề II.3. Nhờ tính chất bắc cầu, $X \rightarrow Y$ và $Y \rightarrow Z$ suy ra $X \rightarrow Z$. Nhờ tính chất phản xạ, $Z \rightarrow A$ nên $X \rightarrow A$ lại do tính bắc cầu. Vì vậy A thuộc X^+.

Bây giờ chúng ta chứng minh phần đảo: nếu A thuộc X^+ thì A là phần tử của tập được trả về bởi thuật toán II.1.

Giả sử A thuộc X^+ nhưng A không thuộc tập $X^{(j)}$ được trả về bởi Thuật toán II.1. Chú ý rằng $X^{(i)} = X^{(i+1)}$ bởi vì đó là điều kiện để Thuật toán II.1 kết thúc.

Xét quan hệ r tương tự như Hình II.3: r có hai bộ giống nhau ở các thuộc tính của $X^{(i)}$ nhưng khác nhau ở tất cả các thuộc tính khác. Chúng ta khẳng định rằng r thoả F. Thật vậy, gọi $U \rightarrow V$ là một phụ thuộc trong F bị vi phạm bởi r. Thế thì $U \subseteq X^{(i)}$ và V không thể là một tập con của $X^{(i)}$ nếu xảy ra vi phạm (lập luận tương tự như trong phần chứng minh Định lý II.1). Vì vậy không thể bằng với $X^{(i)}$ như đã giả định.

Do đó, quan hệ r cũng phải thoả $X \rightarrow A$. Lý do là A được giả thiết là thuộc X^+ , và vì thế $X \rightarrow A$ được suy ra từ F bởi hệ tiên đề Armstrong. Vì các tiên đề này là đúng đắn nên bất kỳ quan hệ nào thoả F cũng thoả $X \rightarrow A$. Nhưng cách duy nhất $X \rightarrow A$ có thể đúng trong r là A thuộc $X^{(i)}$ bởi vì nếu không thì hai bộ của r, chắc chắn rằng giống nhau ở X, sẽ không giống nhau ở A và vì vậy vi phạm $X \rightarrow A$. Chúng ta kết luận rằng A thuộc tập $X^{(i)}$ được trả về bởi Thuật toán II.1.

II.12. Tính tương đương của các tập phụ thuộc

Định nghĩa :

Gọi F và G là các tập phụ thuộc hàm. Ta nói F và G là tương đương, ký hiệu $F \equiv G$ *, nếu:* $F^+ = G^+$

Khi F và G *tương đương*, ta nói F *phủ* G, hay G *phủ* F

Bổ đề II.4:

F và G tương đương nếu và chỉ nếu G là tập con của F^+ *và F là tập con của* G^+

$F \equiv G \Leftrightarrow G \subseteq F^+$ *và* $F \subseteq G^+$

Chứng minh:

a. Điều kiện đủ: giả thiết F tương đương G

Lưu ý (i): với mọi tập phụ thuộc hàm G, ta có $G \subseteq G^+$

a.1. phải chứng minh $G \subseteq F^+$ hay $\forall g \in G \Longrightarrow g \in F^+$

Gọi g là một phụ thuộc hàm bất kỳ thuộc G, ta có g thuộc G^+ (do Lưu ý (i) nên $G \subseteq G^+$). Hơn nữa ta cũng có $G \subseteq F^+$ (do giả thiết F tương đương với G). Nên g cũng thuộc F^+ .

a.2. phải chứng minh $F \subseteq G^+$ hay $\forall f \in F \Longrightarrow f \in G^+$
Tương tự chứng minh trên.

b. Điều kiện cần: giả thiết $G \subseteq F^+$ và $F \subseteq G^+$
Phải chứng minh F tương đương G hay $F^+ = G^+$
Lưu ý (ii):với mọi tập phụ thuộc hàm G,ta có $(G^+)^+ = G^+$
Lưu ý (iii):với các tập phụ thuộc hàm F,G bất kỳ, ta có nếu $F \subseteq G$ thì $F^+ \subseteq G^+$

b.1. ta chứng minh $G^+ \subseteq F^+$
Do giả thiết $G \subseteq F^+$ và do lưu ý (ii) nên $G^+ \subseteq (F^+)^+$
Hơn nữa do lưu ý (iii) ta có $(F^+)^+ = (F^+)^+$ nên $G^+ \subseteq F^+$
b.2. ta chứng minh $F^+ \subseteq G^+$
Tương tự chứng minh trên.
b.3. do (b.1) và (b.2) ta có $F^+ = G^+$ □

Nhận xét :

Áp dụng kết quả của bổ đề II.4, chúng ta dễ dàng kiểm tra tính tương đương của F và G theo các bước sau:

Với mỗi phụ thuộc $Y \rightarrow Z$ thuộc F, kiểm tra xem $Y \rightarrow Z$ có thuộc G^+ hay không bằng cách dùng thuật toán II.1 để tính Y^+ ứng với G rồi kiểm chứng lại xem có phải $Z \subseteq Y^+$ hay không . Nếu một phụ thuộc $Y \rightarrow Z$ nào đó thuộc F nhưng không thuộc G^+ thì chắc chắn $F^+ \neq G^+$. Nếu mỗi phụ thuộc của F đều thuộc G^+ thì ta có được kết quả $F \subseteq G^+$.

Để chứng minh rằng mỗi phụ thuộc trong G cũng thuộc F^+ chúng ta sử dụng phương pháp tương tự và nếu được kết quả $G \subseteq F^+$.

Áp dụng bổ đề II.4 ta có $F \equiv G$.

II.13. Phủ cực tiểu

Đối với một tập phụ thuộc đã cho, chúng ta có thể tìm một tập tương đương có một số đặc tính hữu ích. Một đặc tính đơn giản và quan trọng là các vế phải của những phụ thuộc này được tách thành những thuộc tính độc nhất.

II.13.1. Bổ đề II.5

Mỗi tập phụ thuộc F tương đương với một tập phụ thuộc G trong đó các vế phải không có quá một thuộc tính.

Chứng minh:

Gọi G là tập phụ thuộc $X \rightarrow A$ sao cho với một phụ thuộc $X \rightarrow Y$ thuộc F, A thuộc Y. Thế thì $X \rightarrow A$ được suy ra từ $X \rightarrow Y$ bằng qui tắc phân rã. Do đó $G \subseteq F^+$.

Nhưng ta cũng có $F \subseteq G^+$ bởi vì nếu $Y = A_1 \ldots . A_n$ thì $X \rightarrow Y$ được suy ra từ $X \rightarrow A_1, \ldots, X \rightarrow A_n$ nhờ qui tắc hợp. Vì vậy F và G tương đương □

Rõ ràng chúng ta cần một lý thuyết thiết kế lược đồ CSDL đưa ra nhiều hạn chế hơn là chỉ yêu cầu vế phải chỉ có một thuộc tính.

II.13.2. Tập phụ thuộc cực tiểu

Một tập phụ thuộc F là cực tiểu nếu:

1. *Vế phải của mỗi phụ thuộc trong F chỉ có một thuộc tính.*
2. *Không tồn tại bất kỳ một phụ thuộc $X \rightarrow A$ nào trong F mà tập $F - \{X \rightarrow A\}$ tương đương với F.*
3. *Không tồn tại bất kỳ một phụ thuộc $X \rightarrow A$ nào trong F sao cho có một tập con thực sự Z của X mà $(F - \{X \rightarrow A\}) \cup \{Z \rightarrow A\}$ tương đương với F.*

Về trực quan, (2) bảo đảm rằng không có phụ thuộc nào trong F là dư thừa. Chúng ta có thể kiểm tra xem $X \rightarrow A$ có dư thừa hay không bằng cách tính X^+ ứng với F - $\{X \rightarrow A\}$ rồi so sánh kết quả với X^+ ứng với F.

Điều kiện (3) bảo đảm rằng không có thuộc tính nào ở vế trái là dư thừa. Chúng ta có thể kiểm tra các phụ thuộc dư thừa ở vế trái như sau:

Thuộc tính B trong X đối với phụ thuộc $X \rightarrow A$ là dư thừa nếu và chỉ nếu A thuộc $(X - \{B\})^+$ khi bao đóng được lấy ứng với F.

Bởi vì theo (1), mỗi vế phải chỉ có một thuộc tính nên chắc chắn rằng không có thuộc tính nào ở vế phải là dư thừa.

Thí dụ :

Tập phụ thuộc F = {AB → C; C → DE} không phải là tập phụ thuộc cực tiểu vì phụ thuộc hàm C → DE có vế phải gồm 2 thuộc tính (vi phạm điều kiện 1)

Tập phụ thuộc F = {AB → C; C → D; AB → D} thỏa điều kiện 1 nhưng vi phạm điều kiện 2 vì ta có thể bỏ phụ thuộc hàm AB → D.

II.13.3. Phủ cực tiểu

Cho trước tập phụ thuộc hàm F. Một tập phụ thuộc hàm G được gọi là phủ cực tiểu (minimal cover) của F nếu:

- *G là một tập phụ thuộc cực tiểu*
- *và G tương đương F (G phủ F).*

II.13.4. Định lý II.3

Mỗi tập phụ thuộc F đều có ít nhất một phủ cực tiểu.

Chứng minh:

Nhờ Bổ đề II.5, chúng ta có thể giả sử rằng các vế phải trong F đều có một thuộc tính. Chúng ta sẽ tìm kiếm lặp đi lặp lại các vi phạm điều kiện (2) [các phụ thuộc dư thừa] và điều kiện (3) [các thuộc tính dư thừa ở vế trái], và sửa đổi tập phụ thuộc cho thích ứng. Bởi vì mỗi khi sửa đổi, chúng ta xoá một phụ thuộc (2) hoặc xoá một thuộc tính trong một phụ thuộc (3), quá trình này không thể tiếp tục mãi, cuối cùng chúng ta sẽ thu được một tập phụ thuộc không vi phạm các điều kiện (1), (2), hoặc (3).

Đối với điều kiện (2), chúng ta xét mỗi phụ thuộc $X \to Y$ trong tập phụ thuộc hiện tại F, và nếu $F - \{X \to Y\}$ tương đương với F thì xoá $X \to Y$ ra khỏi F. Chú ý rằng thứ tự xét các phụ thuộc có thể loại bỏ các tập phụ thuộc khác nhau. Chẳng hạn với tập F:

$$\begin{array}{ll} A \to B & A \to C \\ B \to A & C \to A \\ B \to C & \end{array}$$

Chúng ta có thể loại bỏ cả $B \to A$ lẫn $A \to C$ hoặc có thể loại bỏ $B \to C$ nhưng không thể loại bỏ cả ba được.

Đối với điều kiện (3), chúng ta xét mỗi phụ thuộc $A_1 \ldots A_k \to B$ trong tập phụ thuộc F, và mỗi thuộc tính A_i ở vế trái theo một thứ tự nào đó. Nếu

$(F - \{A_1 \ldots A_{i-1} A_i A_{i+1} \ldots A_k \to B\})$

$\cup \{A_i \ldots A_{i-1} A_{i+1} \ldots A_k \to B\}$

tương đương với F thì xoá A_i ra khỏi vế trái của $A_1 \ldots A_k \to B$.

Một lần nữa, thứ tự các thuộc tính bị loại bỏ có thể ảnh hưởng đến kết quả.

Chẳng hạn cho tập phụ thuộc

$AB \to C \qquad A \to B \qquad B \to A$

Chúng ta có thể loại bỏ A hoặc B ra khỏi $AB \to C$ nhưng không thể loại cả hai.

Bạn đọc hãy chứng minh rằng khi loại bỏ tất cả các vi phạm của (3) trước rồi đến tất cả các vi phạm của (2), chúng ta sẽ có được phủ cực tiểu nhưng thực hiện ngược lại thì không chắc. □

Thí dụ II.4:

Xét tập phụ thuộc F của Thí dụ II.3. Nếu dùng thuật toán của Bổ đề II.5 để tách vế phải, chúng ta thu được:

$AB \to C$	$D \to E$	$CG \to B$
$C \to A$	$D \to G$	$CG \to D$
$BC \to D$	$BE \to C$	$CE \to A$
$ACD \to B$		$CE \to G$

Rõ ràng $CE \to A$ là dư thừa bởi vì nó được suy ra từ $C \to A$. $CG \to B$ cũng dư thừa do các phụ thuộc $CG \to D$, $C \to A$ và suy ra $CG \to B$. Ngoài ra không còn phụ thuộc nào dư thừa nữa. Tuy nhiên, chúng ta có thể thay $ACD \to B$ bằng $CD \to B$, vì có thể suy ra $CD \to B$ từ $ACD \to B$ và $C \to A$. Bây giờ không còn rút gọn các phụ thuộc được nữa. Kết quả là một phủ cực tiểu của F được trình bày như trong Hình II.4 (a).

Một phủ cực tiểu khác, được xây dựng từ F bằng cách loại bỏ các phụ thuộc $CE \to A$,$CG \to D$, và $ACD \to B$ được trình bày trong Hình II.4 (b). Chú ý rằng hai phủ cực tiểu này có số lượng phụ thuộc khác nhau. □

(a)	(b)
$AB \rightarrow C$	$AB \rightarrow C$
$C \rightarrow A$	$C \rightarrow A$
$BC \rightarrow D$	$BC \rightarrow D$
$CD \rightarrow B$	$D \rightarrow E$
$D \rightarrow E$	$D \rightarrow G$
$D \rightarrow G$	$BE \rightarrow C$
$BE \rightarrow C$	$CG \rightarrow B$
$CG \rightarrow D$	$CE \rightarrow G$
$CE \rightarrow G$	

- Hình II.4 – Hai phủ cực tiểu –

II.14. Tính chất của phụ thuộc hàm

Cho lược đồ quan hệ R xác định trên tập thuộc tính U. Cho f : X→Y là một phụ thuộc hàm trên U. Gọi r là một quan hệ của R. Ta có các tính chất sau:

II.14.1. Tính chiếu

Lấy $W \subseteq U$ sao cho $X \subseteq W$ và $Y \cap W <> 0$ ta có :
Nếu r thỏa X→Y thì r[W] cũng thỏa $X \rightarrow (Y \cap W)$

Thí dụ:
Cho R(ABC), với U=ABC, gọi r là một quan hệ của R.
Ta có :
Nếu r thỏa f : A→BC thì r[AB] cũng thỏa f : A→B
Ở đây ta lấy W = AB

r = (A B C)	r[AB] = (A B)
a1 b1 c1	a1 b1
a2 b1 c2	a2 b1

Bạn hãy chứng minh tính chất trên

II.15. Ứng dụng khái niệm phụ thuộc hàm vào khóa

Khi nói đến các tập thực thể, chúng ta đã giả sử rằng tồn tại một khoá, đó là tập các thuộc tính xác định duy nhất một thực thể. Cũng có một khái niệm tương tự cho các quan hệ có các phụ thuộc hàm. Ta có định nghĩa về khóa sau.

II.15.1. Định nghĩa khóa : dùng phụ thuộc hàm

Cho R là một lược đồ quan hệ với các thuộc tính $A_1A_2 \ldots A_n$ và cho trước tập các phụ thuộc hàm F. Gọi X là một tập con của $A_1A_2 \ldots A_n$ chúng ta nói X là một khoá của (R,F) nếu:

(i) Phụ thuộc hàm $X \rightarrow A_1A_2 \ldots A_n$ thuộc F^+.
Nghĩa là có sự phụ thuộc của tất cả các thuộc tính vào tập thuộc tính X được cho trước hoặc được suy ra từ những phụ thuộc đã cho, và

(ii) Không có tập Y nào là tập con thực sự của X, mà có phụ thuộc $Y \rightarrow A_1A_2 \ldots A_n$ thuộc F^+.

Thí dụ :

Trong Thí dụ II.1. Gọi R=ABC và F= {A → B, B → C} Ta thấy chỉ có một khoá duy nhất là A, bởi vì A → ABC thuộc F^+ và không có bất kỳ tập thuộc tính X nào không chứa A mà X → ABC đúng.

II.15.2. Thuật toán tìm khoá : vét cạn

Từ định nghĩa trên, ta có thể xây dựng một thuật toán tìm khóa của một lược đồ quan hệ một cách hệ thống hơn. Đã có nhiều thuật toán được xây dựng, tư tưởng chính của những thuật toán đó dựa trên các bước sau:

NHẬP:
tập các thuộc tính R, tập các phụ thuộc hàm F

XUẤT:
các khóa của (R,F)

PHƯƠNG PHÁP:

1. *Xây dựng các tập con khác rỗng của R*
2. *Với mỗi tập $K \subseteq R$ đã xây dựng ở bước 1:*
   Nếu K thỏa điều kiện (i)
   Thì K là một siêu khóa
   Cuối nếu
   Cuối mọi
   Gọi S là tập các siêu khóa
3. *Tìm siêu khóa nhỏ nhất (điều kiện ii)*
   Với mỗi tập $K \in S$ đã xây dựng ở bước 1:
   Nếu $\exists K' \in S$ sao cho $K' \subset K$ ($K' \neq K$)
   Thì loại bỏ K khỏi S
   Cuối nếu
   Cuối mọi
   S chính là tập chứa các khóa của (R,F)

II.15.3. Thuật toán tìm khoá dựa trên đồ thị

Vấn đề của thuật toán trên là kết quả của bước 1 có thể là một tập rất lớn các tập con của R. Nếu R có n thuộc tính thì số lượng các tập con khác rỗng của R là 2^n. Do đó cần giới hạn số tập con cần khảo sát dựa vào các tính chất sau:

a) Mỗi nút của đồ thị là tên một thuộc tính của R

b) Mỗi phụ thuộc hàm được thể hiện trên 1 cung có hướng của đồ thị

c) Nút lá: B là nút (thuộc tính) lá nếu:

- tồn tại một phụ thuộc hàm thuộc F sao cho B xuất hiện bên vế phải của phụ thuộc hàm nầy.
- và không tồn tại một phụ thuộc hàm thuộc F sao cho B xuất hiện bên vế trái của phụ thuộc hàm nầy.

Trên đồ thị phụ thuộc hàm, nút thuộc tính lá có cung vào mà không có cung ra.

Nhận xét: Thuộc tính lá không xuất hiện trong khóa

a) Nút gốc: A là nút (thuộc tính) gốc nếu:

- không tồn tại một phụ thuộc hàm thuộc F sao cho A xuất hiện bên vế phải của phụ thuộc hàm nầy.

Trên đồ thị phụ thuộc hàm, nút thuộc tính gốc không có cung vào (và không bắt buộc phải có cung ra).

Nhận xét: mọi thuộc tính gốc phải xuất hiện trong mọi khóa

Như vậy khóa của lược đồ quan hệ phải bao phủ tập các nút gốc, đồng thời không chứa bất kỳ nút lá nào của đồ thị.

CÁC BƯỚC:

1. *Vẽ đồ thị phụ thuộc hàm và xác định các tập:*
   - GOC: tập gồm các nút gốc
   - LA : tập gồm các nút lá
   - TG = R - (GOC $\cup$ LA) : tập gồm các nút trung gian
2. *Xuất phát từ tập X=GOC, ta tính bao đóng X^+ dựa vào các phụ thuộc hàm trong F.*
   <u>Nếu</u> $X^+ = R$
   <u>Thì</u> : (R,F) chỉ có duy nhất 1 khóa là GOC
   <u>Ngược lại</u> :bổ sung từng tập con klhác rỗng của tập TG vào X (theo thứ tự từ tập gồm 1 phần tử, 2 phần tử, ...). Tính X^+ cho tới khi tìm được X sao cho $X^+ = R$
   <u>Cuối nếu</u>

III. PHỤ THUỘC ĐA TRỊ

III.1. Phụ thuộc đa trị

Trong những phần trước chúng ta đã giả thiết rằng chỉ có một loại phụ thuộc là phụ thuộc hàm. Thật ra có nhiều loại phụ thuộc, và ít nhất có một loại phụ thuộc rất thường gặp trong thế giới thực là *phụ thuộc đa trị* (multivalued dependency).

III.1.1. Định nghĩa

Giả sử chúng ta có một lược đồ quan hệ R , và X, Y là các tập con của R.

Về trực quan, *ta nói rằng $X \rightarrow\rightarrow Y$, đọc là "X đa xác định Y" hoặc "có một phụ thuộc đa trị của Y vào X" , nếu với những giá trị đã biết cho các thuộc tính của X, tồn tại một tập gồm zero hoặc nhiều giá trị cho các thuộc tính của Y, và tập giá trị Y này không liên hệ gì với những giá trị của các thuộc tính trong R - X - Y.*

Về hình thức, *ta nói $X \rightarrow\rightarrow Y$ là một phụ thuộc đa trị được định nghĩa trên R nếu với r là một quan hệ của R, và q_1 và q_2 là hai bộ trong r, với $q_1[X] = q_2[X]$ (nghĩa là q_1 và q_2 giống nhau ở các thuộc tính của X) thì r cũng chứa các bộ q_3 và q_4, trong đó :*

1. $q_3[X] = q_4[X] = q_1[X] = q_2[X]$
2. $q_3[Y] = q_1[Y]$ và $q_3[Z] = q_2[Z]$; với $Z=R - X - Y$
3. $q_4[Y] = q_2[Y]$ và $q_4[Z] = q_1[Z]$

Nghĩa là chúng ta có thể trao đổi các giá trị Y của q_1 và q_2, thu được hai bộ mới q_3 và q_4 thuộc r. Chú ý rằng chúng ta không giả thiết X và Y là tách biệt trong định nghĩa trên.

Lưu ý: ta có thể loại bỏ mệnh đề (3). Tồn tại của bộ q_4 suy ra từ bộ q_3 khi đảo vị trí của q_1 và q_2 trong định nghĩa.

Thí dụ II.6:

Chúng ta hãy xét lược đồ gồm các thuộc tính CTHRSG. Hình II.5 trình bày một quan hệ của lược đồ này. Ở trường hợp đơn giản này chỉ có một khóa học và hai sinh viên, nhưng có nhiều đặc điểm mà chúng ta hy vọng sẽ đúng trong mọi quan hệ của lược đồ. Một khoá học (lớp học, **C**ourse) có thể học trong nhiều buổi (giờ học, **H**ours), mỗi buổi ở một phòng (**R**oom) khác nhau. Mỗi sinh viên (**S**tudent) có một bộ cho mỗi lớp (khóa học) và mỗi buổi học của lớp đó. Điểm (**G**rade) được lập lại ở mỗi bộ.

C	**T**	**H**	**R**	**S**	**G**
CS101	Minh	T.hai 9	P.222	Dung	B+
CS101	Minh	T.tu 9	P.333	Dung	B+
CS101	Minh	T.sau 9	P.222	Dung	B+
CS101	Minh	T.hai 9	P.222	Lan	C
CS101	Minh	T.tu 9	P.333	Lan	C
CS101	Minh	T.sau 9	P.222	Lan	C

-Hình II.5 Một quan hệ mẫu cho lược đồ CTHRSG-

Vì thế trong trường hợp tổng quát, chúng ta hy vọng rằng phụ thuộc đa trị C $\rightarrow\rightarrow$ HR đúng: nghĩa là có một tập các cặp *giờ học-phòng* kèm với mỗi khoá học và không có liên quan gì với những thuộc tính khác. Chẳng hạn trong định nghĩa hình thức của phụ thuộc đa trị, chúng ta có thể xem X $\rightarrow\rightarrow$Y là C$\rightarrow\rightarrow$ HR và chọn.

q_1 = CS101, Minh, T.hai 9, P.222, Dung, B+
q_2 = CS101, Minh, T.tu 9, P.333, Lan , C

nghĩa là q_1 là bộ đầu tiên, q_2 là bộ thứ năm trong Hình II.5.

Thế thì chúng ta hy vọng rằng có thể trao đổi q_1[HR] = (M9, 222) với q_2[HR] = (W9, 333) để thu được hai bộ

q_3 = CS101, Minh, T.hai 9, P.222, Lan , C

q_4 = CS101, Minh, T.tu 9, P.333, Dung, B+

Nhìn thoáng qua, Hình II.5 cho thấy rằng q_3 và q_4 thực sự thuộc r , tương ứng là bộ thứ tư và thứ hai.

Cần phải nhấn mạnh rằng C $\rightarrow\rightarrow$ HR đúng không phải do nó đúng trong quan hệ của Hình II.5. Nó đúng bởi vì một khoá học c bất kỳ, nếu nó được dạy vào giờ h_1 trong phòng γ_1 với giảng viên t_1 và sinh viên s_1 có điểm g_1, và nó cũng được dạy vào giờ h_2 trong phòng γ_2 với giảng viên t_2 và sinh viên s_2 có điểm g_2, thì nó cũng được dạy vào giờ h_1 trong phòng γ_1 với giảng viên t_2 và sinh viên s_2 có điểm g_2.

Cũng chú ý rằng C$\rightarrow\rightarrow$ H không đúng, và C $\rightarrow\rightarrow$ R cũng thế.

Thật vậy, hãy xét quan hệ r của Hình II.5 với các bộ q_1 và q_2 ở trên. Nếu C$\rightarrow\rightarrow$ H đúng, chúng ta hy vọng rằng sẽ tìm được bộ

CS101, Minh , T.tu 9, P.333, Lan , C

trong r, mà chúng ta không tìm thấy.

Cũng nhận xét tương tự cho C$\rightarrow\rightarrow$ R.

Tuy nhiên còn có một số phụ thuộc đa trị khác như: $C\to\to SG$ và $HR \to\to SG$. Ngoài ra cũng có những phụ thuộc đa trị tầm thường như $HR\to\to R$. Bạn hãy tự kiểm chứng.

III.2. Các tiên đề cho phụ thuộc hàm, phụ thuộc đa trị

Bây giờ chúng ta sẽ trình bày một tập các tiên đề đúng đắn và đầy đủ dùng để suy diễn về tập phụ thuộc hàm và tập phụ thuộc đa trị trên tập thuộc tính U.

Ba tiên đề đầu tiên chính là các tiên đề Armstrong cho phụ thuộc hàm được lập lại ở đây.

A1: Tính phản xạ phụ thuộc hàm.

Nếu $Y \subseteq X \subseteq U$ thì $X\to Y$

A2: Tính tăng trưởng của phụ thuộc hàm.

Nếu $X \to Y$ và $Z \subseteq U$ thì $XZ\to YZ$

A3: Tính bắc cầu của phụ thuộc hàm.

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

Ba tiên đề sau áp dụng cho các phụ thuộc đa trị.

A4: Tính bù của phụ thuộc đa trị.

Nếu $X\to\to Y\}$ thì $X \to\to (U - X - Y)$

A5: Tính tăng trưởng của phụ thuộc đa trị.

Nếu $X \to\to Y$ và $V \subseteq W$ thì $WX \to\to VY$

A6: Tính bắc cầu của phụ thuộc đa trị:

Nếu $X \to\to Y$ và $Y\to\to Z$ thì $X \to\to (Z - Y)$

Cũng nên so sánh A4-A6 với A1-A3. Tiên đề A4, là qui tắc bù, không có đối tác ở phụ thuộc hàm. Tiên đề A1, là tính phản xạ, dường như cũng không có đối tác tương ứng ở phụ thuộc đa trị, nhưng có thể suy ra $X \to\to Y$ khi $Y\subseteq X$ từ A1 và tiên đề A7 (bên dưới) là nếu có phụ thuộc $X\to Y$ thì cũng có phụ thuộc $X\to\to Y$. A6 bị hạn chế nhiều hơn so với đối tác của nó là A3. Khẳng định tổng quát hơn, $X\to\to Y$ và $Y\to\to Z$ suy ra $Y\to\to Z$. Chẳng hạn trong Thí dụ II.6, $C\to\to HR$ đúng, và chắc chắn HR $\to\to$ H cũng đúng, nhưng C $\to\to$ H sai. Bù lại, A5 mạnh hơn so với tiên đề tăng trưởng tương ứng là A2.

Chúng ta có thể thay A2 bằng: $X \to Y$ và $V \subseteq W$ suy ra $WX \to VY$, nhưng đối với phụ thuộc hàm, qui tắc này dễ dàng chứng minh các tiên đề A1, A2 và A3.

Hai tiên đề cuối cùng liên quan đến phụ thuộc hàm và phụ thuộc đa trị.

A7: Nếu $X \to Y$ thì $X \to\to Y$
A8: Nếu $X \to\to Y$ và $Z \subseteq Y$, và tập W tách biệt với Y mà chúng ta có $W \to Z$ thì $X \to Z$

III.3. Tính đúng đắn và đầy đủ của các tiên đề

Chúng ta không chứng minh rằng các tiên đề A1-A8 là đúng đắn và đầy đủ. Thực ra, chúng ta sẽ chứng minh rằng một số tiên đề là đúng đắn, nghĩa là chúng được suy ra từ định nghĩa của phụ thuộc hàm và phụ thuộc đa trị, dành cho bạn đọc chứng minh tính đúng đắn của các tiên đề còn lại, cũng chứng minh rằng mọi suy diễn có giá trị đều có thể thực hiện bằng cách sử dụng các tiên đề này (Tính đầy đủ của các tiên đề).

Chúng ta sẽ chứng minh tiên đề A6, là tính bắc cầu của phụ thuộc đa trị. Giả sử một quan hệ γ trên tập các thuộc tính *U* thoả $X \to\to Y$ và $Y \to\to Z$, nhưng không thoả phụ thuộc $X \to\to (Z-Y)$. Thế thì tồn tại các bộ μ và ν trong γ với $\mu[X] = \nu[X]$, nhưng bộ ϕ với $\phi[X] = \mu[X]$, $\phi[Z-Y] = \mu[Z-Y]$, và

$\phi[U-X-(Z-Y)] = \nu[U-X-(Z-Y)]$

không thuộc γ. Bởi vì $X \to\to Y$ đúng, suy ra rằng bộ Ψ, trong đó $\Psi[X] = \mu[X]$, $\Psi[Y] = \nu[Y]$ và

$\Psi[U-Y-Z] = \nu[U-Y-Z]$

thuộc γ. Bây giờ Ψ và ν giống nhau ở Y, thế nên do $Y \to\to Z$, suy ra rằng γ có bộ ω, trong đó $\omega[Y] = \nu[Y]$, $\omega[Z] = \Psi[Z]$, và

$\omega\,[U\text{-}Y\text{-}Z] = \nu\,[U\text{-}Y\text{-}Z]$

Chúng ta khẳng định rằng $\omega[X] = \mu[X]$, bởI vì ở các thuộc tính trong $Z \cap X$, ω gống với Ψ, mà Ψ lại giống với μ. Ở các thuộc tính của $X\text{-}Z$, ω giống với ν, và ν giống với μ ở X. Chúng ta cũng khẳng định rằng $\omega[Z\text{-}Y] = \mu[Z\text{-}Y]$, bởi vì ω giống với Ψ ở $Z\text{-}Y$, mà Ψ giống với μ ở $Z\text{-}Y$. Cuối cùng, chúng ta khẳng định rằng $\omega[V] = \nu[V]$, với $V=U\text{-}X\text{-}(Z_x\text{-}Y)$. Thật vậy, chắcchắn rằng ω giống với ν ở $V\text{-}Z$, và nhờ các phép biến đổi tập hợp, chúng ta có thể chứng minh $V \cap Z=(Y \cap Z)\text{-}X$. Nhưng ω giống với Ψ ở Z, mà Ψ giống với ν ở Y, vì thế ω giống với ν ở $V \cap Z$ cũng như ở $V\text{-}Z$. Do đó ω giống với ν ở V. Nếu xem lại định nghĩa của ϕ, chúng ta thấy rằng $\omega = \phi$. Nhưng chúng ta đã khẳng định rằng ω thuộc γ, vì vậy ϕ cũng thuộc γ, mâu thuẫn với giả thiết của chúng ta. Vì thế cuối cùng $X \to\to Z\text{-}Y$ đúng, khẳng định A6 đúng.

Bây giờ chúng ta sẽ chứng minh A8. Giả sử điều ngược lại là chúng ta có một quan hệ γ thoả $X \to\to Y$ và $W \to Z$, với $Z \subseteq Y$ và $W \cap Y$ là tập trống, nhưng $X \to Z$ không đúng. Thế thì tồn tại các bộ ν và μ trong γ sao cho $\nu[X] = \mu[X]$, nhưng $\nu[Z] \neq \mu[Z]$. Áp dụng $X \to\to Y$ cho ν và μ, chúng ta phải có một bộ ϕ trong γ sao cho $\phi[X] = \mu[X] = \nu\,[X]$, $\phi[Y] = \mu[Y]$, và $\phi[U\text{-}X\text{-}Y] = \nu\,[U\text{-}X\text{-}Y]$. Bởi vì $W \cap Y$ trống, nên ϕ và ν giống nhau ở W. Vì $Z \subseteq Y$, ϕ và μ giống nhau ở Z. Bởi vì ν và μ không giống nhau ở Z, suy ra rằng ϕ và ν không giống ở Z. Chúng ta kết luận rằng $X \to Z$ không thể không đúng, đó chính là khẳng định của qui tắc A8.

Phần chứng minh còn lại của định lý sau dành cho bạn đọc.

Lưu ý:

nên nhớ rằng trong định nghĩa của phụ thuộc đa trị thực sự chỉ cần tồn tại q_3, không nhất thiết phải có q_4 như trong mệnh đề thứ 3 của định nghĩa. Vì thế vi phạm phụ thuộc đa trị có thể được xác định bằng sự vắng mặt của q_3 (không cần phải q_3 hoặc q_4) trong quan hệ r.

III.3.1. Định lý II.4 (been, ragin, floward 1997)

Các tiên đề A1-A8 là đúng đắn và đầy đủ cho các phụ thuộc hàm và phụ thuộc đa trị. Nghĩa là nếu D là tập các phụ thuộc hàm và phụ thuộc đa trị trên một tập thuộc tính U, và D' là tập các phụ thuộc hàm và đa trị hệ quả của D (nghĩa là mỗi quan hệ trên U thỏa D cũng thỏa những phụ thuộc trong D') thì D' chính là tập các phụ thuộc suy ra từ D bởi các tiên đề A1-A8.

III.4. Các qui tắc suy diễn bổ sung cho phụ thuộc đa trị

Một số qui tắc khác cũng có ích trong các suy diễn về các phụ thuộc hàm và đa trị. Dĩ nhiên các qui tắc hợp, phân rã và giả bắc cầu đã đề cập trong Bổ đề II.2 vẫn áp dụng được cho các phụ thuộc hàm.

Một số qui tắc khác là:

1. *Qui tắc hợp của phụ thuộc đa trị.*
   Nếu $X\rightarrow\rightarrow Y$ *và* $X\rightarrow\rightarrow Z$ *thì* $X\rightarrow\rightarrow YZ$
2. *Qui tắc giả bắc cầu của phụ thuộc đa trị.*
   Nếu $X\rightarrow\rightarrow Y$ *và* $WY\rightarrow\rightarrow Z$ *thì* $WX\rightarrow\rightarrow (Z-WY)$
3. *Qui tắc giả bắc cầu pha trộn.*
   Nếu $X\rightarrow\rightarrow Y$ *và* $XY\rightarrow Z$ *thì* $X\rightarrow (Z-Y)$
4. *Qui tắc phân rã và phụ thuộc đa trị.*

   Nếu $X\rightarrow\rightarrow Y$, *và* $X\rightarrow\rightarrow Z$

   thì $X\rightarrow\rightarrow (Y\cap Z)$ *và* $X\rightarrow\rightarrow (Y-Z)$ *và* $X\rightarrow\rightarrow (Y-Z)$

Chúng tôi để phần chứng minh này cho bạn đọc; phương pháp chứng minh cũng tương tự như phương pháp đã được dùng để chứng minh các quy tắc A6 và A8 ở trên, hoặc có thể chứng minh từ các tiên đề A1-A8.

Chúng ta cần chú ý rằng qui tắc phân rã của phụ thuộc đa trị không mạnh bằng qui tắc tương ứng của phụ thuộc hàm. Quy tắc của phụ thuộc hàm cho phép chúng ta suy ra trực tiếp từ $X \rightarrow Y$ rằng $X \rightarrow A$ đúng với mọi các thuộc tính A trong Y. Qui tắc của phụ thuộc đa trị chỉ cho phép chúng ta kết luận $X \rightarrow\rightarrow A$ từ $X \rightarrow\rightarrow Y$ nếu chúng ta có thể tìm được Z sao cho $X \rightarrow\rightarrow Z$, hoặc $Z \cap Y = A$ hoặc $Y - Z = A$

III.5. Bao đóng của phụ thuộc hàm và phụ thuộc đa trị

Cho trước tập các phụ thuộc hàm và phụ thuộc đa trị D, chúng ta muốn tìm bao đóng của D, ký hiệu D^+, là tập chứa tất cả các phụ thuộc hàm và phụ thuộc đa trị được suy ra từ D.

Chúng ta có thể tính D^+ bằng cách khởi đầu với D và áp dụng các tiên đề A1-A8 đến khi không còn suy ra được một phụ thuộc mới nào nữa. Tuy nhiên quá trình này có chi phí thời gian là hàm mũ theo kích thước của D.

Thông thường chúng ta chỉ muốn biết xem một phụ thuộc hàm $X \rightarrow Y$ hoặc phụ thuộc đa trị $X \rightarrow\rightarrow Y$ nào đó có suy ra được từ D hay không.

III.5.1. Cơ sở phụ thuộc

Như đã trình bày khi nói về phụ thuộc hàm, để kiểm tra một phụ thuộc hàm $X \rightarrow Y$ có thuộc D^+ hay không, ta tính X^+ là bao đóng của tập thuộc tính X đối với tập các phụ thuộc hàm trong D, sau đó xem Y có phải là tập con của X^+ hay không, nghĩa là $Y = A_1A_2 \ldots A_k$,với A_i là thuộc tính thuộc X^+

Tương tự, để kiểm tra xem một phụ thuộc đa trị $X \to\to Y$ có đúng hay không (thuộc D^+ hay không), chúng ta chỉ cần xác định *cơ sở phụ thuộc* của X và xem Y –X có phải là hợp của một số tập trong cơ sở đó hay không.

Nhận xét:

- *Bao đóng* của $X = A_1A_2 \ldots A_k$,với A_i là *thuộc tính*

- *Cơ sở phụ thuộc* của $X = Y_1Y_2 \ldots Y_k$,với Y_i là *tập các thuộc tính*

Với qui tắc phân rã của phụ thuộc đa trị, cùng với quy tắc hợp, dẫn đến khẳng định sau đây về các tập Y sao cho $X \to\to Y$ đúng với một tập X đã cho.

III.5.2. Định lý II.4 :

Nếu U là tập chứa tất cả các thuộc tính thì chúng ta có thể phân hoạch U-X thành các tập thuộc tính $Y_1 ,...,Y_k$ sao cho nếu $Z \subseteq U-X$ thì $X \to\to Z$ đúng nếu và chỉ nếu Z là hợp của một số Y_i.

Nhận xét:

- Mỗi Y_i là không rỗng, với i=1, 2, …, k

- Mỗi cặp các Y_i, Y_j là phân biệt (có giao là rỗng), với i,j=1, 2, …, k

- *U-X là hợp của các tập Y_1 , Y_2 ,...,Y_k*

- Z là con của U-X

Chứng minh:

Khởi đầu phân hoạch toàn bộ U-X thành một khối (W_1=U-X).

Ta có thể phân họach như trên bởi vì do tiên đề A1 (tính phản xạ của phụ thuộc hàm) ta có phụ thuộc hàm hiển nhiên $X \rightarrow X$, hơn nữa do tiên đề A7 (liên hệ giữa phụ thuộc hàm và phụ thuộc đa trị) ta có $X \rightarrow\rightarrow X$ đúng. Áp dụng tiên đề A4 (tính bù của phụ thuộc đa trị) ta có $X \rightarrow\rightarrow (U-X-X)$ tức là $X \rightarrow\rightarrow (U-X)$ đúng.

Giả sử rằng tại một điểm nào đó chúng ta có các phân hoạch $W_1, \ldots\ldots, W_n$ và $X \rightarrow\rightarrow W_i$ với i = 1,2,….n.

Nếu $X \rightarrow\rightarrow Z$ đúng và Z không phải là hợp của một số W_i, hãy thay mỗi W_i có $W_i \cap Z$ và W_i - Z đều không trống bởi $W_i \cap Z$ và W_i - Z .

Áp dụng qui tắc phân rã cho $X \rightarrow\rightarrow W_i$ và $X \rightarrow\rightarrow Z$, ta có $X \rightarrow\rightarrow (W_i \cap Z)$ và $X \rightarrow\rightarrow (W_i - Z)$ đúng.

Bởi vì chúng ta không thể phân hoạch vô hạn một tập các thuộc tính hữu hạn, cuối cùng chúng ta thấy rằng mỗi Z có $X \rightarrow\rightarrow Z$ đúng đều là hợp của một số phân hoạch.

Nhờ qui tắc hợp, X đa xác định hợp của một tập phân hoạch bất kỳ.

Chúng ta gọi tập $Y_1 \ldots\ldots Y_k$ được xây dựng cho X từ tập các phụ thuộc hàm và đa trị D là *cơ sở phụ thuộc* (dependency basis) của X (ứng với D).

Thí dụ II.7.:

Trong Thí dụ II.6 chúng ta đã nhận xét rằng $C \to\to HR$. Vì thế theo qui tắc bù, $C \to\to TSG$. Chúng ta cũng biết rằng $C \to T$. Vì thế nhờ tiên đề A7, $C \to\to T$. Theo quy tắc phân rã, $C \to\to SG$. Cũng có thể kiểm chứng rằng không có một thuộc tính đơn nào trừ T hoặc C được xác định đa trị bởi C. Vì thế cơ sở phụ thuộc cho C là {T, HR, SG}. Về trực quan chúng ta thấy rằng, đi kèm với mỗi khoá học là các tập giảng viên (chỉ có duy nhất một tập), các cặp giờ học – phòng cho biết thời gian và địa điểm khoá học, và các cặp sinh viên - điểm, là danh sách sinh viên của khoá học.

Nhận xét:

Trong trường hợp tổng quát với $X \to\to Z$ mà $Z \not\subset U-X$, tức là $Z \cap X$ không rỗng. Áp dụng qui tắc phân rã cho $X \to\to Z$ và $X \to\to X$ ta có các phụ thuộc đa trị $X \to\to (Z-X)$ và $X \to\to (Z \cap X)$ cũng đúng.

Với $X \to\to (Z - X)$, vì $(Z - X) \subseteq (U - X)$ nên do định lý trên $(U - X)$ là hợp của một số các Y_i là cơ sở phụ thuộc của X.

Với $X \to\to (Z \cap X)$, vì $(Z \cap X) \subseteq X$ nên do tiên đề A1 và tiên đề A7 ta có $X \to\to Ai$ với $(Z \cap X) = A_1 A_2 \dots A_m$

III.5.3. Thuật toán II.2: Tính Cơ sở phụ thuộc

Để tính cơ sở phụ thuộc của X đối với D ta chỉ cần tính cơ sở phụ thuộc của X đối với tập các phụ thuộc đa trị M trong D là đủ.

Khi đó đòi hỏi M phải bao gồm:

1. Tất cả các phụ thuộc đa trị thuộc D và
2. Với mỗi phụ thuộc hàm X→Y thuộc D thì thay bằng tập các phụ thuộc đa trị $X \rightarrow\rightarrow A_1$, $X \rightarrow\rightarrow A_2$, … $X \rightarrow\rightarrow A_n$, trong đó $Y= A_1A_2…A_n$, tức là A_i thuộc Y và mỗi A_i là một thuộc tính đơn.

Một định lý khác của Beeri [1980] cho chúng ta cách lấy ra những phụ thuộc hàm không tầm thường từ cơ sở phụ thuộc được tính ứng với tập các phụ thuộc đa trị M. Beeri đã chứng minh được rằng nếu X không chứa A thì $X \rightarrow A$ đúng nếu và chỉ nếu:

1. A là tập độc nhất trong cơ sở phụ thuộc cho X ứng với tập phụ thuộc đa trị M, và

2. Có một tập thuộc tính Y không chứa A, sao cho $Y \rightarrow Z$ là một trong những phụ thuộc của D và A thuộc Z.

Ta có thuật tóan tính cơ sở phụ thuộc sau:

NHẬP:

Tập phụ thuộc đa trị *M* trên tập các thuộc tính *U* và tập $X \subseteq U$.

XUẤT:

Cơ sở phụ thuộc cho *X* ứng với *M*.

PHƯƠNG PHÁP:

Chúng ta khởi đầu với một tập các tập hợp ***S*** mà cuối cùng sẽ trở thành cơ sở phụ thuộc. Lúc ban đầu, ***S*** chỉ chứa một tập là U-X; nghĩa là ***S*** $= \{U\text{-}X\}$.

Chúng ta tìm lặp đi lặp lại các phụ thuộc $V \rightarrow\rightarrow W$ trong M và một tập Y trong ***S*** sao cho Y có giao với W nhưng không giao với V cho đến khi không còn thay đổi nào đối với ***S***. Sau đó thay Y bằng $Y \cap W$ và Y-W trong ***S***. Tập các tập hợp **S** cuối cùng cũng là cơ sở phụ thuộc cho X.

Bởi vì thuật toán II.2 chỉ tách các tập trong ***S***, và nó sẽ kết thúc khi không thể tách được các tập nữa nên dễ thấy rằng nó có chi phí thời gian và hàm đa thức theo kích thước của M và U. Thực tế nếu được cài đặt cẩn thận, Thuật toán sẽ chạy trong thời gian tỉ lệ với số lượng phụ thuộc trong M nhân với luỹ thừa ba của số lượng thuộc tính trong U. Phép chứng minh khẳng định này và chứng minh tính đúng đắn của Thuật toán 7.6 có thể tham khảo trong Beeri [1890]

Chương III: THIẾT KẾ CSDL MỨC QUAN NIỆM

I. DẠNG CHUẨN CỦA LƯỢC ĐỒ QUAN HỆ

Như đã đề cập trong phần I và II của chương II, trong một số quan hệ có thể chứa các thông tin trùng lắp (dư thừa), nên việc cập nhật dữ liệu (qua các phép tính thêm,sửa và hủy) gây ra những dị thường. Vì vậy các quan hệ trên cần thiết phải được biến đổi thành các dạng phù hợp hơn. Quá trình đó được gọi là *chuẩn hóa*.

Quan hệ được chuẩn hóa là quan hệ trong đó mỗi miền của một thuộc tính chỉ chứa những giá trị *nguyên tố* tức là không phân nhỏ được nữa và do đó mỗi giá trị trong quan hệ cũng là nguyên tố.

Quan hệ có chứa các miền giá trị là không nguyên tố gọi là *quan hệ không chuẩn hóa*.

Mỗi quan hệ thuộc một trong các dạng sau:

dạng không chuẩn hóa
↓
dạng chuẩn 1
↓
dạng chuẩn 2
↓
dạng chuẩn 3
↓
dạng chuẩn BOYCE-CODD

Khi một lược đồ quan hệ được thiết kế ở dạng chuẩn càng cao (như 3NF, BCNF) thì khả năng dư thừa thông tin trong quan hệ sẽ giảm. Đặt biệt, nếu lược đồ quan hệ đạt BCNF thì quan hệ đó sẽ không có thông tin dư thừa.

Các dạng chuẩn có vai trò quan trọng nhất là dạng chuẩn 3 (3NF) và dạng chuẩn Boyce Codd (BCNF) . Mục đích của chúng là tránh được các dư thừa và các bất thường.

Chúng ta cần lưu ý: để xác định dạng chuẩn của một lược đồ quan hệ ta chỉ dựa vào tập các phụ thuộc hàm được định nghĩa trên lược đồ quan hệ đó. Trong dạng chuẩn 4 (4NF) thì ngoài tập các phụ thuộc hàm ta phải xét đến tập các phụ thuộc đa trị.

I.1. Dạng chuẩn 1 (First Normal Form : 1NF)

Định nghĩa:

Một lược đồ quan hệ R được gọi là ở dạng chuẩn 1 (1NF) nếu và chỉ nếu toàn bộ các miền có mặt trong R đều chỉ chứa các giá trị nguyên tố.

Định nghĩa nầy cho ta thấy rằng bất kỳ quan hệ chuẩn hóa nào cũng ở 1NF.

Chúng ta cần lưu ý: trong định nghĩa các dạng chuẩn còn lại luôn kèm điều kiện trước tiên là phải đạt 1NF.

Cho lược đồ quan hệ:

CHUYEN_MON (MAGV, MON_GD), trong đó:

MAGV là mã số của giáo viên và MON_GD là chuỗi gồm các môn học mà giáo viên có khả năng giảng dạy.

Xét thể hiện sau :

CHUYEN_MON (MAGV, MON_GD)
GV1 , CTDL,C,PASCAL
GV2 , CSDL,TKCSDL

Khi đó MON_GD không phải là thuộc tính nguên tố.

Một trường hợp đặc biệt liên quan đến các thuộc tính có kiểu là ngày dương lịch (Datetime). Các thuộc tính nầy thực chất là thuộc tính kép (tích của các thuộc tính: ngày, tháng, năm). Tuy nhiên, chúng có thể được xem là thuộc tính đơn (thuộc tính nguyên tố) nếu như không có hoặc hiếm khi có nhu cầu truy xuất đến từng thành phần riêng lẻ: ngày, tháng hay năm.

Nếu không có chú thích gì thêm, ta qui ước rằng những thuộc tính có miền giá trị là ngày dương lịch đều là thuộc tính nguyên tố.

I.2. Dạng chuẩn 2 (2NF)

Trước khi nghiên cứu dạng chuẩn 2, xét ví dụ sau đây:

Thí dụ III.1:
Cho lược đồ CSDL gồm 2 quan hệ: THI và SINHVIEN

THI (MONTHI GIAOVIEN)

MONTHI	GIAOVIEN
3	A
4	B
5	C

SINHVIEN(MONTHI MASV TENSV DIACHI DIEM)

MONTHI	MASV	TENSV	DIACHI	DIEM
3	11	Lan	X	8
3	12	Ha	Y	6
4	11	Lan	X	7
4	12	Ha	Y	6
5	11	Lan	X	7
5	13	Tu	Z	2

- Hình III.1 – Cơ sở dữ liệu vi phạm 2NF –

Ta thấy MONTHI là khóa của quan hệ THI và MONTHI+MASV là khóa của quan hệ SINHVIEN.

Trong quan hệ thứ hai các thuộc tính MONTHI, MASV, DIEM nói đến thông tin về kết quả thi của sinh viên. Khi đó MASV, TENSV, DIACHI nói về thông tin của đối tượng sinh viên.

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

- Trong quan hệ SINHVIEN, việc lưu trữ thông tin 1 sinh viên ví dụ như “Lan” phải lặp lại 3 lần tên, 3 lần địa chỉ. Rõ ràng là thông tin bị dư thừa (trùng lắp).

- Quá trình cập nhật dữ liệu gây nên những bất thường như sau:

Phép cập nhật

Do lý do nêu trên, khi cần sửa địa chỉ của "Lan" chẳng hạn, cần phải sửa 3 lần. Nếu việc sửa đổi bị sót sẽ xãy ra tình trạng dữ liệu không nhất quán: một sinh viên có thể có các địa chỉ khác nhau. Hơn nữa khi sửa đổi thông tin về một sinh viên lại không liên quan gì đến thông tin về kết quả thi.

Thật ra để xác định các thông tin đặc trưng về một sinh viên, chỉ cần mã số sinh viên là xác định được duy nhất thông tin về họ.

Phép thêm mới

Trong quan hệ sinh viên chỉ chứa thông tin về những sinh viên đã thi (có điểm). Nếu muốn chèn thêm một sinh viên mới (chưa thi) thì không được vì khóa MONTHI, MASV là không đầy đủ. Bất thường nầy chỉ được khắc phục nếu loại bỏ những thông tin về kết quả thi ra khỏi quan hệ.

Phép hủy bỏ

Giả sử rằng với lý do nào đó cần loại bỏ môn thi thứ 5 mà danh sách sinh viên vẫn giữ nguyên. Khi đó ở quan hệ THI xóa bộ (5,C) còn ở quan hệ SINHVIEN nếu xóa môn thi thứ 5 (xóa 2 bộ cuối cùng) thì thông tin về sinh viên "Tu" sẽ mất.

Nhận xét:

Thật ra quan hệ SINHVIEN *chưa đạt dạng chuẩn 2* nên mới xảy ra các bất thường trên.

Để khắc phục những bất lợi trên, quan hệ SINHVIEN có thể tách thành hai quan hệ SVIEN(MASV, TENSV, DIACHI) và quan hệ KETQUA(MONTHI, MASV, DIEM).

Lúc nầy cơ sở dữ liệu thành ba quan hệ và các quan hệ nầy đều ở dạng chuẩn 2.

Cơ sở dữ liệu mới được trình như sau:

THI (MONTHI GIAOVIEN)

MONTHI	GIAOVIEN
3	A
4	B
5	C

SVIEN(MASV TENSV DIACHI)

MASV	TENSV	DIACHI
11	Lan	X
12	Ha	Y
13	Tu	Z

KETQUA(MONTHI MASV DIEM)

MONTHI	MASV	DIEM
3	11	8
3	12	6
4	11	7
4	12	6
5	11	7
5	13	2

- Hình III.2 – Cơ sở dữ liệu đạt 2NF –

Định nghĩa:

Một lược đồ quan hệ R với tập các phụ thuộc hàm F, được gọi là ở dạng chuẩn 2 (2NF) nếu nó ở dạng chuẩn 1 và nếu mỗi ***thuộc tính không khóa*** *của R đều* ***phụ thuộc đầy đủ*** *vào khóa.*

Nhắc lại:

Trong một lược đồ quan hệ có ít nhất 1 khóa nhưng có thể có nhiều khóa. Mỗi khóa gồm 1 hay nhiều thuộc tính. Các thuộc tính tham gia vào 1 khóa nào đó được gọi là *thuộc tính khóa*. Các thuộc tính còn lại (không tham gia vào bất kỳ 1 khóa nào) được gọi là *thuộc tính không khóa*.

Định nghĩa:

Cho lược đồ quan hệ R = $\{A_1, A_2, \ldots, A_n\}$ và F là tập các phụ thuộc hàm. Cho X và Y là hai tập thuộc tính khác nhau của R. Ta nói Y *phụ thuộc hàm đầy đủ* vào X nếu Y *phụ thuộc hàm* vào X nhưng không phụ thuộc hàm vào bất kỳ một tập con thực sự nào của X.

Nghĩa là muốn *Y phụ thuộc hàm đầy đủ vào X* thì phải thỏa cả 2 điều kiện sau:

1. $X \rightarrow Y \in F^+$
2. $\forall X' \subseteq R$: nếu ($X' \subset X$ và $X' \neq X$) thì $X' \rightarrow Y \notin F^+$

Bây giờ chúng ta hãy xem lại quan hệ SINHVIEN đã trình bày ở *Hình III.1*

Với tập các thuộc tính R = {MONTHI, MASV, TENSV, DIACHI, DIEM}.

Và tập các phụ thuộc hàm F = {f1, f2} như sau:

f1 : MASV→TENSV, DIACHI
f2 : MONTHI, MASV→DIEM

Ta thấy khóa của lược đồ quan hệ trên là K = {MONTHI, MASV} vì:

(i) K+ = { MONTHI, MASV }
= { MONTHI, MASV, DIEM} do f1
= {MONTHI,MASV,DIEM,TENSV,DIACHI} do f2
= R

(ii) MONTHI→DIEM $\notin F^+$ và MASV→DIEM $\notin F^+$

Vậy các thuộc tính không khóa là: TENSV, DIACHI và DIEM.

Xét thuộc tính không khóa TENSV. Nhận xét rằng một thuộc tính bất kỳ của lược đồ đều phụ thuộc hàm vào khóa của lược đồ đó. Nghĩa là MONTHI, MASV→TENSV $\in F^+$

Do MASV $\rightarrow$ TENSV $\in F^+$ và

$\{MASV\} \subset K$ và $\{MASV\} \neq K$

nên MASV không phụ thuộc hàm đầy đủ vào khóa.

Kết luận: lược đồ quan hệ trên vi phạm dạng chuẩn 2.

Bạn hãy kiểm chứng: DIACHI cũng không phụ thuộc hàm đầy đủ vào khóa, nhưng DIEM thì phụ thuộc hàm đầy đủ vào khóa.

Cả ba lược đồ quan hệ: THI, SVIEN và KETQUA ở *Hình III.2* đều đạt dạng chuẩn 2.

I.3. Dạng chuẩn 3 (3NF)

Để trình bày dạng chuẩn 3, ta đưa ra khái niệm về *phụ thuộc bắc cầu* như sau:

Định nghĩa:

Cho lược đồ quan hệ R = {$A_1, A_2, \ldots, A_n$} và F là tập các phụ thuộc hàm. Cho X là tập con của R và A là một thuộc tính thuộc R. Thuộc tính A được gọi là phụ thuộc bắc cầu vào tập thuộc tính X nếu tồn tại một tập con Y của R sao cho $X \rightarrow Y \in F^+$, $Y \rightarrow A \in F^+$, nhưng $Y \rightarrow X \notin F^+$, và $A \notin XY$.

Nghĩa là: để A phụ thuộc bắc cầu vào X, ta phải tìm được tập Y thỏa các điều kiện sau:

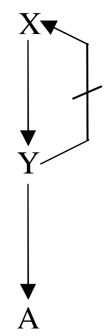
(i) $A \notin XY$; thuộc tính A không thuộc X và A không thuộc Y.

(ii) phụ thuộc hàm $X \rightarrow Y \in F^+$

(iii) phụ thuộc hàm $Y \rightarrow A \in F^+$

(iv) phụ thuộc hàm $Y \rightarrow X \notin F^+$

Tính bắc cầu thể hiện qua sơ đồ sau:



Cũng như ở 2NF việc loại bỏ phụ thuộc bắc cầu để đi đến 3NF nhằm loại bỏ những dị thường gây ra trong trong quá trình cập nhật dữ liệu. Ta có định nghĩa về dạng chuẩn 3 sau:

Định nghĩa:

Một lược đồ quan hệ R với tập các phụ thuộc hàm F, được gọi là ở dạng chuẩn 3 (3NF) nếu nó ở dạng chuẩn thứ hai và nếu mỗi ***thuộc tính không khóa*** *của R đều* ***không*** *được* ***phụ thuộc bắc cầu*** *vào khóa.*

Thí dụ III.2:

Cho lược đồ quan hệ R (SAIP) với các phụ thuộc hàm $SI \rightarrow P$ và $S \rightarrow A$.

Ta thấy SI là khóa của R vì $\{S,I\}^+ = R$
và $\{S\}^+ = \{S,A\} \neq R$ và $\{I\}^+ = \{I\} \neq R$.
Xét thuộc tính không khóa A, chọn tập trung gian Y là S.
Ta có:
(i) $A \notin \{S,I\} \cup \{S\}$.
(ii) $SI \rightarrow S \in F^+$,do tính phản xạ (tiên đề A1)
(iii) $S \rightarrow A \in F^+$,do giả thiết
(iv) $S \rightarrow SI \notin F^+$ vì $\{S\}^+ = \{S,A\}$

Vậy thuộc tính không khóa A *phụ thuộc bắc cầu* vào khóa SI, nên lược đồ quan hệ trên vi phạm 3NF.

Hơn nữa, lược đồ trên cũng không ở 2 NF vì thuộc tính không khóa A *không phụ thuộc đầy đủ* vào khóa SI do phụ thuộc hàm $S \rightarrow A$.

Thí dụ III.3:

Cho lược đồ quan hệ R (CSZ) với các phụ thuộc hàm $CS \rightarrow Z$ và $Z \rightarrow C$.

Ta thấy R có 2 khóa là SC và SZ. Bạn hãy kiểm chứng.

Vì vậy tất cả các thuộc tính đều là thuộc tính khóa, do đó lược đồ trên đạt 3NF.

Thí dụ III.4:

Cho lược đồ quan hệ R (SIDM) và các phụ thuộc hàm $SI \rightarrow D$ và $SD \rightarrow M$.

Bạn hãy kiểm chứng: lược đồ nầy chỉ có 1 khóa là SI.

Các thuộc tính không khóa là M và D. Lược đồ nầy vi phạm 3NF vì thuộc tính không khóa M phụ thuộc bắc cầu vào khóa SI khi chọn tập trung gian $Y = \{S,D\}$.

Ta có $SI \rightarrow SD \in F^+$

và $SD \rightarrow M \in F^+$

và $SD \rightarrow SI \notin F^+$

và $M \notin \{S,D\} \cup \{S,I\}$.

I.4. Dạng chuẩn BOYCE-CODD (BCNF)

Dạng chuẩn có điều kiện khắt khe hơn là dạng chuẩn Boyce Codd với định nghĩa sau:

Định nghĩa :

Lược đồ quan hệ R với tập các phụ thuộc hàm F được gọi được gọi là ở dạng chuẩn Boyce-Codd nếu X→A đúng trên R, với A là thuộc tính không thuộc X thì X là một khóa bao hàm.

Nhận xét:

(1) Theo định nghĩa trên ta phải xét mọi phụ thuộc hàm không tầm thường X→A của F^+.

(2) X là một khóa bao hàm (siêu khóa) nghĩa là X phải chứa một khóa nào đó của R.

Nói cách khác, những phụ thuộc không tầm thường duy nhất là những phụ thuộc trong đó một khoá xác định hàm một hoặc nhiều thuộc tính khác. Như thế chúng ta phải tìm các phụ thuộc $X \rightarrow A$ có vi phạm không chỉ trong những phụ thuộc đã cho mà còn trong những phụ thuộc được suy ra từ những phụ thuộc này. Tuy nhiên là nếu trong tập phụ thuộc F đã cho không có các vi phạm, và F chỉ chứa những phụ thuộc mà vế phải chỉ có một thuộc tính duy nhất thì không có vi phạm trong các phụ thuộc của F^+.

Thí dụ III.5:

Xét lược đồ quan hệ R (CSZ) với các phụ thuộc hàm CS→Z và Z→C.

Ta thấy R có 2 khóa là SC và SZ. Bạn hãy kiểm chứng.

Vì vậy tất cả các thuộc tính đều là thuộc tính khóa, do đó lược đồ trên đạt 3NF.

Lược đồ quan hệ CSZ với những phụ thuộc này không có dạng BCNF vì $Z \rightarrow C$ đúng trong CSZ nhưng Z không phải là khoá của CSZ và cũng không chứa một khoá.

Như vậy lược đồ trên đạt 3NF nhưng lại vi phạm BCNF.

Định lý III.1:

Nếu một lược đồ quan hệ R với tập các phụ thuộc hàm F là ở BCNF thì nó cũng ở 3NF.

Chứng minh:

Giả sử lược đồ quan hệ R là ở BCNF nhưng không ở 3NF. Như vậy tồn tại một thuộc tính không khóa A phụ thuộc bắc cầu vào khóa X, nghĩa là có tập Y sao cho:

(i) $A \notin XY$; thuộc tính A không thuộc X và A không thuộc Y.

(ii) phụ thuộc hàm $X \rightarrow Y \in F^+$

(iii) phụ thuộc hàm $Y \rightarrow A \in F^+$

(iv) phụ thuộc hàm $Y \rightarrow X \notin F^+$

Do (i) nên $Y \rightarrow A$ không phải là phụ thuộc hàm tầm thường. Và do (iv) ta có Y không phải là *khóa bao hàm* vì nếu ngược lại thì Y phải xác định hàm mọi tập thuộc tính của R tức là $Y \rightarrow X \in F^+$.

Theo định nghĩa về dạng chuẩn ta có (R,F) vi phạm BCNF. Điều nầy mâu thuẫn với giả thiết.

I.5. Một định nghĩa khác cho dạng chuẩn 3

Trong một số tình huống, dạng chuẩn BCNF đòi hỏi một điều kiện quá khắt khe, theo nghĩa là không thể chuyển lược đồ quan hệ thành dạng đó bằng cách phân rã mà không làm mất đi đặc tính bảo toàn các phụ thuộc. Dạng chuẩn 3 cung cấp phần lớn các ưu điểm của BCNF như loại bỏ được các bất thường có liên đới, và điều kiện của nó có thể đạt được với một lược đồ CSDL tuỳ ý mà không phải bỏ đặc tính bảo toàn phụ thuộc hoặc đặc tính nối không mất.

Do đó ta đưa thêm một định nghĩa khác (tương đương với định nghĩa cũ ở mục I.3 chương III) cho dạng chuẩn 3 như sau:

Định nghĩa :

Lược đồ quan hệ R với tập các phụ thuộc hàm F được gọi được gọi là ở dạng chuẩn 3 nếu X→A đúng trên R, với A là thuộc tính không thuộc X thì hoặc X là một khóa bao hàm hoặc A là thuộc tính khóa.

Chú ý rằng các định nghĩa của dạng chuẩn Boyce Codd và dạng chuẩn 3 đều giống nhau trừ mệnh đề " hoặc A là thuộc tính khóa", chính nó làm cho dạng chuẩn 3 bớt khắt khe hơn dạng chuẩn Boyce Codd.

Giống như BCNF, về nguyên tắc, chúng ta không chỉ xét tập phụ thuộc F đã cho mà còn phải xét tất cả các phụ thuộc trong F^+ để kiểm tra một vi phạm dạng chuẩn 3. Tuy nhiên chúng ta có thể chứng minh rằng nếu F chỉ chứa các phụ thuộc được phân rã sao cho các vế phải chỉ có một thuộc tính duy nhất thì chỉ cần kiểm tra những phụ thuộc của F.

Nhận xét:

Với mọi $X \rightarrow A \in F^+$ mà A là thuộc tính $\notin$ X. Xét các điều kiện sau:

(i) $X \supseteq K$; với K là một khóa của (R,F).
(ii) $A \in K$; với K là một khóa của (R,F).

Nếu thỏa (i) thì (R,F) đạt BCNF, đương nhiên nó cũng đạt 3NF. Nếu không thỏa (i) thì (R,F) vi phạm BCNF.

Nếu thỏa (ii) thì (R,F) đạt 3NF. Nếu không thỏa cả 2 điều kiện trên thì (R,F) mới vi phạm 3NF, đương nhiên nó cũng vi phạm BCNF.

Thí dụ III.6:

Cho lược đồ quan hệ SAIP với các phụ thuộc SI $\rightarrow$ P và S $\rightarrow$ A.

Ta thấy lược đồ nầy chỉ có một khóa duy nhất là SI. Xét phụ thuộc hàm S $\rightarrow$ A. Ta có:

- S không phải là khoá bao hàm (không thỏa i).
- A là thuộc tính không khóa (không thỏa ii)

Kết luận: lược đồ nầy vi phạm điều kiện 3NF.

Thí dụ III.7:

Xét lược đồ quan hệ R (CSZ) với các phụ thuộc hàm CS$\rightarrow$Z và Z$\rightarrow$C.

Ta thấy R có 2 khóa là SC và SZ. Lược đồ nầy có dạng 3NF bởi vì tất cả các thuộc tính của nó đều là thuộc tính khóa. Do đó với phụ thuộc hàm X$\rightarrow$A$\in$ F^+ bất kỳ, đương nhiên thỏa điều kiện (ii).

Phụ thuộc hàm Z$\rightarrow$C làm cho lược đồ vi phạm BCNF do Z không chứa một khóa nào của lược đồ.

Như vậy ta có 2 định nghĩa về 3NF. Thật ra 2 định nghĩa nầy là tương đương do định lý sau:

Định lý III.2:

Các định nghĩa về dạng chuẩn 3 ở mục I.3 và I.5 trong chương III là tương đương.

I.6. Ý nghĩa của dạng chuẩn

Mục đích của dạng chuẩn BCNF là loại bỏ dư thừa mà các phụ thuộc hàm có thể gây ra.

Giả sử rằng chúng ta có một lược đồ quan hệ R ở dạng BCNF, thế thì liệu có một dư thừa cho phép chúng ta tiên đoán giá trị của một thuộc tính bằng cách so sánh hai bộ rồi áp dụng một phụ thuộc hàm.

Nghĩa là, chúng ta có hai bộ giống nhau ở một tập thuộc tính X và không giống nhau ở tập thuộc tính Y, trong khi đó ở thuộc tính A còn lại, giá trị ở một trong hai bộ này cho phép chúng ta tiên đoán giá trị trong bộ còn lại. Hai bộ này trông giống như sau:

X	Y	A
x	y_1	a
x	y_2	?

Ở đây, x, y_1, y_2 biểu diễn cho các danh sách giá trị ở các tập thuộc tính X và Y. Lưu ý phải có $y_1 \neq y_2$, nếu không 2 bộ trên trở thành 1 bộ.

Nếu chúng ta có thể dùng phụ thuộc hàm để suy ra giá trị được chỉ ra bởi dấu chấm hỏi thì giá trị đó phải là a, và phụ thuộc được dùng phải là $Z \rightarrow A$, với $Z \subseteq X$.

Tuy nhiên, Z không thể là một khoá bao hàm, bởi vì nếu như thế thì hai bộ ở trên sẽ là cùng một bộ, bởi vì chúng giống nhau ở Z và do phụ thuộc hàm $Z \rightarrow Y$ nên $y_1 = y_2$. Vì thế, R không có dạng BCNF như đã giả thiết.

Chúng ta kết luận rằng trong quan hệ có dạng BCNF, không giá trị nào có thể được tiên đoán từ những giá trị khác bằng cách chỉ dùng các phụ thuộc hàm.

Dĩ nhiên, dạng chuẩn 3NF, ít khắt khe hơn BCNF, không thể loại bỏ được tất cả dư thừa.

Một thí dụ kinh điển là lược đồ CSZ. Lược đồ này có dạng 3NF, nhưng cho phép các cặp bộ như:

C	Z	Z
c	s_1	z
?	s_2	z

và nhờ phụ thuộc $Z \rightarrow C$, chúng ta có thể suy ra giá trị chưa biết là c. Chú ý rằng những bộ này không vi phạm phụ thuộc $CS \rightarrow Z$.

II. THIẾT KẾ CƠ SỞ DỮ LIỆU

II.1. Phân rã một lược đồ quan hệ

Thí dụ III.8:

Chúng ta hãy xét lược đồ cơ sở dữ liệu chỉ gồm một quan hệ sau:

SINHVIEN(MASV	TENSV	DIACHI	MALP	TENLP)
11	Lan	X	CNA1	Cu nhan A1
12	Hai	Y	CNA1	Cu nhan A1
13	Tu	Z	CNA2	Cu nhan A2

- Hình III.3 – Cơ sở dữ liệu chưa phân rã –

Trong lược đồ nầy có các ràng buộc (phụ thuộc hàm) sau:

- Khi biết mã sinh viên ta có thể xác định duy nhất một tên, địa chỉ và mã lớp của sinh viên đó, nghĩa là có phụ thuộc hàm MASV → TENSV, DIACHI, MALP.

- Khi biết mã lớp có thể xác định duy nhất một tên lớp, nghĩa là có phụ thuộc hàm MALP → TENLP.

Khóa của lược đồ trên là MASV. Bởi vì bao đóng của {MASV} đối với hai phụ thuộc hàm trên là tập chứa tất cả các thuộc tính của lược đồ. Hơn nữa {MASV} chỉ gồm một thuộc tính nên thỏa tính "nhỏ nhất".

Ta thấy lược đồ nầy đạt 2NF do khóa chỉ có một thuộc tính nên luôn thỏa điều kiện "phụ thuộc đầy đủ vào khóa". Nhưng lược đồ trên lại vi phạm 3NF do phụ thuộc bắc cầu MASV→MALP→TENLP.

Vậy dạng chuẩn cao nhất mà lược đồ trên có thể đạt là 2NF. Do đó quan hệ trên có chứa các thông tin trùng lắp. Giả sử lớp "CNA1" có 100 sinh viên thì tên lớp "Cu nhan A1" sẽ lặp lại 100 lần.

Để giải quyết vấn đề trên ta dùng phép "phân rã", tức là tách lược đồ quan hệ trên thành các lược đồ quan hệ con với mong muốn *các lược đồ quan hệ con mới nầy sẽ đạt dạng chuẩn cao hơn lược đồ quan hệ ban đầu*. Như vậy sẽ giảm (hay không còn) các thông tin bị dư thừa trong các quan hệ mới.

Bây giờ ta phân rã (tách) lược đồ quan hệ SINHVIEN ban đầu thành hai lược đồ quan hệ con SVIEN và LOP như sau:

SVIEN(MASV TENSV DIACHI MALP)

11	Lan	X	CNA1
12	Hai	Y	CNA1
13	Tu	Z	CNA2

LOP(MALP TENLP)

CNA1	Cu nhan A1
~~CNA1~~	~~Cu nhan A1~~
CNA2	Cu nhan A2

- Hình III.4 – Cơ sở dữ liệu sau khi phân rã –

Lược đồ quan hệ SVIEN có chứa phụ thuộc hàm MASV → TENSV, DIACHI, MALP. Lược đồ nầy có khóa là MASV và đạt BCNF.

Với lược đồ quan hệ LOP có chứa phụ thuộc hàm MALP → TENLP. Lược đồ nầy có khóa là MALP và cũng đạt BCNF.

Như vậy các lược đồ quan hệ mới đều đạt dạng chuẩn cao nhất (BCNF), cao hơn dạng chuẩn của lược đồ ban đầu (2NF). Do đó các quan hệ mới không còn chứa các thông tin trùng lắp.

Tóm lại:

Mục đích của phép phân rã là tạo ra một lược đồ cơ sở dữ liệu mới có dạng chuẩn cao hơn lược đồ cơ sở dữ liệu ban đầu.

Ngoài mục đích đã nêu, ta còn mong muốn phép phân rã đạt hai yêu cầu là:

- *có nối không mất* (bảo toàn thông tin)
- *bảo tòan phụ thuộc* (bảo toàn phụ thuộc hàm)

Hai yêu cầu nầy sẽ được nói rõ trong các phần sau. Bây giờ ta đưa ra một định nghĩa cho phép phân rã.

Định nghĩa:

Phân rã lược đồ quan hệ $R = \{A_1, A_2, \ldots, A_n\}$ *là thay nó bằng một tập* $\rho = \{R_1, R_2, \ldots, R_k\}$ *trong đó* R_i *là tập con của R sao cho*

$$R = R_1 \cup R_2 \cup \ldots \cup R_k$$

Các tập R không nhất thiết phải tách biệt.

Theo thí dụ trên ta có R={MASV, TENSV, DIACHI, MALP, TENLP}. Và phép phân rã $\rho = \{R_1, R_2\}$

với R_1= {MASV, TENSV, DIACHI, MALP }

và R_2= {MALP, TENLP}.

II.2. Phân rã có nối không mất

Tính nối không mất (bảo toàn thông tin) là yêu cầu quan trọng của phép phân rã. Ta xem lại thí dụ đã nêu như sau:

Quan hệ cũ:

		R (MASV	TENSV	DIACHI	MALP	TENLP)
	t_1	11	Lan	X	CNA1	Cu nhan A1
r	t_2	12	Hai	Y	CNA1	Cu nhan A1
	t_3	13	Tu	Z	CNA2	Cu nhan A2

- Hình III.5a – Cơ sở dữ liệu chưa phân rã –

Các quan hệ mới:

		R_1 (MASV	TENSV	DIACHI	MALP)
	t_{11}	11	Lan	X	CNA1
r_1	t_{12}	12	Hai	Y	CNA1
	t_{13}	13	Tu	Z	CNA2

		R_2 (MALP	TENLP)
	t_{21}	CNA1	Cu nhan A1
r_2	t_{22}	CNA2	Cu nhan A2

- Hình III.5b – Cơ sở dữ liệu sau khi phân rã –

Như vậy ta thay lược đồ R={MASV,TENSV,DIACHI, MALP,TENLP} bằng R_1= {MASV,TENSV,DIACHI,MALP} và R_2 = {MALP, MALP}.

Giả sử r là quan hệ (thể hiện) hiện tại của lược đồ R, ta có r là tập gồm các bộ t_1, t_2 và t_3 . Nếu CSDL sử dụng các lược đồ R_1 và R_2 thay cho R, vậy để chứa các thông tin hiện tại thì các quan hệ r_1 của R_1 và r_2 của R_2 sẽ gồm những bộ như thế nào ? Thật ra quan hệ của hai lược đồ mới này chính là hình chiếu của r trên tập các thuộc tính R_1 và R_2.

Nghĩa là $r_1 = \pi_{R1}(r)$ và $r_2 = \pi_{R2}(r)$

Bạn hãy thực hiện lại phép chiếu một quan hệ lên một tập các thuộc tính như đã trình bày trong phần các phép toán đại số ở chương I. Lưu ý loại bỏ các bộ giống nhau khi chiếu r lên tập các thuộc tính của R_2. Kết quả là:

$r_1 = \{ t_{11}, t_{12}, t_{13} \}$ và $r_2 = \{ t_{21}, t_{22} \}$

Làm thế nào chúng ta biết được r_1 và r_2 chứa các thông tin giống như r ? Một cách biết được điều đó là kiểm tra xem có thể tính được r khi chỉ biết r_1 và r_2 hay không ? Chúng ta biết rằng cách duy nhất để hồi phục lại r là lấy nối tự nhiên của r_1 và r_2 . Bạn hãy tính để cho ra kết quả phép nối tự nhiên của r_1 và r_2 như đã trình bày trong phần các phép toán đại số ở chương I. Lưu ý trong quá trình nối phải so khớp các thuộc tính cùng tên gọi.

Gọi s là kết quả của phép nối tự nhiên, tức là $s = r_1 * r_2$. Có 2 trường hợp xảy ra:

(1) Nếu $s \neq r$ thì khi biết r_1 và r_2 chúng ta không có cách nào để khẳng định quan hệ gốc của lược đồ R là r hay s. Nghĩa là nếu nối tự nhiên không khôi phục được quan hệ gốc thì không có cách nào khôi phục để thu được một quan hệ duy nhất.

(2) Nếu $s = r$ tức là các lược đồ mới R_1 và R_2 có thể thay thế cho lược đồ R ban đầu vì khi cần thiết ta có thể khôi phục được quan hệ gốc từ các quan hệ của các lược đồ mới.

Với phép phân rã luôn có được trường hợp thứ 2 với mọi quan hệ r bất kỳ của lược đồ gốc R (thỏa các phụ thuộc hàm), được gọi là phép phân rã bảo toàn thông tin hay còn gọi là phép phân rã *có nối không mất*.

Bạn hãy kiểm chứng phép phân rã ở ví dụ trên đạt yêu cầu *có nối không mất*.

II.2.1. Các nối không mất

Ta xem định nghĩa sau về tính chất nối không mất của một phép phân rã.

Định nghĩa:

Nếu R là một lược đồ quan hệ được phân rã thành các lược đồ R_1, R_2,…..R_k và F là tập phụ thuộc hàm, ta gọi đây là phân rã không mất (ứng với F) nếu với mỗi quan hệ r của R thoả F, chúng ta có:

$$r = \pi_{R1}(r) * \pi_{R2}(r) * \ldots * \pi_{Rk}(r)$$

Nghĩa là mỗi quan hệ r là nối tự nhiên của các hình chiếu của nó trên các R_i. Như chúng ta đã thấy, đặc tính nối không mất là cần thiết vì quan hệ bị phân rã cần phải được khôi phục lại từ phân rã của chính nó.

Lưu ý:

Ta chỉ xét các quan hệ r của lược đồ R mà *r thỏa F*.

Một số khẳng định về ánh xạ chiếu nối được trình bày trong Bổ đề III.1.

Trước tiên chúng ta đưa ra một số ký hiệu.

Nếu ρ ={ R_1, R_2,….R_k} là một phân rã thì m_ρ là một ánh xạ được định nghĩa là $m_\rho(r) = \pi_{R1}(r) * \pi_{R2}(r) * \ldots * \pi_{Rk}(r)$. Nghĩa là m_ρ (r) là nối tự nhiên các hình chiếu của r trên các lược đồ quan hệ trong ρ .

Vì vậy điều kiện nối không mất ứng với tập phụ thuộc F có thể diễn tả là: với mọi r thoả F, chúng ta có r = m_ρ (r).

Bổ đề III.1:

Gọi R là một lược đồ quan hệ, $\rho = \{R_1, R_2, \ldots R_k\}$ là một phân rã của R và r là một quan hệ của R, Gọi $r_i = \pi_{ri}(r)$. Thế thì:

a) $r \subseteq m_\rho(r)$.

b) Nếu $s = m_\rho(r)$ thì $\pi_{ri}(s) = r_i$

c) $m_\rho(m_\rho(r)) = m_\rho(r)$.

Chứng minh:

a) Gọi μ là một bộ thuộc r, và với mỗi i, gọi $\mu_i = \mu[R_i]$. Thế thì μ_i thuộc r_i với mọi i. Theo định nghĩa của nối tự nhiên, μ thuộc $m_\rho(r)$ vì μ giống μ_i ở các thuộc tính của R_i với mọi i.

b) Nếu $s = m_\rho(r)$ thì $r \subseteq s$, suy ra rằng $\pi_{ri}(r) \subseteq \pi_{ri}(s)$. Nghĩa là $r_i \subseteq \pi_{ri}(s)$. Để chứng minh $\pi_{ri}(s) \subseteq r_i$, giả sử rằng μ_i thuộc $\pi_{ri}(s)$ với một trị số I nào đó. Thế thì có một bộ μ thuộc s sao cho $\mu[R_i] = \mu_i$. Bởi vì μ thuộc s, nên có bộ v_j thuộc r_j sao cho $\mu[R_i] = v_j$ với mỗi j. Do vậy ở trường hợp cụ thể bên trên, $\mu[R_i]$ thuộc r_i. Nhưng $\mu[R_i] = \mu_i$ nên μ_i cũng thuộc r_i.
và do đó $\pi_{Ri}(s) \subseteq r_i$. Chúng ta kết luận rằng $r_i \subseteq \pi_{Ri}(s)$.

(c) Nếu $s = m_\rho(r)$ thì do (b), $\pi_{Ri}(s) = r_i$.
do đó $m_\rho(s) = r_1 * r_2 * \ldots r_k = m_\rho(r)$

Chúng ta nhận xét rằng nếu với mỗi i, r_i là một quan hệ nào đó của R_i và $s = r_1 * r_2 * \ldots r_k$
thì $\pi_{ri}(s)$ không nhất thiết phải bằng r_i. Lý do là r_i có thể chứa các bộ *khiếm khuyết*, là các bộ không khớp với bất kỳ bộ nào khi chúng ta lấy nối.

Chẳng hạn nếu cho R_1= AB, R_2 = BC , $r_1 = \{a_1b_1\}$, $r_2 = \{b_1c_1, b_2c_2\}$ thì $s = \{a_1b_1c_1\}$ và $\pi_{BC}(s) = \{b_1c_1\} \neq r_2$. Nhưng nói chung $\pi_{Ri}(s) \subseteq r_i$ và nếu mỗi r_i là chiếu của quan hệ r thì $\pi_{Ri}(s) = r_i$.

Khả năng lưu trữ các bộ khiếm khuyết là một ưu điểm của phân rã. Như chúng ta đã đề cập trước đây, bù lại, chúng ta phải tính toán nhiều nối hơn khi trả lời các câu vấn tin nếu có phân rã lược đồ quan hệ. Khi xem xét trên mọi phương diện, nhìn chung phân rã chỉ được sử dụng nhằm giải quyết các vấn đề đã được mô tả trong Phần II.

II.2.2. Kiểm tra tính chất nối không mất

Theo ví dụ trên, có thể kiểm tra một quan hệ cụ thể r của lược đồ R (r thỏa F) có được khôi phục bằng các quan hệ r_i mới hay không. Bằng cách chiếu r lên tập các thuộc tính R_i , sau đó nối tự nhiên các hình chiếu nầy lại, cuối cùng so sánh kết quả với r ban đầu.

Tuy nhiên, ta chưa thể khẳng định phép phân rã nầy là *có nối không mất* vì không thể kiểm tra được *mọi quan hệ* của R. Thuật toán sau giúp ta kiểm tra tính chất "nối không mất" của một phân rã.

Thuật toán III.1:
Kiểm tra tính chất nối không mất của một phân rã.

NHẬP:

- lược đồ quan hệ $R = A_1 \ldots . A_n$
- một tập phụ thuộc hàm F
- một phân rã $\rho = \{ R_1, \ldots R_k\}$

XUẤT:

- một khẳng định ρ có phải là một phân rã có nối không mất hay không.

PHƯƠNG PHÁP:

Bước 1:

Chúng ta có n thuộc tính và k lược đồ con, nên ta xây dựng một bảng gồm n cột và k hàng:

- Cột thứ j tương ứng với thuộc tính A_j
- Hàng thứ i tương ứng với lược đồ quan hệ R_i.

Ở vị trí hàng i và cột j, chúng ta đặt ký hiệu:

- a_j nếu A_j thuộc R_i
- b_{ij} nếu A_j không thuộc R_i

Bước 2:

Xét lặp đi lặp lại mỗi phụ thuộc $X \rightarrow Y$ trong F cho đến khi không còn thay đổi nào nữa trong bảng.

Mỗi lần xét $X \rightarrow Y$, chúng ta tìm những hàng giống nhau ở tất cả các cột cho các thuộc tính trong X. Nếu thấy hai hàng như thế, hãy làm cho các ký hiệu của hai hàng này bằng nhau ở các thuộc tính của Y. Khi làm cho hai ký hiệu bằng nhau, nếu một trong hai ký hiệu là a_j thì cho ký hiệu kia trở thành a_j. Nếu hai ký hiệu là b_{ij} và b_{li} thì có thể cho chúng trở thành b_{ij} hoặc b_{li} một cách tuỳ ý. Điều quan trọng cần phải nhớ là khi cho hai ký hiệu bằng nhau, tất cả các xuất hiện của chúng trong bảng cũng phải cho bằng như thế; chúng ta thực hiện không đầy đủ nếu chỉ cho bằng nhau những ký hiệu nằm trong vi phạm của phụ thuộc $X \rightarrow Y$.

Nếu sau khi sửa đổi các hàng của bảng như trên, chúng ta thu được một hàng $a_1 \ldots a_n$ (hàng chứa toàn a) thì phân rã này có nối không mất. Ngược lại đây không phải là phân rã có nối không mất . □

Thí dụ III.9:
Xét lại thí dụ Hình III.5 với:
Lược đồ R ={MASV,TENSV,DIACHI, MALP,TENLP}
và F = { f1, f2) trong đó:
f1 : MASV→ TENSV,DIACHI, MALP
f2 : MALP→ TENLP
và phân rã ρ = { R_1, R_2 } trong đó:
R_1 = {MASV,TENSV,DIACHI,MALP}
R_2 = {MALP, MALP}.

Để kiểm tra xem phân rã nầy có đặc tính có nối không mất hay không ? Áp dụng thuật giải trên ta thực hiện các bước sau:

Bước 1: lập bảng khởi đầu
- điền các giá trị a_i và b_{ij} như sau:

R	MASV	TENSV	DIACHI	MALP	TENLP
R_1	a_1	a_2	a_3	a_4	b_{15}
R_2	b_{21}	b_{22}	b_{23}	a_4	a_5

Vì MASV có trong R_1 nên tại hàng 1 cột 1 ta có giá trị a_1
Ngược lại MASV không có trong R_2 nên tại hàng 2 cột 1 ta có giá trị b_{21}

Bước 2: sửa lại dữ liệu sao cho thỏa các phụ thuộc hàm trong F.

Bạn hãy xem bảng trên là một quan hệ (thể hiện) r nào đó của lược đồ quan hệ R. Quan hệ nầy gồm có các bộ sau:
t_1 = (a_1, a_2, a_3 , a_4 , b_{15}) và t_2 = (b_{21}, b_{22}, b_{23}, a_4 , a_5)

Bộ t_1 cho thông tin sau: sinh viên có mã số là “a_1” có tên là “a_2”, …

Vấn đề đặt ra là quan hệ r với giá trị của các bộ như trên có thỏa các phụ thuộc hàm trong F hay không ? Nếu không, ta phải sửa lại các giá trị để r thỏa F. Ta hãy xét lần lượt các phụ thuộc hàm trong F.

- Xét f1 : MASV→ TENSV,DIACHI, MALP

X → Y

Ta có: t_1.MASV = $(a_1) \neq (b_{21})$ = t_2.MASV

Vậy hai bộ là khác nhau ở vế X nên r thỏa f1.

- Xét f2 : MALP→ TENLP

X → Y

Ta có: t_1.MALP = $(a_4) = (a_4)$ = t_2.MALP (giống X)

nhưng: t_1.TENLP = $(b_{15}) \neq (a_5)$ = t_2.TENLP (khác Y)

Quan hệ trên có hai bộ có giá trị giống nhau ở các thuộc tính thuộc vế X, nhưng khác nhau ở các thuộc tính thuộc vế Y nên r vi phạm f2.

Vậy ta phải sửa lại các giá trị sao cho quan hệ trên không còn vi phạm f2. Sửa lại giá trị thuộc tính TENLP của bộ t_1 từ "b_{15}" thành "a_5". Lưu ý: chỉ sửa giá trị các thuộc tính trong vế Y và ưu tiên chọn a_j . Bảng mới như sau:

R	MASV	TENSV	DIACHI	MALP	TENLP
R_1	a_1	a_2	a_3	a_4	a_5
R_2	b_{21}	b_{22}	b_{23}	a_4	a_5

Sau khi sửa ta thấy quan hệ nầy thỏa f2.

Ta dừng thuật toán ở đây và kết luận đây là một *phân rã có nối không mất* vì ta phát hiện được một dòng (dòng thứ 1) có các giá trị gồm toàn a là (a_1, a_2, a_3, a_4, a_5).

Thí dụ III.10:

Xét lại phân rã R = SAIP thành R_1 = SA và R_2 = SIP. Các phụ thuộc là S → A và SI → P, và bảng khởi đầu là:

R	S	A	I	P
R_1	a_1	a_2	b_{13}	b_{14}
R_2	a_1	b_{22}	a_3	a_4

Bởi vì S → A là hai hàng giống nhau ở cột S, chúng ta có thể làm cho các gía trị của cột A bằng nhau, cho b_{22} thành a_2 .

Bảng kết quả là:

R	S	A	I	P
R_1	a_1	a_2	b_{13}	b_{14}
R_2	a_1	a_2	a_3	a_4

Bởi vì hàng thứ hai đều là a, đây là nối không mất.

Thí dụ III.11:

Xét phân rã R = SAIP thành R_1 = SAP và R_2 = SI. Các phụ thuộc là S → A và SI → P, và bảng khởi đầu là:

R	S	A	I	P
R_1	a_1	a_2	b_{13}	a_4
R_2	a_1	b_{22}	a_3	b_{24}

Bởi vì S → A là hai hàng giống nhau ở cột S, chúng ta có thể làm cho các gía trị của cột A bằng nhau, cho b_{22} thành a_2 .

Bảng kết quả là:

R	S	A	I	P
R_1	a_1	a_2	b_{13}	a_4
R_2	a_1	a_2	a_3	b_{24}

- Hình III.7 –

Quan hệ nầy cũng thỏa SI → P bởi vì giá trị các thuộc tính ở vế trái phụ thuộc hàm (SI) của hai bộ là khác nhau.

Ta thấy quan hệ nầy thỏa tất cả các phụ thuộc hàm, nhưng không có dòng nào gồm toàn a, nên đây *không phải là phân rã có nối không mất*.

Nhận xét:

Bạn hãy xem bảng kết quả cuối cùng Hình III.7 là một quan hệ r nào đó của lược đồ R. Quan hệ nầy gồm các bộ sau:

$t_1 = (a_1, a_2, b_{13} , a_4)$ và $t_2 = (a_1, a_2, a_3, b_{24})$

Ta có:

- r thỏa tất cả các phụ thuộc hàm
- bộ $(a_1, a_2, a_3 , a_4) \notin r$ (*)

Ta cũng có $(a_1, a_2, a_4) \in \pi_{SAP}(r)$ và $(a_1, a_3) \in \pi_{SI}(r)$ do bảng ban đầu đã xuất hiện các giá trị "a" tại cột ứng với các thuộc tính của dòng đại diện cho lược đồ con tương ứng, và giá trị "a" không thay đổi trong quá trình chạy thuật toán (chỉ sửa "b" thành "a"). Nên theo kết quả của phép nối tự nhiên ta có $(a_1, a_2, a_3, a_4) \in \pi_{SAP}(r) * \pi_{SI}(r)$ (**).

Tóm lại: do (*) và (**) ta có $\pi_{SAP}(r) * \pi_{SI}(r) \neq r$, nên theo định nghĩa đây không phải là phân rã có nối không mất.

Thí dụ III.12:

Một thí dụ khác phức tạp hơn, gọi R = ABCDE và R_1 = AD, R_2 = AB, R_3 = BE, R_4 = CDE và R_5 = AE.

Giả sử có các phụ thuộc hàm:

$$A \rightarrow C \qquad DE \rightarrow C$$
$$B \rightarrow C \qquad CE \rightarrow A$$
$$C \rightarrow D$$

Bảng khởi đầu được trình bày trong Hình III.8a. Chúng ta có thể áp dụng $A \rightarrow C$ để cho các ký hiệu b_{13}, b_{23} và b_{53} bằng nhau. Sau đó chúng ta dùng $B \rightarrow C$ để cho các ký hiệu này bằng bới b_{33} : kết quả được trình bày trong Hình III.8b, trong đó b_{13} được chọn làm ký hiệu đại diện. Bây giờ chúng ta dùng $C \rightarrow D$ để cho các ký hiệu a_4 , b_{24}, b_{34} và b_{54} bằng nhau; ký hiệu kết quả phải là a_4. Thế rồi $DE \rightarrow C$ cho phép chúng ta cho b_{13} bằng với a_3, và $CE \rightarrow A$ cho b_{31} và b_{41} bằng với a_4. Kết quả được trình bày trong Hình III.8c. Bởi vì hàng giữa đều là a nên phân rã này có tính chất nối không mất. □

Bạn đọc có thể tưởng lầm rằng chúng ta có thể đơn giản hoá Thuật toán III.1 bằng cách chỉ cho bằng nhau khi một trong các ký hiệu là a_i. Thí dụ trên cho thấy rằng nhận định như thế là không đúng; nếu chúng ta không cho b_{13} ,b_{23} và b_{53} bằng nhau, chúng ta không bao giờ thu được một hàng chỉ chứa toàn a.

a)

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{23}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{53}	b_{54}	a_5

b)

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{13}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	b_{54}	a_5

c)

R	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	a_1	a_2	a_3	a_4	a_5
R_4	a_1	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	a_3	a_4	a_5

- Hình III.8 –

Định lý III.3 :

Thuật toán III.1 xác định chính xác một phân rã có hay không có tính chất nối không mất.

Chứng minh:

Giả sử bảng cuối cùng sinh ra từ Thuật toán III.1 không có các hàng có toàn a. Chúng ta có thể xem bảng này như một quan hệ r của lược đồ R; các hàng là các bộ, và a_j , b_{ij} là các giá trị riêng biệt, mỗi giá trị được lấy từ miền của thuộc tính A_j. Quan hệ r thoả phụ thuộc F bởi vì Thuật toán III.1 sửa đổi bảng mỗi khi tìm thấy một vi phạm. Chúng ta khẳng định rằng $r \neq m_\rho(r)$. Rõ ràng r không chứa bộ a_1 ,a_2… a_n. Nhưng đối với mỗi lược đồ R_i có một bộ μ trong r, đó chính là bộ ở hàng i, sao cho $\mu_i\{R_i\}$ chứa toàn a.Vì vậy nối của các π_{R_i} (r)chứa bộ có toàn a, vì ràng bộ này giống μ_I với mọi i. Chúng ta kết luận rằng nếu bảng cuối cùng của Thuật toán III.1 không có một hàng nào chứa toàn a thì phân rã ρ không có tính chất nối không mất; chúng ta đã tìm ra một quan hệ r của R thỏa F mà $m_\rho(r) \neq r$.

Ngược lại, giả sử rằng bảng cuối cùng có một hàng toàn a. Nói chung chúng ta có thể xem một bảng T bất kỳ như là một dạng biểu diễn hình ảnh của biểu thức phép tính quan hệ miền:

$\{a_1, a_2 \dots a_n \mid (\exists b_{11}) \dots (\exists b_{kn}) (R(w_i) \wedge \dots \wedge R(w_k))\}$ (III.1)

trong đó w_i là hàng thứ i của T. Khi T là bảng khởi đầu, công thức (III.1) định nghĩa hàm m_ρ. Thật vậy, chú ý rằng $m_\rho(r)$ chứa bộ $a_1, a_2 \dots a_n$ nếu và chỉ nếu với mỗi i, r chứa một bộ có a_j ở thành phần thứ j nếu A_j là một thuộc tính của R_i và trong mỗi thuộc tính khác là một giá trị nào đó được biểu thị bằng b_{ij} .

Bởi vì chúng ta giả sử rằng mọi quan hệ r của lược đồ R đều thoả các phụ thuộc F, chúng ta có thể suy ra rằng mỗi phép biến đổi do Thuật toán III.1 thực hiện trên bảng (bằng cách đồng nhất các ký hiệu) đều được tiến hành mà không ảnh hưởng đến tập các bộ được tạo ra bởi (công thức III.1), miễn là biểu thức đó thay đổi để phản ảnh các thay đổi của bảng. Phép chúng minh khẳng định này rất phức tạp, nhưng ý tưởng thì rõ ràng; chúng ta chỉ làm đồng nhất những ký hiệu nếu trong công thức III.1, chúng được áp dụng cho một quan hệ r thoả F, những ký hiệu này chỉ được gán cùng một giá trị bằng một cách nào đó.

Bởi vì cuối cùng có một hàng chứa toàn a, biểu thức phép tính miền cho bảng này có dạng:

$\{a_1 \dots a_n \mid (\exists b_{11}) \dots (\exists b_{kn}) (R(a_1 \dots a_n) \wedge \dots)\}$ (III.2)

Rõ ràng giá trị của (công thức III.2), khi được áp dụng cho quan hệ r của R, là một tập con của r. Tuy nhiên, nếu thoả F thì giá trị của (III.2) là $m_\rho(r)$ và theo Bổ đề III.1a, $r \subseteq m_\rho(r)$. Vì thế khi r thoả F, (III.2) tính được chính r, vì thế $r = m_\rho(r)$. Điều đó nói lên rằng phân rã ρ có nối không mất ứng với F. □

Thuật toán III.1 có thể áp dụng cho các phân rã với số lượng lược đồ bất kỳ. Tuy nhiên, đối với các phân rã thành hai lược đồ, chúng ta có một phép kiểm tra đơn giản hơn, đó là nội dung của định lý sau đây.

Định lý III.4:

Nếu $\rho = (R_1, R_2)$ là một phân rã của R, và F là tập các phụ thuộc hàm, thì ρ có nối không mất ứng với F nếu và chỉ nếu $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ hoặc $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$. Chú ý rằng những phụ thuộc này không nhất thiết thuộc tập F; chỉ cần chúng thuộc F^+.

	$R_1 \cap R_2$	$R_1 - R_2$	$R_2 - R_1$
hàng cho R_1	aa….a	aa….a	bb….b
hàng cho R_2	aa….a	bb….b	aa….a

- Hình III.9 – Một bảng hai hàng tổng quát –

Chứng minh:

Bảng khởi đầu, được sử dụng trong Thuật toán III.1, được trình bày trong Hình III.9, chúng ta đã lược bỏ những chỉ số trên a và b vì không quan trọng.

Bằng phép qui nạp trên số lượng ký hiệu được xác định bằng thuật toán III.1 chúng ta có thể chứng minh rằng nếu ký hiệu b trên cột của thuộc tính A bị đổi thành a, thì A thuộc $(R_1 \cap R_2)^+$. Đồng thời cũng bằng phép qui nạp trên số các bước cần để chứng minh biểu thức $(R_1 \cap R_2) \rightarrow Y$ nhờ các tiên đề Armstrong, chúng ta cũng có thể chứng minh rằng mọi ký hiệu b trong các cột của Y được đổi thành a. Vì vậy toàn bộ hàng cho R_1 trở thành a nếu và chỉ nếu
$R_2 - R_1 \subseteq (R_1 \cap R_2)^+$ nghĩa là $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$, và tương tự, toàn bộ hàng cho R_2 trở thành a nếu và chỉ nếu $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$. □

Thí dụ III.13:

Giả sử R = ABC và F = {A $\rightarrow$ B}. Thế thì phân rã thành R_1 = AB và R_2 = AC có nối không mất bởi vì:

$R_1 \cap R_2$ = AB $\cap$ AC = A

$R_1 - R_2$ = AB – AC = B, và A $\rightarrow$ B đúng nghĩa là phụ thuộc hàm $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ thuộc F^+.

Tuy nhiên nếu chúng ta phân rã R thành R_1 = AB và R_2 = BC không phải là có nối không mất bởi vì:

$R_1 \cap R_2$ = AB $\cap$ BC = B

$R_1 - R_2$ = AB – AC = A, và B $\rightarrow$ A không đúng nghĩa là phụ thuộc hàm $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ không thuộc F^+.

$R_2 - R_1$ = AC – AB = C, và B $\rightarrow$ C không đúng nghĩa là phụ thuộc hàm $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$ cũng không thuộc F^+.

Chúng ta có thể thấy được bằng cách xét quan hệ r = $\{a_1b_1c_1, a_2b_1c_2\}$ của R. Thế thì π_{AB} (r) * π_{BC} (r) = $\{a_1b_1c_1, a_1b_1c_2, a_2b_1c_1, a_2b_1c_2\}$

Đó là một tập bao hàm thật sự của r.

II.3. Phân rã bảo toàn phụ thuộc

Chúng ta đã hiểu được rằng một phân rã cần phải có đặc tính nối không mất vì nó cho phép khôi phục lại một quan hệ ban đầu từ các hình chiếu của nó. Một đặc tính quan trọng khác của phân rã $\rho = (R_1, \ldots, R_k)$ của lược đồ quan hệ R là có thể suy ra được tập phụ thuộc F của R từ các hình chiếu của F trên các R_i.

Cho lược đồ quan hệ R và F là tập các phụ thuộc hàm. Cho phân rã $\rho = (R_1, \ldots R_k)$. Ta có các định nghĩa sau.

Định nghĩa:

Hình chiếu của F trên một tập các thuộc tính Z ký hiệu là $\pi_Z(F)$ là tập các phụ thuộc $X \rightarrow Y$ thuộc F^+ sao cho $XY \subseteq Z$ (chú ý rằng $X \rightarrow Y$ không nhất thiết thuộc F; chỉ cần thuộc F^+).

Định nghĩa:

Ta nói phân rã ρ có bảo toàn tập phụ thuộc hàm F nếu hợp của tất cả các phụ thuộc trong các hình chiếu của F trên các lược đồ con tương đương với F.

Gọi $F_i = \pi_{R_i}(F)$, với i = 1, 2, …, k

Đặt $G = F_1 \cup F_2 \cup \ldots \cup F_k = \cup_1^k F_k$

Theo định nghĩa thì: ρ có bảo toàn phụ thuộc $\Leftrightarrow G \equiv F$

Nhắc lại $G \equiv F$ (G tương đương F) nghĩa là $G^+ = F^+$.

Lý do ρ cần bảo toàn tập F đó là vì các phụ thuộc trong F có thể được xem là các ràng buộc toàn vẹn (intergrity constraint) cho lược đồ quan hệ R.

Nếu các phụ thuộc hình chiếu không suy ra được F thì khi biểu diễn R bằng $\rho=(R_1 \ldots R_k)$ chúng ta có thể thấy rằng giá trị hiện hành của các R_i có thể biểu diễn một quan hệ R không thoả F, ngay cả nếu ρ có nối không mất ứng với F.

Khi đó mỗi thao tác cập nhật trên một R_i, sẽ cần phải thực hiện một phép nối để kiểm tra lại rằng các ràng buộc không bị vi phạm.

Thí dụ III.14:

Chúng ta hãy xét Thí dụ "Sắp thời khóa biểu" trong đó chúng ta có các thuộc tính *Phòng học (P)*, *Giờ học (G)*, và *Môn học (M)*. Theo qui định thì mỗi môn học chỉ được bố trí học vào 1 phòng học duy nhất, do đó có phụ thuộc hàm M $\rightarrow$ P. Lưu ý mỗi môn học có thể học ở những giờ khác nhau. Đương nhiên với 1 phòng và 1 giờ cụ thể ta chỉ có thể sắp cho 1 môn học duy nhất, do đó có phụ thuộc hàm PG $\rightarrow$ M.

Phân rã của lược đồ quan hệ PGM thành GM và PM có nối không mất, bởi vì

(GM $\cap$ PM) $\rightarrow$ (PM – GM) nghĩa là M $\rightarrow$ P.

Tuy nhiên, chiếu của F = {PG $\rightarrow$ M , M $\rightarrow$ P} trên GM chỉ cho những phụ thuộc tầm thường (suy ra từ tính phản xạ), còn chiếu trên PM cho ra M $\rightarrow$ P và những phụ thuộc tầm thường khác. Chúng ta có thể thấy rằng M $\rightarrow$ P và các phụ thuộc tầm thường không suy ra PG $\rightarrow$ M được, vì thế phân rã này không bảo toàn các phụ thuộc.

Chẳng hạn nối hai quan hệ trong Hình III.10(a) và (b) là quan hệ của Hình III.10 (c). Hình III.10(a) thỏa các phụ thuộc tầm thường như mọi quan hệ khác. Hình III.10(b) thỏa các phụ thuộc tầm thường và phụ thuộc M $\rightarrow$ P.Tuy nhiên nối của chúng trong Hình III.10(c) vi phạm PG $\rightarrow$ M.

G	M
T.hai.7.00-9.30	CSDL
T.hai.7.00-9.30	TKCSDL

(a)

P	M
P.101	CSDL
P.101	TKCSDL

(b)

P	G	M
P.101	T.hai.7.00-9.30	CSDL
P.101	T.hai.7.00-9.30	TKCSDL

(c)

- Hình III.10 – Một nối vi phạm phụ thuộc hàm –

Nhận xét:

Giả sử chúng ta sử dụng hai lược đồ con PG và PM để lưu trữ thông tin thay vì sử dụng lược đồ ban đầu PGM.

Vì phân rã nầy là không bảo toàn phụ thuộc (bị mất phụ thuộc hàm) nên các lược đồ con sẽ không chứa một phụ thuộc hàm nào đó, ở đây là PG → M bị mất.

Khi nhập dữ liệu vào các lược đồ con ta không thể kiểm tra được phụ thuộc hàm PG → M mà chỉ kiểm tra được phụ thuộc hàm M → P. Do đó khi ta khôi phục lại quan hệ ban đầu bằng cách lấy nối tự nhiên của hai quan hệ con, ta thấy quan hệ nầy có thể vi phạm phụ thuộc PG → M.

Để có thể kiểm tra được phụ thuộc PG → M, mỗi khi có thao tác nhập liệu trên các quan hệ con, *ta phải lấy nối tự nhiên để tạo ra quan hệ ban đầu* có đầy đủ các thuộc tính. Quan hệ có đủ các thuộc tính nầy chắc chắn chứa phụ thuộc PG → M, nên ta có thể kiểm tra dữ liệu có thỏa phụ thuộc nầy hay không.

Với một phép phân rã có bảo toàn phụ thuộc thì ta *không cần thực hiện phép nối* để kiểm tra các phụ thuộc trong F khi có thao tác nhập liệu trên các quan hệ con. Bởi vì các phụ thuộc hàm được chứa trên các lược đồ con thì *tương đương* với tập phụ thuộc hàm ban đầu F.

Do đó khi có thao tác nhập liệu vào một quan hệ con của lược đồ R_i nào đó ta chỉ cần kiểm tra các phụ thuộc trong hình chiếu của F trên R_i là đủ, đó là F_i . Khi đó kết quả nối tự nhiên của các quan hệ con đương nhiên sẽ thỏa tất cá các phụ thuộc hàm trong F vì hợp các F_i sẽ tương đương với F.

Thí dụ III.15:

Chúng ta nên nhớ rằng một phân rã có thể có nối không mất ứng với một tập các phụ thuộc F nhưng không bảo toàn tập phụ thuộc hàm F như ví dụ trên.

Sau đây ta xét một phân rã bảo toàn phụ thuộc nhưng lại không có tính chất nối không mất. Chẳng hạn như trường hợp $F = \{A \rightarrow B, C \rightarrow D\}$, R = ABCD, và $\rho=\{AB,CD\}$.

Phép phân rã nầy có hai lược đồ con, để kiểm tra tính nối không mất ta áp dụng định lý III.4.

Vì $R_1 \cap R_2 = AB \cap CD = \Phi$ nên không có phụ thuộc hàm dạng $R_1 \cap R_2 \rightarrow R_1 - R_2$ hay $R_1 \cap R_2 \rightarrow R_2 - R_1$. Do đó phân rã nầy không có đặc tính nối không mất.

Để kiểm tra tính bảo toàn phụ thuộc ta có:

$F_1 = \pi_{R1} (F) = \pi_{AB} (F) = \{ A \rightarrow B \}$ và

$F_2 = \pi_{R2} (F) = \pi_{CD} (F) = \{ C \rightarrow D \}$

Nên $G = F_1 \cup F_2 = \{ A \rightarrow B, C \rightarrow D \}$

Theo định nghĩa thì phép phân rã trên có bảo toàn phụ thuộc vì $G \equiv F$

II.3.1. Kiểm tra tính bảo toàn phụ thuộc

Về nguyên tắc chúng ta có thể kiểm tra xem một phân rã $\rho = (R_1,\ldots,R_k)$ có bảo toàn tập phụ thuộc F hay không. Chúng ta chỉ cần tính F^+ rồi chiếu nó trên tất cả các thành phần R_i. Sau đó lấy hợp của các tập phụ thuộc kết quả rồi kiểm tra xem tập này có tương đương với F hay không.

Tuy nhiên trong thực tế, tính F^+ là một công việc hết sức khó khăn vì số lượng các phụ thuộc chứa trong nó thường là hàm mũ theo kích thước của F. Nhưng có một cách để kiểm tra tính bảo toàn này mà không cần phải tính F^+ ; phương pháp này có chi phí thời gian tỷ lệ với hàm đa thức theo kích thước của F.

Thuật toán III.2:

Kiểm tra tính bảo toàn các phụ thuộc.

NHẬP:

Một phân rã $\rho = (R_1,\ldots,R_k)$ và tập các phụ thuộc F.

XUẤT:

Một khẳng định là ρ có bảo toàn F hay không.

PHƯƠNG PHÁP:

Chúng ta gọi G là $\cup^k_{i=1} \pi_{Ri}$ (F). Chú ý rằng chúng ta không tính G; đơn giản chúng ta muốn xem nó có tương đương F hay không.

Nhắc lại: do bổ đề II.4 ta có

$$G \equiv F \Leftrightarrow$$

(i) $G \subseteq F^+$ và

(ii) $F \subseteq G^+$

Điều kiện (i) là hiển nhiên là do định nghĩa của phép chiếu F trên các R_i .

Do đó để kiểm tra xem G có tương đương F hay không, ta chỉ cần kiểm tra điều kiện (ii).

Để kiểm tra (ii), ta phải xét mỗi phụ thuộc $X \rightarrow Y$ trong F và xác định xem $X \rightarrow Y$ có thuộc G^+ hay không. Tương đương với việc tính X^+ (bao đóng của X) đối với các phụ thuộc hàm trong G, sau đó xét X^+ có chứa Y hay không.

Thủ thuật để tính X^+ mà không cần có G là xét lặp đi lặp lại kết quả tính bao đóng X^+ ứng với các hình chiếu của F trên các R_i.

Chúng ta định nghĩa phép toán R trên các thuộc tính Z ứng với một tập phụ thuộc F là phép thế

$Z = Z \cup (Z \cap R)^+ \cap R)$, bao đóng được lấy ứng với F.

Phép toán này nối Z với những thuộc tính A sao cho $(Z \cap R) \rightarrow A$ thuộc $\pi_R(F)$.

Do đó chúng ta tính X^+ ứng với G bằng cách khởi đầu với X, qua danh sách các R_i, ta lần lượt thực hiện các phép toán R_i với mỗi i. Nếu tại một vòng nào đó không có phép toán R_i nào làm thay đổi các tập thuộc tính hiện có thì chúng ta đã thực hiện xong: tập kết quả là X^+.

Về hình thức, thuật toán được viết là:

```
Z := X
While " vẫn còn thay đổi với Z " do
    For i := 1 to k do
        Z := Z ∪ (( Z ∩ Ri)+ ∩ Ri ) /* bao đóng được
        lấy ứng với F */
```

Nếu Y là một tập con của Z, là kết quả thực hiện các bước trên, thì $X \rightarrow Y$ thuộc G^+. Nếu mỗi $X \rightarrow Y$ thuộc F đều thuộc G^+, thuật toán trả lời "yes", ngược lại thuật toán trả lời "no". □

Thí dụ III.16:

Xét tập các thuộc tính ABCD với phân rã {AB, BC, CD} và tập các phụ thuộc F= {$A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D\pi$, $D \rightarrow A$}. Nghĩa là trong F^+, mỗi thuộc tính đều xác định tất cả các thuộc tính còn lại.

Lúc đầu chúng ta không nhận được phụ thuộc $D \rightarrow A$, nhưng điều đó không đúng. Khi chiếu F, thực sự chúng ta đã chiếu F^+ trên các lược đồ quan hệ, vì thế chiếu trên AB chúng ta không chỉ thu được $A \rightarrow B$ mà còn thu được $B \rightarrow A$.

Tương tự, chúng ta có được $C \rightarrow B$ thuộc $\pi_{BC}(F)$ và $D \rightarrow C$ thuộc $\pi_{CD}(F)$, và ba phụ thuộc này suy ra được $D \rightarrow A$.

Vì thế chúng ta có thể hy vọng rằng thuật toán 7.3 sẽ cho phép khẳng định rằng $D \rightarrow A$ được suy ra từ:
$G = \pi_B(F) \cup \pi_{BC}(F) \cup \pi_{CD}(F)$

Chúng ta khởi đầu với $Z = \{D\}$. Áp dụng phép toán AB không thêm được thuộc tính nào bởi vì

$\{D\} \cup ((\{D\} \cap \{A,B\}^+ \cap \{A,B\})$

cũng chỉ là $\{D\}$.

Tương tự, phép toán BC cũng không làm thay đổi Z.

Tuy nhiên khi chúng ta áp dụng phép toán CD thì thu được

$$Z = \{D\} \cup ((\{D\} \cap \{CD\}^+ \cap \{C,D\})$$
$$= \{D\} \cup (\{D\}^+ \cap \{C,D\})$$
$$= \{D\} \cup (\{A,B,C,D\} \cap \{C,D\})$$
$$= \{C,D\}$$

Tương tự ở vòng kế phép toán BC được áp dụng cho $Z = \{C,D\}$ cho ra $Z = \{B,C,D\}$, và ở dòng thứ ba, phép toán AB đặt Z thành $\{A,B,C,D\}$,sau đó không thay đổi Z được nữa.

Vì thế ứng với G, $\{D\}^+ = \{A,B,C,D\}$có chứa A nên chúng ta kết luận rằng $D \rightarrow A$ thuộc G^+.

Bạn hãy kiểm tra rằng các phần tử khác của F cũng thuộc G^+ (thực sự chúng thuộc G). Do đó chúng ta kết luận rằng phân rã này bảo toàn được tập phụ thuộc F. □

Định lý III.5 :

Thuật toán III.2 xác định chính xác $X \rightarrow Y$ có thuộc G^+ hay không.

Chứng minh:

Mỗi lần chúng ta thêm một thuộc tính vào Z, chúng ta đều sử dụng một phụ thuộc trong G, vì thế khi thuật toán trả lời "yes", nó phải đúng nghĩa là $X \rightarrow Y$ là hệ quả của G.

Ngược lại, giả sử $X \rightarrow Y$ thuộc G^+. Thế thì có một chuỗi các bước mà qua đó, bằng cách dùng Thuật toán tính bao đóng để lấy bao đóng của X ứng với G, cuối cùng chúng ta sẽ gom tụ được tất cả các thuộc tính của Y. Mỗi bước này đều có sử dụng một phụ thuộc trong G, và phụ thuộc đó phải thuộc $\pi_{R_i}(F)$ với một trị số i nào đó. Bởi vì G là hợp của chúng. Gọi một phụ thuộc như thế là $U \rightarrow V$. Dùng phép quy nạp trên số các phụ thuộc được áp dụng trong Thuật toán bao đóng để chứng minh rằng cuối cùng thi U trở thành một tập con của Z, và trong vòng kế tiếp, phép toán R, chắc chắn sẽ thêm vào Z tất cả các thuộc tính của V nếu chúng chưa có trong đó. □

II.4. Phân rã có nối không mất thành dạng BCNF

Cho đến giờ chúng ta đã giới thiệu qua những đặc tính cần có đối với các *lược đồ quan hệ*: BCNF hoặc yếu hơn là 3NF.

Trong phần II.2 và II.3 chúng ta thấy hai đặc tính tổng quát quan trọng nhất của các *lược đồ cơ sở dữ liệu*, đặc tính nối không mất và đặc tính bảo toàn phụ thuộc.

Bây giờ sẽ kết hợp những ý tưởng này lại với nhau, nghĩa là xây dựng những lược đồ cơ sở dữ liệu có những đặc tính mong muốn cho lược đồ cơ sở dữ liệu và đối với mỗi lược đồ quan hệ cũng có những đặc tính mong muốn cho lược đồ quan hệ.

Chúng ta sẽ nhận thấy rằng mọi lược đồ quan hệ đều có một phân rã không mất thành dạng chuẩn Boyce-Codd và nó cũng có thể phân rã thành dạng 3NF có nối không mất và vẫn bảo toàn phụ thuộc. Tuy nhiên có lẽ không có một phân rã nào cho một lược đồ quan hệ thành dạng Boyce-Codd mà vẫn bảo toàn được phụ thuộc.

Lược đồ quan hệ CSZ là một thí dụ điển hình. Nó không có dạng Boyce-Codd bởi vì phụ thuộc $Z \rightarrow C$ đúng, nhưng nếu chúng ta phân rã CSZ theo một cách sao cho CSZ không phải là một trong những lược đồ trong phân rã này thì phụ thuộc $CS \rightarrow Z$ không suy ra được từ các phụ thuộc hình chiếu.

Trước khi đưa ra thuật toán phân rã, chúng ta cần đến đặc tính của các phân rã nối không mất như dưới đây:

Bổ đề III.2:

Giả sử R là một lược đồ quan hệ với các phụ thuộc hàm F. Gọi $\rho = (R_1,\ldots,R_n)$ là một phân rã của R có nối không mất ứng với F, và gọi $\rho = (S_1,S_2)$ là một phân rã có nối không mất của R_1 ứng với $\pi_{R1}(F)$. Thế thì phân rã của R thành $(S_1,S_2,R_2,\ldots,R_n)$ cũng có nối không mất ứng với F.

Chứng minh:

Giả sử chúng ta lấy quan hệ r của R rồi chiếu nó trên $R_1,\ldots,R_n$ để thu được các quan hệ tương ứng là $r_1,\ldots r_n$. Sau đó lại chiếu r_1 trên S_1và S_2 để có s_1,s_2. Đặc tính nối không mất cho phép nối s_1,và s_2 để khôi phục chính xác r_1 rồi chúng ta có thể nối r_1, $r_2,\ldots$, r_n để khôi phục lại r. Bởi vì nối tự nhiên có tính kết hợp và thứ tự trong đó chúng ta thực hiện phép nối không là vấn đề quan trọng , vì thế chúng ta có thể khôi phục lại r mà không phụ thuộc vào thứ tự chúng ta lấy các nối của $s_1,s_2,r_1,\ldots r_n$. □

Chúng ta có thể sử dụng Bổ đề III.2 để xây dựng Thuật toán phân rã một lược đồ quan hệ R thành BCNF, tuy đơn giản nhưng tốn khá nhiều thời gian.

Nếu chúng ta tìm thấy một vi phạm BCNF trong F, gọi là $X \rightarrow A$, chúng ta phân rã R thành các lược đồ R – A và XA. Cả hai đều nhỏ hơn R, bởi vì XA không thể là tất cả các thuộc tính của R (nếu thế thì X chắc chắn là khoá bao hàm, nên $X \rightarrow A$ sẽ không vi phạm BCNF).

Theo định lý III.4, phân rã R – A và XA có nối không mất, bởi vì giao của hai lược đồ là X, và $X \rightarrow XA$. Chúng ta tính chiếu của các phụ thuộc của R trên R – A và XA rồi lại áp dụng bước phân rã trên cho các lược đồ. Bổ đề III.2 khẳng định rằng tập lược đồ thu được bằng cách phân rã cho đến khi tất cả các lược đồ đều có dạng BCNF sẽ là một phân rã không mất.

Vấn đề là phép chiếu các phụ thuộc có thể có chi phí thời gian tỷ lệ hàm mà theo kích thước của lược đồ R và tập phụ thuộc ban đầu. Tuy nhiên, chúng ta cũng có một cách để tìm một phân rã nối không mất thành những lược đồ có dạng BCNF với chi phí thời gian tỷ lệ hàm đa thức theo kích thước của tập phụ thuộc và lược đồ quan hệ. Có điều kỹ thuật này đôi khi lại phân rã một quan hệ đã có dạng BCNF. Bổ đề tiếp theo đưa ra một số đặc tính khác của các lược đồ BCNF.

Bổ đề III.3:

a) Mỗi lược đồ có hai thuộc tính đều có dạng BCNF.

b) Nếu R không có dạng BCNF thì chúng ta có thể tìm được các thuộc tính A và B trong R sao cho $(R – AB) \rightarrow A$. Phụ thuộc $(R – AB) \rightarrow B$ có thể cũng đúng trong trường hợp này.

Chứng minh:

Đối với trường hợp (a), gọi AB là một lược đồ. Ở lược đồ này chỉ có hai phụ thuộc không tầm thường có thể đúng : A → B và B → A. Nếu không có phụ thuộc nào đúng thì chắc chắn không có vi phạm BCNF. Nếu chỉ A → B đúng thì A là một khoá, vì thế chúng ta không có vi phạm nào. Nếu chỉ B → A đúng thì B là khoá, vì nếu cả hai đúng, cả A và B đều là khoá, vì thế không bao giờ có vi phạm BCNF.

Đối với trường hợp (b), giả sử có một vi phạm X → A trong R. Thế thì R phải có một thuộc tính B không thuộc XA, nếu không thì X là một khoá bao hàm, và X → A không phải là vi phạm. Do đó (R – AB) → A chính là điều phải chứng minh. □

Bổ đề III.3 cho chúng ta tìm các vi phạm BCNF trong một lược đồ quan hệ R có n thuộc tính bằng cách chỉ xét n(n1)/2 cặp thuộc tính {A,B} và tính bao đóng của R – AB ứng với tập phụ thuộc F đã cho. Như đã khẳng định, thuật toán đó thực hiện với thời gian $O(n^2)$, nhưng một dữ liệu được thiết kế cẩn thận có thể làm cho nó chạy nhanh hơn, với thời gian là O(n); dù sao, chi phí thời gian cũng là hàm da thức theo kích thước của R. Nếu không tồn tại A hoặc B sao cho $(R- AB)^+$ chứa A hoặc B thì theo Bổ đề III.3(b), chúng ta biết rằng R có dạng BCNF.

Điều quan trọng cần phải biết rằng đảo lại sẽ không đúng. Có thể R có dạng BCNF nhưng vẫn có một cặp {A,B} như trên. Chẳng hạn nếu R = ABC, và F = {C → A, C → B} thì R có dạng BCNF, nhưng R – AB = C, và xác định được A (và cả B).

Trước khi phân tích thuật toán phân rã BCNF, chúng ta cần có một ghi nhận nữa về hình chiếu của các phụ thuộc, cụ thể là:

Bổ đề III.4:

Nếu chúng ta có một tập phụ thuộc F trên R và chúng ta chiếu các phụ thuộc trên $R_1 \subseteq R$ để được F_1 rồi lại chiếu F_1 trên $R_2 \subseteq R_1$ để được F_2 thì

$$F_2 = \pi_{R2}(F)$$

Nghĩa là chúng ta có thể giả sử rằng F là tập phụ thuộc của R_1, dù rằng F có thể có những thuộc tính không có trong R_1.

Chứng minh:

Nếu $XY \subseteq R_2$ thì $X \rightarrow Y$ thuộc F^+ nếu và chỉ nếu nó thuộc F_1^+. □

Bổ đề III.4 có môt hệ quả hết sức quan trọng. Đó là , nếu chúng ta phân rã các lược đồ quan hệ như trong Bổ đề III.2 thì thật sự chúng ta không bao giờ phải tính những phụ thuộc hình chiếu trong quá trình phân rã. Nghĩa là chúng ta chỉ sử dụng các phụ thuộc đã cho, lấy các bao đóng của tập thuộc tính khi cần thiết chứ không phải tính tất cả các hình chiếu của các phụ thuộc, là thuật toán có chi phí hàm mũ theo số lượng thuộc tính trong lược đồ. Chính nhận xét này và bổ đề III.3(b) cho phép chúng ta thực hiện , với chi phí thời gian là hàm đa thức theo kích thước của lược đồ và các phụ thuộc đã cho, nhằm tìm ra được một phân rã thành dạng BCNF có nối không mất cho lược đồ.

Thuật toán III.3:

Phân rã nối không mất thành dạng chuẩn Boyce-Codd.

NHẬP:

Lược đồ quan hệ R và các phụ thuộc hàm F.

XUẤT:

Một phân rã của R có nối không mất, sao cho mỗi lược đồ quan hệ trong phân rã có dạng Boyce-Codd ứng với hình chiếu của F trên lược đồ đó.

PHƯƠNG PHÁP:

Trọng tâm của thuật toán là lấy lược đồ quan hệ R rồi phân rã nó thành hai lược đồ. Một lược đồ có tập các thuộc tính XA; nó có dạng BCNF và phụ thuộc $X \to A$ đúng. Lược đồ thứ hai sẽ là R – A, do đó nối của R – A với XA là nối không mất.

Thực hiện đệ qui thủ tục phân rã, với R – A ở vị trí của R cho đến khi chúng ta đạt được một lược đồ đáp ứng được các điều kiện của Bổ đề III.3(b); chúng ta biết rằng lược đồ này có dạng BCNF. Thế thì bổ đề III.2 bảo đảm rằng lược đồ này cộng với các lược đồ BCNF được tạo ra từ mỗi bước đệ qui có nối không mất.

```
Z:= Ri  /* bất kỳ lúc nào Z cũng là một lược đồ của phân
           rã mà có thể không có dạng  BCNF */
Repeat
   Phân rã Z thành Z – A và XA, trong đó XA có dạng
   BCNF và X → A;
    /* sử dụng thủ tục con của Hình 7,6(b) */
    thêm XA vào phân rã;
     Z := Z - Ai
Until không thể phân rã Z bằng Bổ đề 7.7(b);
    Thêm  Z vào trong phân rã
```

(a) Chương trình chính.

```
If   không chứa A và B sao cho A thuộc (Z – AB)+
               then
     cần nhớ rằng tất cả các bao đóng được lấy ứng
               với F */
     return thông báo rằng Z có dạng BCNF và
               không phân rã được.
else begin
     tìm một cặp A và B
     Y := Z;
     While Y chứa A và B sao cho (Y- AB) → A do
     Y := Y – B;
    Return phân rã Z – A và Y;
     /* ở đây Y là XA trong chương trình chính */
               end
```

(b) Thủ tục phân rã

- Hình III.11 – Chi tiết của thuật toán III.3 –

Chi tiết Thuật toán được trình bày trong Hình III.11. Hình III.11 (a) là thủ tục chính, nó phân rã lặp đi lặp lại lược đồ quan hệ Z chưa biết là có dạng BCNF hay không; khởi đầu Z là R. Hình III.11 (b) là thủ tục phân rã, nó xác định không thể phân rã được Z hoặc phân rã Z thành Z – A và XA trong đó X → A. Tập thuộc tính được chọn bằng cách khởi đầu với Y = Z, loại bỏ B lặp đi lặp lại, là thuộc tính trong cặp AB sao cho X → A, trong đó X = Y – AB. Cần nhớ rằng X → B đúng hay sai là điều không quan trọng. □

Thí dụ III.17:

Chúng ta hãy xét lược đồ quan hệ CTHRSG, với:

C = course (lớp, khoá học)
T = teacher (giảng viên)
H = hour (giờ học)
R = room (phòng học)
S = student (sinh viên)
G = grade (điểm số).

Các phụ thuộc hàm F chúng ta giả sử tồn tại là:

$C \rightarrow T$ Mỗi khoá có một giảng viên

$HR \rightarrow C$ Tại một giờ học, một phòng chỉ có một lớp

$HT \rightarrow R$ Tại một giờ học, một giảng viên chỉ có mặt tại trong một phòng học

$CS \rightarrow G$ Mỗi sinh viên có một điểm số cho mỗi khoá.

$HS \rightarrow R$ Tại một giờ học, một sinh viên chỉ có mặt tại một phòng học.

Bởi vì thuật toán III.3 không xác định thứ tự xét cặp AB, chúng ta sẽ đưa ra một chiến lược thống nhất theo thứ tự CTHRSG cho các thuộc tính và sử dụng lần lượt thuộc tính đầu tiên, rồi đến thuộc tính thứ hai, rồi thứ ba, v.v…

Chúng ta bắt đầu với toàn bộ lược đồ CTHRSG, và cặp đầu tiên được xét đến là CT.

Chúng ta thấy rằng $(HRSG)^+$ chứa C; nó cũng chứa T, nhưng không quan trọng. Do đó chúng ta bắt đầu vòng lặp while của Hình III.11 (b) với A = C, B = T, và Y = CHRSG.

Bây giờ chúng ta thử cặp CH vào vai trò {A,B} nhưng $(RSG)^+$ không chứa C lẫn H. chúng ta gặp may nhờ cặp kế tiếp là CR, bởi vì $(HSG)^+$ chứa R. Vì vậy chúng ta có A = R, B = C, và chúng ta đặt Y là HRSG, bằng cách bỏ đi B như thường lệ.

Với Y = HRSG chúng ta không gặp may cho đến khi chúng ta thử cặp RG, khi đó chúng ta tìm thấy $(HS)^+$ chứa R. Vì vậy chúng ta có A = R và B = G, và Y được gán cho HRS.

Tại điểm này, chúng ta không loại được thuộc tính nào khác ra khỏi Y, bởi vì phép kiểm tra của Bổ đề III.3(b) không thành công trên mỗi cặp trong HRS. Do vậy chúng ta có thể phân rã CTHRSG thành

1. HRS, có vai trò của XA, với X = HS và A = R
2. Z = CTHRSG – R, đó là CTHSG

Bây giờ chúng ta thực hiện trên Z = CTHSG trong chương trình chính. Danh sách các cặp AB và các tập thuộc tính còn lại sau khi đã loại bỏ B là:

1. Trong CTHSG: A = T, B = H, để lại Y = CTSG
2. Trong CTSG: A = T, B = S, để lại Y = CTG
3. Trong CTG: A = T, B = G, để lại Y = CT

Chắc chắn rằng CT có dạng BCNF do Bổ đề III.3 (a). Vì thế chúng ta thêm CT vào phân rã.

Thuộc tính T đóng vai trò của A, vì thế trong chương trình chính chúng ta loại T đi và chuyển sang lược đồ Z = CHSG, là một lược đồ không có dạng Boyce-Codd.

Trong CHSG, cặp thành công đầu tiên là A = G và B = H, để lại Y = CSG. Chúng ta không tìm được cặp nào cho phép phân rã lược đồ này theo Bổ đề III.3 (b), vì thế chúng ta thêm CSG vào trong phân rã, rồi áp dụng chương trình chính cho lược đồ đã loại bỏ A, nghĩa là Z = CHS.

Chúng ta thấy rằng lược đồ này không thể phân rã được nhờ Bổ đề III.3 (b), do vậy nó cũng thuộc dạng BCNF, và chúng ta kết thúc.

Chú ý rằng chúng ta thu được các nối không mất tại mỗi bước nếu chúng ta tổ hợp các lược đồ theo thứ tự ngược lại với thứ tự tìm thấy chúng. Nghĩa là:

CHS * CSG là nối không mất nhờ phụ thuộc CS $\rightarrow$ G

CHSG * CT là nối không mất nhờ phụ thuộc C $\rightarrow$ T, và

CTHSG * HRS là nối không bị mất nhờ HS $\rightarrow$ R.

Trong mỗi trường hợp trên, phụ thuộc hàm cần phải có là phụ thuộc có dạng X $\rightarrow$ A đã được xây dựng bởi thủ tục của Hình III.11 (b). Theo Bổ đề III.2, các nối này bảo đảm rằng phân rã đầy đủ (HRS, CT, CSG, CHS) có nối không mất. □

CÁC VẤN ĐỀ NẢY SINH TRONG CÁC PHÂN RÃ BCNF "TUỲ TIỆN"

Trong phân rã của Thí dụ III.17 bốn lược đồ quan hệ trên lưu lại các loại thông tin sau:

1. Vị trí (phòng học) của mỗi sinh viên tại mỗi giờ học (lược đồ HRS).
2. Giảng viên của mỗi lớp (lược đồ CT)
3. Điểm cho khoá học của sinh viên (lược đồ CSG)
4. Thời khoá biểu các khoá học và giờ học cho mỗi sinh viên (lược đồ CHS)

Các thông tin thu được không như kết quả thiết kế như khi chúng ta thực hiện phân rã nối không mất thành BCNF theo cách thủ công.

Cụ thể là chúng ta không thể biết phòng học cho một khoá học nếu không nối các quan hệ CHS và HRS lại, và thậm chí không thể tìm ra phòng nếu không có sinh viên nào lấy khoá học đó. Có thể chúng ta sẽ thay HRS bằng CHR. Nó cho phép phân bổ các phòng học theo khoá học chứ không phải theo sinh viên, và phù hợp với các thời khoá biểu được sử dụng trong nhiều trường. Không may là, câu hỏi về "giá trị" của các phân rã khác nhau không phải là mục tiêu của lý thuyết này. Nếu không có sẵn một lược đồ cụ thể mà chỉ cần xác nhận lại rằng nó có một nối không mất và mỗi lược đồ thành phần có dạng BCNF, thế thì chúng ta có thể chọn cặp AB ngẫu nhiên trong Thuật toán III.3 với hy vọng rằng sau một vài lần chọn thử như thế, chúng ta sẽ thu được một phân rã có vẻ "tự nhiên".

Một vấn đề khác liên quan đến phân rã được chọn là một số phụ thuộc trong F, cụ thể là TH → R và HR → C không được bảo toàn trong phân rã này. Nghĩa là, hình chiếu của F trên HRS, CT, CSG và CHS là bao đóng của các phụ thuộc sau (bạn hãy kiểm chứng được điều này).

$$CS \to G \qquad HS \to R$$
$$C \to T \qquad HS \to C$$

Chú ý rằng phụ thuộc cuối cùng, HS → C, thuộc hình chiếu của F trên CHS nhưng không phải là một phụ thuộc cho trước; ba phần tử còn lại là của chính F. Bốn phụ thuộc này không kéo theo TH → R hoặc HR → C. Chẳng hạn quan hệ của CTHRSG được trình bày dưới đây không thoả TH → R lẫn HR → C, nhưng hình chiếu của nó trên HRS, CT, CSG và CHS thoả tất cả các phụ thuộc được chiếu.

C	T	H	R	S	G
c_1	t	h	r_1	s_1	g_1
c_2	t	h	r_2	s_2	g_2
c_1	t	h	r_1	s_3	g_3

HIỆU QUẢ CỦA PHÂN RÃ BCNF

Chúng ta thấy rằng Thuật toán III.3 có chi phí thời gian là một hàm đa thức theo n, là chiều dài của lược đồ quan hệ R và các phụ thuộc F.

Chúng ta cũng đã nhận xét rằng thuật toán tính bao đóng ứng với F có chi phí thời gian là hàm đa thức theo n; sự thực chỉ cần O(n) nếu sử dụng các cấu trúc dữ liệu đặc biệt.

Thủ tục của Hình III.11 (b) chạy trên một tập con Z của các thuộc tính, mà chắc chắn không thể có nhiều hơn n thuộc tính. Mỗi lần chạy qua vòng lặp. tập Y giảm kích thước, vì thế chỉ có tối đa n vòng lặp. Cũng chỉ có tối đa n^2 cặp thuộc tính A và B, vì thế phép kiểm tra (Y – AB) → A được thực hiện tối đa n^3 lần. Bởi vì phép kiểm tra này có chi phí theo hàm đa thức, và thời gian đó chiếm ưu thế trong thân vòng lặp, chúng ta kết luận rằng Thuật toán của Hình III.11 (b) thực hiện với thời gian theo hàm đa thức.

Chi phí chủ yếu của chương trình chính của Hình III.11 (a) là gọi thủ tục con, và lời gọi này được thực hiện chỉ một lần cho mỗi lần thực hiện vòng lặp. Bởi vì Z giảm kích thước khi đi qua vòng lặp, tối đa có thể n vòng, vì thế toàn bộ thuật toán có chi phí theo hàm đa thức

II.5. Phân rã thành 3NF có bảo toàn phụ thuộc

Từ các Thí dụ III.6 và III.7, chúng ta đã thấy là không phải lúc nào cũng có thể phân rã một lược đồ thành dạng BCNF mà vẫn bảo toàn được các phụ thuộc. Tuy nhiên chúng ta luôn có thể tìm được một phân rã bảo toàn phụ thuộc thành dạng chuẩn cấp ba như thuật toán và định lý sau sẽ chứng minh.

Thuật toán III.4:
Phân rã bảo toàn phụ thuộc thành dạng chuẩn cấp ba.

NHẬP:
Lược đồ quan hệ R và tập phụ thuộc F, chúng ta có thể giả sử rằng *F là phủ cực tiểu* mà không mất đi tính tông quát.

XUẤT:
Một phân rã bảo toàn phụ thuộc của R sao cho mỗi lược đồ quan hệ đều có dạng 3NF ứng với hình chiếu của F trên lược đồ đó.

PHƯƠNG PHÁP:
Nếu có những thuộc tính của R không nằm trong một phụ thuộc nào của F dù ở vế phải hay trái, thì về nguyên tắc, mỗi thuộc tính nhưng thế đều có thể tạo ra một lược đồ và chúng ta có thể loại chúng ra khỏi R.

Nếu một trong những phụ thuộc trong F có chứa tất cả các thuộc tính của R thì kết xuất chính là R. Ngược lại, phân rã kết xuất ρ sẽ chứa lược đồ XA cho mỗi phụ thuộc X → A trong F. □

Thí dụ III.18:

Xét lại lược đồ CTHRSG của Thí dụ III.17, các phụ thuộc của nó có phủ cực tiểu F:

$C \rightarrow T$	$CS \rightarrow G$
$HR \rightarrow C$	$HS \rightarrow R$
$HT \rightarrow R$	

Thuật toán III.4 sinh ra tập lược đồ quan hệ CT, CHR, THR, CSG và HRS. □

Định lý III.6:

Thuật toán III.4 tạo một phân rã bảo toàn phụ thuộc thành dạng chuẩn cấp ba.

Chứng minh:

Bởi vì các phụ thuộc hình chiếu có chứa phủ của F, phân rã rõ ràng là bảo toàn phụ thuộc.

Chúng ta phải chứng minh rằng lược đồ quan hệ YB, đối với mỗi phụ thuộc hàm Y → B trong phủ cực tiểu, đều có dạng 3NF.

Giả sử X → A vi phạm 3NF trong YB; nghĩa là A không thuộc X, X không phải là khoá bao hàm của YB, và A làthuộc tính không tham gia khóa. Dĩ nhiên, chúng ta cũng biết rằng XA ⊆ YB, và X → A được suy ra từ F. Chúng ta xét hai trường hợp, tuỳ vào A = B hay không.

Trường hợp 1: A = B. Bởi vì A không thuộc X, chúng ta biết $X \subseteq Y$, và bởi vì X không phải là khoá bao hàm của YB, X phải là một tập con thật sự của Y. nhưng thế thì $X \rightarrow B$, và cũng là $X \rightarrow A$, có thể thay thế $Y \rightarrow B$ trong phủ cực tiểu, mâu thuẫn với giả thiết rằng $Y \rightarrow B$ là thành phần của phủ cực tiểu đã cho.

Trường hợp 2: $A \neq B$. Bởi vì Y là khoá bao hàm của YB, nên phải tồn tại một tập $Z \subseteq Y$ là khoá của YB. Nhưng A thuộc Y, bởi vì chúng ta đang giả sử rằng $A \neq B$, và A không thể thuộc Z vì A là phi nguyên tố. Do vậy Z là một tập con thực sự của Y, nên $Z \rightarrow B$ có thể thay thế $Y \rightarrow B$ trong phủ cực tiểu, cũng dẫn đến mâu thuẫn. □

Chúng ta cũng có thể hiệu chỉnh Thuật toán III.4 nhằm tránh các phân rã không cần thiết nếu $X \rightarrow A_1,\ldots,X \rightarrow A_n$ ở vị trí của n lược đồ quan hệ $XA_1,\ldots,XA_n$. Phần này dành cho bạn đọc tự chứng minh rằng lược đồ $XA_1\ldots A_n$ có dạng 3NF.

II.6. Phân rã thành 3NF bảo toàn phụ thuộc và có nối không mất

Như đã phân tích, chúng ta có thể dùng Thuật toán III.3 để phân rã một lược đồ quan hệ R bất kỳ thành một tập các lược đồ $\rho = (R_1,\ldots, R_k)$ sao cho ρ có nối không mất và mỗi R_i có dạng BCNF (vì thế có dạng 3NF).

Hơn nữa, chúng ta cũng có thể dùng Thuật toán III.4 để phân rã R thành $\sigma = (S_1,\ldots, S_m)$ sao cho σ bảo toàn tập phụ thuộc F, và mỗi S_j có dạng 3NF.

Liệu chúng ta có thể tìm được một phân rã thành 3NF mà vẫn có hai đặc tính bảo toàn phụ thuộc và nối không mất? Thật ra chúng ta có thể dùng Thuật toán III.4 để phân rã R thành σ, sau đó chỉ cần nối vào σ một lược đồ quan hệ X, là khoá của R như định lý sau sẽ trình bày.

Định lý III.7:

Gọi σ là một phân rã dạng 3NF của R được xây dựng bằng Thuật toán III.4, và X là khóa của R.

Thế thì $\tau = \sigma \cup \{X\}$ là một phân rã của R mà tất cả các lược đồ quan hệ đều có dạng 3NF; phân rã này có đặc tính bảo toàn phụ thuộc và nối không mất.

Chứng minh:

Có thể chứng minh được rằng bất kỳ vi phạm của 3NF nào trong X cũng khẳng định rằng một tập con thực sự của X xác định được X, và như thế xác định được R, do vậy X không thể là khoá trong trường hợp này. Vì thế X, cũng như các phần tử của σ, phải có dạng 3NF. Rõ ràng là τ bảo toàn phụ thuộc vì σ bảo toàn phụ thuộc.

Để chứng minh rằng τ có nối không mất, chúng ta xây dựng các bảng áp dụng Thuật toán III.1. Chúng ta có thể chứng minh rằng toàn bộ hàng của X sẽ trở thành a như sau.

Xét thứ tự $A_1, A_2, \ldots A_k$, là thứ tự khi thêm các thuộc tính của R-X vào X^+ trong thuật toán tính bao đóng. Chắc chắn rằng cuối cùng thì tất cả thuộc tính sẽ được thêm vào bởi vì X là khoá. Chúng ta chứng minh bằng quy nạp trên i rằng cột tương ứng với A_i trong hàng của X được đặt bằng a_i trong phép kiểm của Thuật toán III.1.

Bước cơ sở i = 0 là hiển nhiên. Giả sử rằng kết quả đúng với i-1. Thế thì A_i được thêm vào X^+ do có một phụ thuộc hàm $Y \rightarrow A_i$, trong đó:

$$Y \subseteq X \cup \{A_1, \ldots A_{i-1}\}$$

Thế thì YA_i thuộc σ, và các hàng cho YA_i và X giống nhau ở Y (tất cả đều là a) sau khi các cột cho $A_1, \ldots A_{i-1}$ ở hàng X được đặt là các a. Vì thế những hàng này được biến đổi thành giống nhau ở thuộc tính Ai trong khi thực hiện Thuật toán III.1. Bởi vì hàng YA_i có ký hiệu a_i nên hàng X cũng thế.□

Hiển nhiên là trong một số trường hợp, τ không phải là tập lược đồ quan hệ nhỏ nhất có các đặc tính của Định lý III.7. Chúng ta có thể xoá bỏ các lược đồ quan hệ trong τ, mỗi lần một lược đồ, miễn là vẫn bảo toàn được các đặc tính mong muốn. Kết quả có thể là nhiều lược đồ CSDL, tuỳ thuộc vào thứ tự loại bỏ lược đồ, bởi vì loại bỏ một lược đồ có thể ngăn cản việc loại bỏ các lược đồ khác.

Thí dụ III.19:

Chúng ta có thể lấy hợp của lược đồ CSDL của CTHRSG được tạo ra trong Thí dụ III.18 với khoá SH, thu được một phân rã nối không mất và bảo toàn phụ thuộc.

Tình cờ, SH lại là tập con của HRS, là một trong những lược đồ quan hệ có trong phân rã. Vì thế chúng ta có thể loại bỏ SH và chấp nhận lược đồ CSDL của Thí dụ 7.18, nghĩa là

(CT, CHR, THR, CSG, HRS)

Mặt dù một số tập con của tập gồm năm lược đồ quan hệ này là những phân rã có nối không mất, chúng ta có thể kiểm chứng rằng các phụ thuộc được chiếu trên bốn trong năm lược đồ này không khẳng định được tập phụ thuộc F.□

II.7. Một số vấn đề cần lưu ý khi phân rã

Với những công cụ đã cho như Thuật toán III.3 và III.4, chúng ta thường bị "cám dỗ" phải phân rã các lược đồ quan hệ. Điều quan trọng cần nhớ là không phải mọi phân rã có nối không mất đều hữu ích, thậm chí một số còn có hại. Sai lầm hay gặp nhất là đi phân rã một lược đồ đã có dạng BCNF chỉ vì dường như có một phân rã có nối không mất và bảo toàn phụ thuộc.

Chẳng hạn chúng ta có thể có một quan hệ cung cấp thông tin về nhân viên, giả tỉ như mã số nhân viên I, tên nhân viên N, gian hàng D và lương S. Bởi vì I là khoá duy nhất trong trường hợp này, chúng ta có I→A với mọi thuộc tính A. Vì vậy hoàn toàn có thể phân rã lược đồ này thành IN, ID và IS. Dễ dàng thấy rằng phân rã này có nối không mất vì I, thuộc tính duy nhất trong phần giao của mỗi cặp, xác định tất cả các thuộc tính; cũng rõ ràng là nó bảo toàn phụ thuộc I→NDS.

Tuy nhiên chính lược đồ INDS lại có dạng BCNF, và có những ưu điểm khi phải trả lời những vấn tin có liên quan đến những thuộc tính khác I. Chẳng hạn nếu chúng ta muốn biết tên và lương của tất cả nhân viên của gian hàng đồ chơi trẻ em, chúng ta phải nối IN*ID*IS của lược đồ đã phân rã, nhưng có thể trả lời được câu hỏi mà không cần lấy bất cứ nối nào nếu chúng ta để nguyên lược đồ quan hệ cũ (và với một chỉ mục trên gian hàng D, chúng ta có thể trả lời câu vấn tin này hết sức hiệu quả).

Hơn nữa, lược đồ được phân rã đòi hỏi rằng mã số nhân viên được lập lại ở nhiều vị trí mặc dù nó không phải là dư thừa.

Tóm lại, khi áp dụng lý thuyết phân rã chúng ta cần phải nhớ rằng phân rã là "cứu cánh", được sử dụng để giải quyết các vấn đề dư thừa và bất thường, chứ không phải là mục đích. Khi áp dụng Thuật toán III.3, chúng ta nên tránh phân rã nếu lược đồ đã có BCNF. Khi sử dụng Thuật toán III.4, chúng ta nên xem xét đến việc tổ hợp các lược đồ quan hệ được tạo ra nếu không có vi phạm 3NF.

Chương IV: THIẾT KẾ CSDL Ở MỨC LOGIC

I. MỤC ĐÍCH

Quá trình thiết kế cơ sở dữ liệu bao gồm 3 giai đoạn tương ứng với 3 mức:

Mức quan niệm :	CÁI GÌ ?	Xác định nội dung của CSDL
Mức lôgic	↓	Chuẩn bị cho giai đoạn sau
Mức vật lý	↓ NHƯ THẾ NÀO ?	Chọn lựa cách cài đặt nội dung với một phần mềm cụ thể

Giai đoạn thiết kế logic là một bước trung gian, nhằm chuẩn bị cho việc lựa chọn ở giai đoạn vật lý được dễ dàng.

Đây hãy còn là một giai đoạn làm việc độc lập với các đặc trưng của phần mềm sẽ dùng sau này. Không những nó cần thiết trong trường hợp người thiết kế dùng một mô hình CSDL quan niệm (như mô hình dữ liệu quan hệ) khác với mô hình CSDL của phần mềm (như mô hình dữ liệu mạng hoặc phân cấp), mà còn cần thiết cả khi phần mềm cài đặt là một hệ quản trị CSDL quan hệ.

Thí dụ IV.1:

Cho cấu trúc quan niệm, kết quả của giai đoạn thiết kế quan niệm, như sau:

Nhanvien (MANV, HOTENNV, MAPH)

Tân từ: Mỗi nhân viên có một mã số (MANV) để phân biệt với các nhân viên khác, có họ và tên (HOTENNV) của nhân viên và trực thuộc một phòng (MAPH)

Phong (MAPH, TENPH)

Tân từ: Mỗi phòng ban có một mã số (MAPH) để phân biệt với các phòng khác, có tên phòng (TENPH)

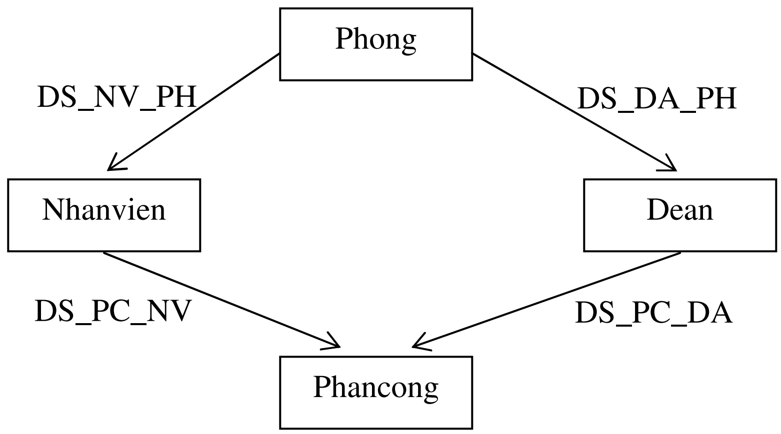
Dean (MADA, TENDA, MAPH)

Tân từ: Mỗi đề án có một mã số (MADA) để phân biệt với các đề án khác, có tên đề án (TENDA) và đề án nầy được phụ trách bởi một phòng (MAPH)

Phancong (MANV, MADA) với ràng buộc là một nhân viên được phân công vào tất cả đề án do phòng mà nhân viên đó trực thuộc phụ trách.

Tân từ: Mỗi nhân viên có thể được phân công thực hiện nhiều đề án và ngược lại mỗi đề án có thể do nhiều nhân viên thực hiện.

++ Nếu phần mềm sẽ được dùng là một hệ *quản trị cơ sở dữ liệu mạng*, thì cần phải biểu diễn cấu trúc trên theo mô hình dữ liệu mạng như sau:



- Hình IV.1 –Một cấu trúc lôgic theo mô hình DL mạng–

Việc xác định cấu trúc trên sẽ được thực hiện trong giai đoạn thiết kế lôgic.

Nếu một hệ quản trị CSDL quan hệ được dùng để cài đặt và nếu các quan hệ được khai báo y nguyên như trong cấu trúc quan niệm thì việc khai thác CSDL sau này có nhiều khả năng sẽ không được hiệu quả.

Thật vậy, thao tác quan trọng thường xảy ra nhất trong một CSDL quan hệ là phép kết. Để thao tác này được thực hiện một cách hiệu quả, hệ quản trị CSDL dựa trên các chỉ mục của các quan hệ liên quan. Do đó, vai trò của người thiết kế là làm thế nào xác định được đủ các chỉ mục cần thiết với số thuộc tính đúng và vừa đủ để khai thác: chỉ mục bao gồm nhiều thuộc tính không cần thiết hoặc tạo ra quá nhiều chỉ mục sẽ gây tốn chỗ và tốn kém trong việc bảo trì hệ thống chỉ mục, và tất nhiên dẫn đến hậu quả là CSDL sẽ hoạt động chậm chạp.

Cân nhắc để chọn lựa tạo chỉ mục nào sẽ được thực hiện trong giai đoạn thiết kế vật lý, tuy nhiên giai đoạn thiết kế lôgic sẽ chọn một biểu diễn ở dạng đồ thị để tạo việc cân nhắc đó được dễ dàng về sau. Dạng biểu diễn đồ thị này còn cho phép làm nổi bật các thuộc tính chung giữa 2 hay nhiều quan hệ (vì đây là cơ sở của phép kết), qua đó giúp cho người thiết kế sau này dễ dàng đánh giá và cân nhắc, cho những chỉ mục.

Trong quá trình xác định cấu trúc lôgic của CSDL, người thiết kế cần chú ý cấu trúc kết quả phải hoàn toàn trung thực với cấu trúc quan niệm, cụ thể là các điều kiện sau cần được bảo đảm:

- Bảo toàn nội dung CSDL:

Nội dung đã được xác định ở mức quan niệm phải được tôn trọng đầy đủ trong cấu trúc lôgic. Đây là một yêu cầu hiển nhiên, vì nội dung CSDL đã được phân tích, chọn lọc kỹ trong những giai đoạn phân tích nhu cầu và thiết kế quan niệm trước đó.

- Bảo toàn sự truy xuất "trực tiếp" đến các quan hệ cấu trúc quan niệm:

Khi một lược đồ quan hệ con R_i hiện diện trong lược đồ cơ sở dữ liệu ở cấu trúc quan niệm, một bộ trong một thể hiện của R_i có thể được truy xuất thẳng, không thông qua một quan hệ nào khác. Cấu trúc lôgic không được thay đổi khả năng truy xuất đó.

- Bảo toàn tính hiệu quả trong việc kiểm tra ràng buộc toàn vẹn:

Giai đoạn thiết kế quan niệm đã có chú ý đến yếu tố này thông qua tiêu chuẩn dạng chuẩn và bảo toàn phụ thuộc hàm. Cấu trúc lôgic cũng phải tôn trọng kết quả này.

Thí dụ IV.2:

Dựa trên cấu trúc quan niệm của Thí dụ IV.1, dưới đây là một số cấu trúc lôgic (được diễn đạt bằng mô hình quan hệ):

- Cấu trúc bảo toàn nội dung:

  Nhanvien (MANV, HOTENNV, MAPH)
  Phong (MAPH, TENPH)
  Da (MADA, TENDA)
  Phutrach (MADA, MAPH)

Với ràng buộc toàn vẹn đi kèm quan hệ Phancong, ta có thể tìm thấy đầy đủ những bộ của quan hệ này qua quan hệ suy diễn sau:

Nhanvien * Phutrach [MANV, MADA]

Đây là phép nối tự nhiên giữa hai quan hệ Nhanvien và Phutrach, hai quan hệ nầy có thuộc tính chung là MAPH.

Do đó, về mặt nội dung, nếu không cài đặt quan hệ Phancong, thông tin được lưu trong CSDL cũng không vì lý do đó mà bị thiếu.

- Cấu trúc không bảo toàn nội dung :
  Nhanvien (MANV, HOTENNV, MAPH)
  Phong (MAPH, TENPH)
  Da (MADA, TENDA)

Cấu trúc nầy không bảo toàn nội dung vì thiếu thông tin của Dean (MADA, MAPH)

- Cấu trúc không bảo toàn truy xuất :
  Nhanvien (MANV, HOTENNV, MAPH)
  Phong (MAPH, TENPH)
  Da (MADA, TENDA)
  Phutrach (MADA, MAPH)

Với cấu trúc nầy, ta không thể truy xuất trực tiếp đến quan hệ Phancong (MANV, MADA) như cấu trúc ở mức quan niệm, mà phải thông quan một phép kết giữa Nhanvien và Phutrach.

Tuy quan hệ Phancong thừa về mặt nội dung, nhưng nó đã hiện diện trong cấu trúc quan niệm, nghĩa là đã được cân nhắc kỹ. Sự hiện diện đó có nghĩa các bộ của Phancong cần được truy xuất thẳng, chứ không phải tính toán từ những quan hệ khác. Nếu Phancong không hiện diện trong cấu trúc lôgic thì khả năng truy xuất thẳng cũng bị mất đi.

- Cấu trúc bảo toàn tính hiệu quả:
  Nhanvien (MANV, HOTENNV, MAPH)
  Phong (MAPH, TENPH)
  Da (MADA, TENDA)
  Phutrach (MADA, MAPH)
  Phancong (MANV, MADA)

Tóm lại, giai đoạn thiết kế lôgic sẽ nhằm mục tiêu xác định một dạng biểu diễn đồ thị thỏa các điều kiện như đã trình bày ở trên. Nếu sau này, một hệ quản trị CSDL mạng được dùng để cài đặt thì việc chuyển từ dạng đồ thị này sang dạng biểu diễn mạng trở nên hiển nhiên.

Chương này nhằm chủ yếu cho việc cài đặt với một hệ quản trị CSDL quan hệ, là phần mềm được dùng phổ biến hiện nay.

II. BIỂU DIỄN CẤU TRÚC QUAN NIỆM DƯỚI DẠNG ĐỒ THỊ

II.1. Một số khái niệm trong lý thuyết đồ thị

• Đồ thị:

Một đồ thị ĐT (N, C) được định nghĩa trên một tập nút $N = (n_1, n_2, \dots, n_n)$ và 1 tập cung $C = (c_1, c_2, \dots, c_m)$.

- Nếu hiện diện cung có hướng, đó là đồ thị có hướng, và các nút được gọi là nút đi hoặc nút đến.
- Nếu tất cả cung là vô hướng, đó là đồ thị vô hướng, và các nút được gọi là nút xuất phát.

• Cung kề cận:

Hai cung (c_1, c_2) được gọi là kề cận nhau khi:

- đối với đồ thị vô hướng: chúng có chung một nút xuất phát;
- đối với đồ thị có hướng: nút đến của c_1 là nút đi của c_2.

• Khuyên:

Cung c là một khuyên nếu hai nút đi/đến (hoặc xuất phát) của c là một.

• Đường đi (đối với đồ thị vô hướng):

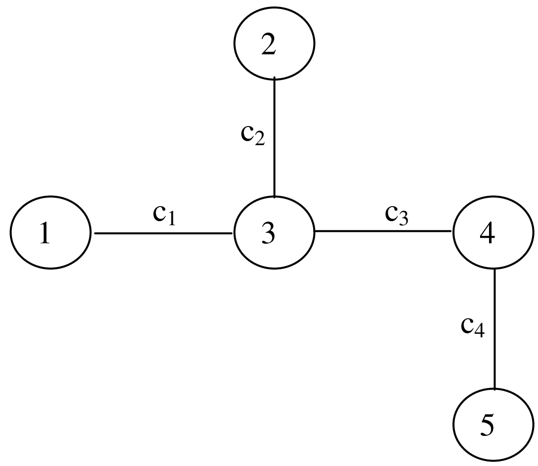
Đó là một chuỗi cung $(c_1, c_2, \dots c_p)$ sao cho:

- c_i và c_{i+1} có chung một nút xuất phát.
- nếu c_i không phải là khuyên hoặc cung đầu hoặc cung cuối thì c_i có chung 1 nút xuất phát với c_{i-1} và nút xuất phát còn lại của c_i cũng là nút xuất phát của c_{i+1} .

++ Một nút xuất phát của c_1 (cung đầu trong chuỗi) không chung một nút xuất phát của c_2 được gọi là nút đầu của đường đi.

++ Một nút xuất phát của c_p (cung cuối trong chuỗi) không chung một nút xuất phát của c_{p-1} được gọi là nút cuối của đường đi.

Thí dụ IV.3:



- Hình IV.2 –Một đồ thị vô hướng–

- (c_1, c_2, c_3, c_4,) không phải là một đường đi
- (c_1, c_3, c_4,) là một đường đi với 1 là nút đầu và 5 là nút cuối.

• <u>Mạch đi</u>: (đối với đồ thị có hướng):

Đó là một chuỗi cung (c_1, c_2, …, c_p) sao cho nút đến của cung c_{i-1} là nút đi của cung c_i, với $i \le p$.

• <u>Chu trình</u>: (đối với đồ thị có hướng):

Là mạch đi trong đó:

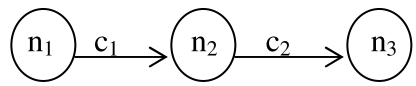
- nút đầu và nút cuối trùng nhau
- 1 cung không xuất hiện 2 lần trong chuỗi.

- Hai đường đi / mạch đi khác nhau khi chúng không có một cung chung nào.
- Một dòng có gốc n_1 là một tập cung $D=(c_1, c_2, \ldots, c_p)$ sao cho :

  - một cung trong tập đó có nút xuất phát (hoặc nút đi) là n_1

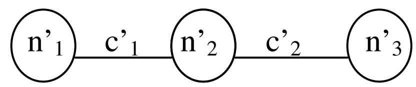
  - $\forall c_i$, $\forall n_i$, nút xuất phát (hoặc nút đi/đến) của c_i thuộc một đường đi (hoặc mạch đi)có nút đầu là n_1, nút cuối là n_i và gồm các cung của tập D.

Thí dụ IV.4:



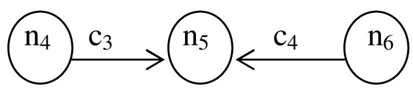
(c_1, c_2) là một dòng có gốc n_1

(c_1, c_2) không phải là một dòng có gốc n_2



(c'_1, c'_2) là một dòng có gốc n'_1

(c'_1, c'_2) cũng là một dòng có gốc n'_2 hoặc n'_3



(c_3, c_4) không phải là dòng của gốc nào cả.

II.2. Đồ thị con đường truy xuất

II.2.1. Định nghĩa

Đồ thị con đường truy xuất (N, C, Q, Cđ, f, g, h, i, j) là một đồ thị có hướng với:

N	: tập nút
$C \subseteq (N \times N)$	: tập cung có hướng
Q	: tập quan hệ Q_i

Cđ	: tập các con đường truy xuất
N	: tập các số tư nhiên
$f: N \rightarrow Q$	: đơn ánh
$g: C \rightarrow Q$	: ánh xạ
$h: C \rightarrow Cđ$	: song ánh
i: Cđ→*N* x *N* x *N*	: ánh xạ; *N* là tập các số tự nhiên
$j: N \rightarrow \{0,1\}$	: ánh xạ; khi $j(n_i)=1$ thì n_i là một nút vào.

- Điều kiện : $f(N) \cup g(C) = Q$
- Q_N được gọi là quan hệ nút nếu $\exists n_i$ thuộc đồ thị con đường truy xuất sao cho:

  $$f(n_i) = Q_N$$

  Q_C được gọi là quan hệ cung nếu $\exists c_j$ thuộc đồ thị con đường truy xuất sao cho:

  $$g(c_j) = Q_C$$

- Tồn tại tối đa 2 cung thuộc $g^{-1}(Q_C)$, 2 cung này có 2 chiều ngược nhau, nút đi của cung thứ nhất là nút đến của cung thứ hai và ngược lại.
- Ký hiệu : - cung : → hoặc →>
  - nút: ①
  - nút vào: ①⇦

II.2.2. Diễn giải

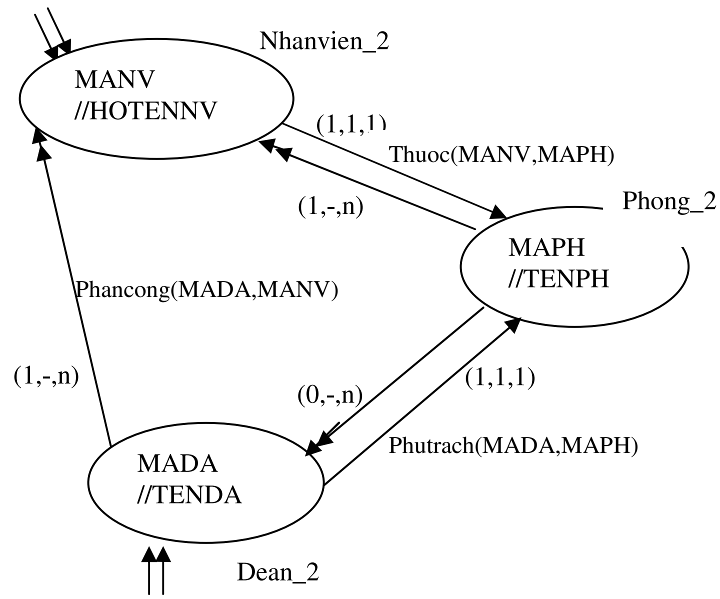
* $N_i \xrightarrow{c_{ij}} N_j$ từ một quan hệ nút $f(N_i)$ có thể truy xuất đến một bộ của quan hệ nút $f(N_j)$ thông qua con đường truy xuất $h(c_{ij})$

Trên mỗi con đường truy xuất có gắn 1 tổ hợp (x_1, x_2, x_3) qua ánh xạ $i(cd_{ij})$: đó là

x_1 : số bộ tối thiểu
x_2 : số bộ trung bình
x_3 : số bộ tối đa

của quan hệ nút f (N_j) có thể truy xuất được từ 1 bộ của quan hệ nút f (N_i)

Thí dụ IV.5:



- Hình IV.3 –Một đồ thị các con đường truy xuất–

Có 2 ngõ vào CSDL: đó là Nhanvien_2 và Dean_2, nghĩa là ta có thể cung cấp giá trị của mã nhân viên để truy xuất ngay một bộ tương ứng trong quan hệ Nhanvien_2. Phong_2 không phải là một ngõ vào, nghĩa là mỗi khi muốn truy xuất một phòng cụ thể, ta phải hoặc duyệt tuần tự các bộ của Phong_2 để tìm, hoặc thông qua một con đường truy xuất đến từ Nhanvien_2 hoặc Dean_2.

Từ một bộ của Nhanvien_2, ta có thể truy xuất trực tiếp một bộ của Phong_2 mà nhân viên trực thuộc, thông qua con đường truy xuất Nhanvien_2 → Phong_2, ngược lại ta cũng có thể có ngay danh sách nhân viên của một phòng thông qua con đường truy xuất Phong_2 → Nhanvien_2 .Nhưng ta không thể truy xuất trực tiếp danh sách đề án mà một nhân viên được phân công vào làm việc, vì không có con đường truy xuất Nhanvien_2 → Dean_2; tất nhiên ta vẫn có thể có được danh sách trên một cách gián tiếp, chẳng hạn thông qua các con đường truy xuất Nhanvien_2 → Phong_2 → Dean_2 nếu có ràng buộc toàn vẹn trên quan hệ Phân công như đã ghi trong Thí dụ IV.1.

Trong đồ thị con đường truy xuất, có một đồ thị đặc biệt trong đó, giữa hai nút của đồ thị, nếu có một cung thì bao giờ cũng có một cung theo chiều ngược lại, và tất cả các nút là những nút vào: đó là đồ thị con đường truy xuất thô.

Với đồ thị của Hình IV.3, nếu từ nút *Đề án 2* đến nút *Nhân viên*_2 có thêm một con đường truy xuất, và nếu nút *Phòng*_2 cũng là nút vào thì đồ thị trở thành đồ thị con đường truy xuất thô.

II.3. Đồ thị quan hệ

Trong quá trình chuyển sang dạng biểu diễn đồ thị, một cấu trúc quan niệm có thể được biểu diễn thành nhiều đồ thị khác nhau, nhưng phải tôn trọng tiêu chuẩn của một biểu diễn trung thực (bảo toàn nội dung, bảo toàn tính truy xuất “ trực tiếp”, bảo toàn tính hiệu quả trong kiểm tra RBTV); tuy vậy không phải tất cả đều có những hiệu quả khai thác hoàn toàn như nhau.

Một đồ thị con đường truy xuất được đơn giản hoá sẽ giúp người thiết kế đánh giá dễ dàng chất lượng của dạng biểu diễn đồ thị này: đó là đồ thị quan hệ.

Khi xác định dạng biểu diễn đồ thị của một cấu trúc quan niệm thành 2 bước:

1. đồ thị quan hệ.
2. đồ thị con đường truy xuất.

Người thiết kế có thể lần lượt phân tích các vấn đề đặt ra trong giai đoạn này dưới 2 khía cạnh:

1. khía cạnh quan niệm trước tiên, với công cụ phân tích là đồ thị quan hệ.
2. rồi đến khía cạnh phương tiện truy xuất dữ liệu, với công cụ phân tích là đồ thị con đường truy xuất.

II.3.1. Định nghĩa

Một đồ thị quan hệ là một đồ thị có hướng, được định nghĩa trên (N_Q, C_Q, Q_Q, f_Q, g_Q, k_Q) với:

N_Q	:	tập nút
$C_Q \subseteq N_Q \times N_Q$	:	tập cung có hướng hoặc không
Q_Q	:	tập quan hệ Q_i
$f_Q : N_Q \rightarrow Q_Q$	:	đơn ánh
$g_Q : C_Q \rightarrow Q_Q$	:	đơn ánh
$k_Q : C_Q \rightarrow \{0,1\}$	:	ánh xạ : $k_Q(c) = 1$ nếu là một cung

có hướng, $k_Q(c) = 0$ nếu là một cung vô hướng.

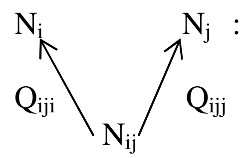
II.3.2. Diễn giải

$$\begin{tikzcd} N_i \arrow[rr, "c_{ij}"] \arrow[d, dash, "f_Q"] & {} \arrow[d, dash, "g_Q"] & N_j \arrow[d, dash, "f_Q"] \\ Q_i & Q_{ij} & Q_j \end{tikzcd}$$

có 1 phụ thuộc hàm $K_{Qi} \rightarrow K_{Qj}$ với K_{Qi} , K_{Qj} lần lượt là một khoá của Q_i ,Q_j ; và quan hệ cung Q_{ij} được hình thành từ tất cả thuộc tính khóa của Q_i ,Q_j : $Q_{ij}^+ = KQ_i^+ \cup KQ_j^+$.

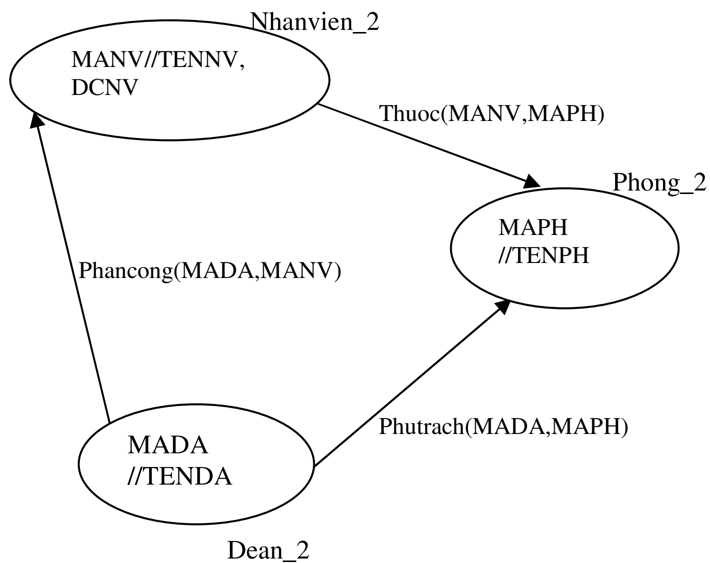
- N_i ——— N_j : không có phụ thuộc hàm giữa $K_{Qi} \rightarrow K_{Qj}$ và quan hệ cung Q_{ij} bao gồm $KQ_i^+ \cup KQ_j^+$.

Đồ thị này có thể đổi thành:



cả hai quan hệ cung Q_{iji} , Q_{ijj} và quan hệ nút Q_{ij} đều được hình thành từ tập $KQ_i^+ \cup KQ_j^+$.

Hình dưới đây minh hoạ một đồ thị quan hệ.



- Hình IV.4 –Một đồ thị quan hệ–

Giữa hai quan hệ *Nhanvien*_2 và *Phong*_2 có một phụ thuộc hàm MANV $\rightarrow$ MAPH ; phụ thuộc hàm này được biểu diễn bởi cung nối từ nút của quan hệ *Nhanvien*_2 sang nút của quan hệ *Phong*_2, quan hệ của cung này được đặt tên là *Thuoc*; nó được hình thành từ phép chiếu của quan hệ

Nhanvien gốc (nằm trong cấu trúc quan niệm) trên các thuộc tính khoá của 2 quan hệ nút liên quan.

II.3.3. Biến đổi một đồ thị quan hệ sang một đồ thị con đường truy xuất thô và ngược lại

(a) Một đồ thị con đường truy xuất thô được biến đổi thành một đồ thị quan hệ như sau:

ĐTCĐTX thô (N, C, Q, Cđ, f, g, h, i, j)

$\downarrow$

ĐTQH (N_Q, C_Q, Q_Q, f_Q, g_Q, k_Q)

với:

- $N_Q = N$
- $Q_Q = Q$
- $f_Q = f$
- $\forall(c, c') \in C$ có chiều ngược nhau, và $g(c) = g(c')$, $\exists c_Q \in C_Q$ sao cho:
  $g_Q(c_Q) = g(c)$ ($= g(c')$)
- $k_Q(c_Q) = 0$ nếu max của c và cuả c' > 1 (cung c_Q là vô hướng)
  $k_Q(c_Q) = 1$ nếu ngược lại (cung c_Q là có hướng)
- $c_Q = c$ nếu c_Q vô hướng hoặc max (c) ≤ 1
  $c_Q = c'$ nếu ngược lại

(b) Từ một đồ thị quan hệ có thể biến đổi thành một đồ thị con đường truy xuất thô như sau:

- $N = N_Q$
- $Q = Q_Q$
- $f = f_Q$
- $\forall c_Q \in C_Q$, $c_Q = (n_1, n_2)$, $\exists$ (c, c') với $c = (n_1, n_2)$
  và $c' = (n_2, n_1)$

và $g(c) = g_Q(c_Q)$
và $g(c') = g_Q(c_Q)$

- Nếu $k_Q(c_Q) = 0$ thì max (c) và max (c') > 1
  Nếu $k_Q(c_Q) = 1$ thì một trong hai max(c),max(c') ≤ 1

- $\forall n \in N_Q$; j(n) =1 (nút vào)

II.4. Chuỗi kết được cài đặt trên đồ thị

Như đã đề cập ở đầu chương, mối quan tâm hàng đầu của người thiết kế ở giai đoạn này là chọn một cấu trúc lô gic cho phép thực hiện hiệu quả phép kết. Cơ sở để đánh giá tiêu chuẩn này là khái niệm chuỗi kết được cài đặt trên đồ thị con đường truy xuất.

II.4.1. Định nghĩa đối với đồ thị con đường truy xuất

Một chuỗi kết $\rho = Q_1 >< Q_2 \ldots >< Q_m$ được cài đặt trên đồ thị con đường truy xuất
(N, C, Q, Cđ, f, g, h, i, j) nếu và chỉ nếu:
- $\forall Q_i$, $i \in (1\ldots m)$, $Q_i \in Q_Q$ và
- $\exists$ 1 dòng D = $(c_1, c_2, \ldots c_p)$ trên đồ thị con đường truy xuất sao cho mỗi cung c_i thoả 2 điều kiện sau:
  + $g(c_i) = Q_j$, $j \in (1\ldots m)$
  + $\forall Q_i$, $i \in (1\ldots m)$:
    - hoặc $\exists$ 1 cung c của D sao cho $g(c) = Q_i$
    - hoặc $\exists$ 1 nút n của D sao cho $f(n) = Q_i$

II.4.2. Định nghĩa đối với đồ thị quan hệ

Một chuỗi kết $\rho = Q_1 >< Q_2 \ldots >< Q_m$ được cài đặt trên đồ thị quan hệ
$(N_Q, C_Q, Q_Q, f_Q, g_Q, k_Q)$ nếu và chỉ nếu:
- $\forall Q_i$, $i \in (1\ldots m)$, $Q_i \in Q_Q$ và

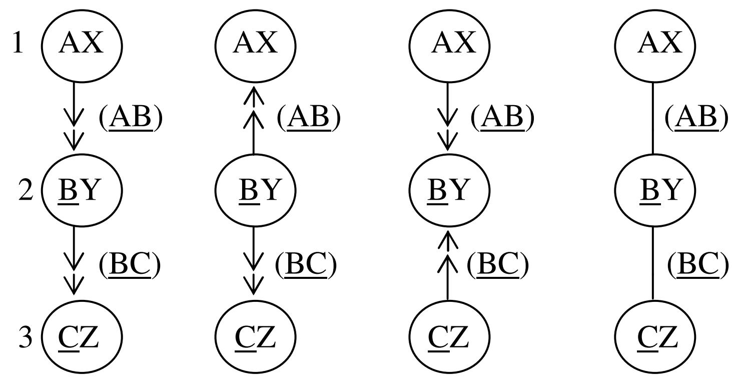
• ∃ 1 dòng D = ($c_1, c_2, \ldots c_p$) trên đồ thị quan hệ sao cho mỗi cung c_i thoả 2 điều kiện sau:

+ $g_Q(c_i) = Q_j$, $j \in (1\ldots m)$

+ $\forall Q_i$, $i \in (1\ldots m)$:

- hoặc $\exists\, c \in D$ sao cho $g_Q(c) = Q_i$
- hoặc $\exists\, n \in D$ sao cho $f_Q(n) = Q_i$

Thí dụ IV.6:



(4.8a) (4.8b) (4.8c) (4.8d)

ρ = (AX) >< (AB) >< (BY) >< (BC) >< (CZ) được cài đặt trên (4.8a), (4.8b), (4.8d) nhưng không được cài đặt trên (4.8c).

II.4.3. Diễn giải

- Nếu một chuỗi kết được cài đặt trên đồ thị (con đường truy xuất hoặc quan hệ) thì sẽ tồn tại một dòng D có gốc n_g. Từ quan hệ Q_g của nút gốc n_g thuộc dòng D, ta có thể truy xuất nhanh những bộ của $Q_i \in \rho$ thông qua các mạch đi (đối với đồ thị con đường truy xuất) hoặc thông qua các đường đi (đối với đồ thị quan hệ) xuất phát từ n_g .
- Trong Vd 4.8 chuỗi kết

ρ = (AX) >< (AB) >< (BY) >< (BC) >< (CZ) được thể hiện như sau:

- Trong (4.8a): $n_g = 1$

  mạch đi :{(1,2), (2,3)} : (AX) >< (AB) >< (BY) >< (BC) >< (CZ)
- Trong (4.8b): $n_g = 2$

  mạch đi :{(2,1), (2,3)} : (BY) >< (AB) >< (AX) >< (BC) >< (CZ)
- Trong (4.8c): chỉ có thể xuất phát từ 1 hoặc 3 nhưng:

  + từ 1 chỉ đến được 2, không đến được hoặc không đi qua 3;
  + từ 3 chỉ đến được 2, không đến được hoặc không đi qua 1.

$\Rightarrow$ + hoặc với gốc $n_g = 1$, dùng mạch đi (1,2), sau đó đọc tuần tự tất cả các bộ của (CZ) và đối sánh với kết quả của mạch đi.

+ hoặc với gốc $n_g = 3$, dùng mạch đi (3,2), sau đó đọc tuần tự tất cả các bộ của (AX) và đối sánh với kết quả của mạch đi.

- Trong (4.8d):có hai dòng gốc là 1 và 3 để thực hiện ρ

III. THUẬT TOÁN BIỂU DIỄN MỘT CẤU TRÚC CƠ SỞ DỮ LIỆU QUAN HỆ SANG ĐỒ THỊ QUAN HỆ

III.1. Mục tiêu

Ngoài 3 mục tiêu đã được đề ra cho quá trình biến đổi từ cấu trúc quan niệm (ở dạng quan hệ) sang cấu trúc ở dạng lô gic, thuật toán sẽ được trình bày trong phần này còn nhằm thêm 2 mục tiêu khác:

- Đối với quan hệ có nhiều khoá, tất cả các khoá đều được gán cho một vai trò ngang nhau; thật vậy việc đánh giá ưu tiên cho khoá nào trong số các khoá của một quan hệ thuộc về lãnh vực cài đặt, và sẽ được cân nhắc ở giai đoạn thiết kế vật lý. Điều này giải thích bước thứ nhất của thuật toán.

-Làm nổi bật những tập thuộc tính chung của mỗi cặp quan hệ, vì đó là cơ sở của phép kết.

III.2. Thuật toán

Vào :

Một phân rã ρ của Q dựa trên F : $\{Q_i\}^n_{i=1}$, mỗi Q_i có tập khoá (K_i)

Ra :

Đồ thị quan hệ tương ứng với ρ .

Các bước :

B1: Biến đổi ρ thành một phân rã đồng nhất ρ_d:

1.1. Với mọi cặp quan hệ con Q_i, Q_j, nếu $K_i \leftrightarrow K_j$, K_i, K_j lần lượt là một khoá của Q_i, Q_j, thì gộp Q_i, Q_j lại thành một quan hệ con.

1.2. Với mỗi Q_i , nếu Q_i^+ có chứa một khoá Kj của Q_j, thì Q_i^+ phải chứa tất cả các khoá của Q_j.

B2: Tạo nút và quan hệ nút:

Mỗi quan hệ Q_i là một nút N_i với $Q_{N_i} = Q_i$.

B3: Tạo nút bản lề và quan hệ (nút) bản lề:

Mục đích làm nổi bật các thuộc tính chung của mỗi cặp quan hệ nút.

3.1. $\forall Q_i$, Q_j, $Q_{ij}^+ = Q_i^+ \wedge Q_j^+$.

3.2. Chừng nào $Q_{ij}^+ <> \varnothing$ thì:

- xác định tất cả khoá của Q [Q_{ij}^+] ; KQ_{ij}^+ ký hiệu tập thuộc tính khoá của Q [Q_{ij}^+] .
- Nếu $\nexists$ $Q_h \in \rho_d$ sao cho 1 khoá của Q_h là một khoá của Q [Q_{ij}^+]

thì tạo 1 nút bản lề N_{b1} với quan hệ Q_{b1} = $Q\ [KQ_{ij}^+\ ]$

Cuối nếu

- $Q_{ij}^+ := Q_{ij}^+ - KQ_{ij}^+$

Cuối chừng nào

B4: Tạo cung:

Chú ý : chỉ tạo số cung tối thiểu xuất phát từ một nút.

4.1.

$\forall N_i$ với Q_i , xác định:

PTH (N_i) = {N_j với $Q_i^+ \supset KQ_j^+$}

PTH_THỪA (N_i) = {$N_j \in$ PTH (N_i) sao cho :

$\exists\ N_h \in$ PTH (N_i) với Q_h sao cho

$KQ_h^+ \supset KQ_j^+$}

LỒNG_KHOÁ(N_i)={N_j với Q_j sao cho$KQ_i^+ \supset KQ_j^+$}

LỒNG_KHOÁ_THỪA(N_i)={$N_j \in$ LỒNG_KHOÁ(N_i)

sao cho:

$\exists\ N_h \in$ LỒNG_ KHOÁ (N_i) với Q_h

sao cho: $KQ_h^+ \supset KQ_j^+$}

Cung (N_i) = (PTH (N_i) - PTH_THỪA (N_i))

$\cup$ (LỒNG_ KHOÁ (N_i) - LỒNG_ KHOÁ_THỪA (N_i))

Cuối $\forall$

4.2.

$\forall N_i \in$ Cung (N_i) thì

tạo 1 cung có hướng từ $N_i \rightarrow N_j$: cij

Cuối $\forall$

4.3. $Q_{ij} = Q_i\ [KQ_i^+ \cup KQ_j^+\ ]$

B.5: Huỷ những nút bản lề thừa

$\forall N_i$ sao cho :

- có một khoá duy nhất là K_k
- Không có thuộc tính nào khác ngoài khoá
- Chỉ có một cung vào uất phát từ nút N_i

Thì : /* vai trò bản lề của N_k không còn cần thiết nữa*/

- Nhập N_k vào N_i
- huỷ cung c_{ik}

Cuối $\forall$

B.6: Min hoá các quan hệ nút:

$\forall N_i$ với Q_i thì:

$\forall N_j \in$ Cung (N_i) với Q_j thì:

Huỷ khỏi Q_i^+ những thuộc tính khoá của Q_j mà không phải thuộc tính khoá của Q_i.

Cuối $\forall$

Cuối $\forall$

B.7: Tạo cung vô hướng :

$\forall N_k$ sao cho :

- $Q_k^+ = KQ_k^+$ (nghĩa là Q_k không có thuộc tính không khoá)
- Chỉ có 2 cung ra khỏi N_k (không có cung vào, đến N_i , N_j với Q_i, Q_j sao cho $KQ_k^+ = KQ_i^+ \cup KQ_j^+$

thì

- tạo 1 cung vô hướng nối N_i ,N_j với $Q_{ij} = Q_k$
- huỷ nút N_k
- huỷ 2 cung xuất phát từ N_k

Cuối $\forall$

VD 4.9:

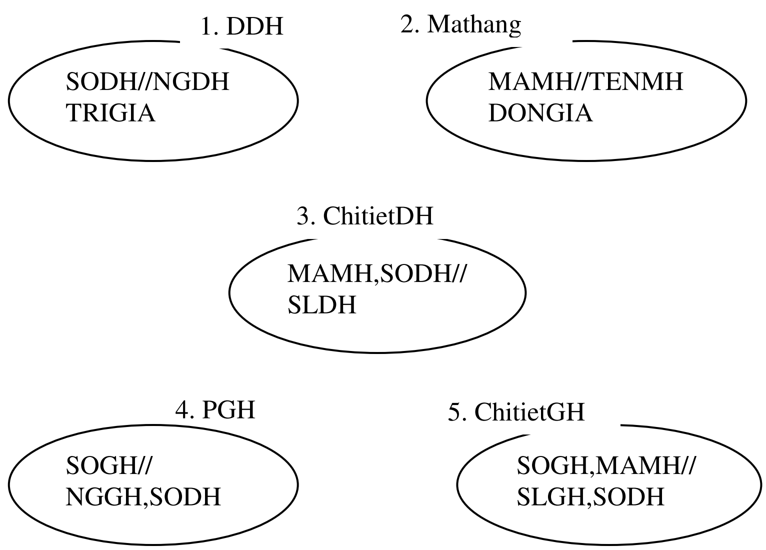
Cho cấu trúc quan niệm như sau:

1. DDH (SODH, NGDH, TRIGIA)
2. Mathang (MAMH, TENMH, DONGIA)
3. ChitietDH (SODH, MAMH, SLDH)
4. PGH (SOGH, NGGH, SODH)
5. ChitietGH (SOGH, MAMH, SLGH, SODH)

B1:

Phân rã ban đầu không thay đổi, vì không có tình trạng khoá tương đương giữa hai quan hệ, và trong các quan hệ cũng không có nhiều khoá.

B2: Tạo nút .



- Hình IV.5 –Kết quả tạo nút ở bước 2–

B.3 Dưới đây là những tập thuộc tính chung không rỗng của từng cặp quan hệ.

1 và 3 : SODH, khoá của 1
1 và 4: -nt-
1 và 5: -nt-

2 và 3: MAMH, khoá cuả 2
2 và 5: -nt-

3 và 5: MAMH, SODH, khoá cuả 3

4 và 5: SOGH, SODH ==>
khoá của tập này là SOGH, lại là khoá của 4, còn lại SODH lại là khoá 1.

==> kết luận: không tạo nút bản lề nào cả.

B.4 : Tạo cung

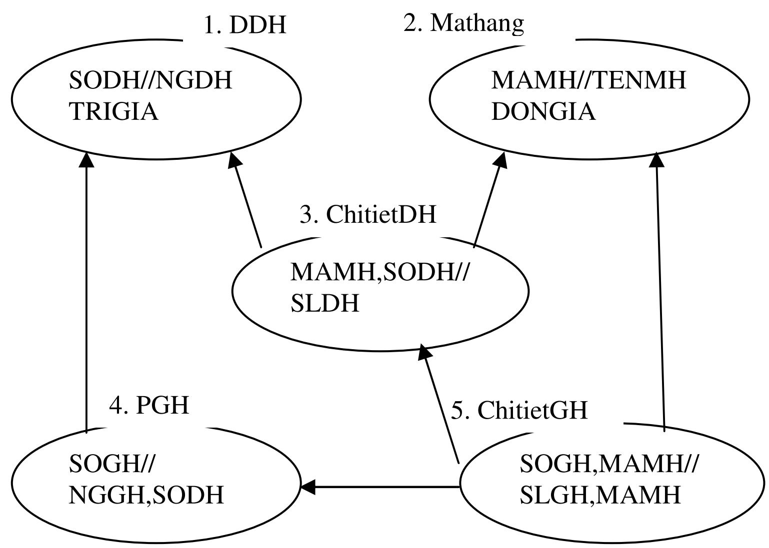
	PTH (Q_i)	PTH_THỪA (Q_i)	Lồng Khóa (Q_i)	LK_THỪA (Q_i)	Cung (Q_i)
1.ĐĐH	∅	-	-	-	∅
2.Mathang	∅	-	-	-	∅
3.ChitietDH	1,2	∅	1,2	∅	1,2
4.PGH	1	∅	∅	-	1
5.ChitietGH	1,2, 3,4	1,2	2,4	∅	2,3,4

<u>Ghi chú:</u> "-" có nghĩa là không cần thiết phải tính tập này, như trong trường hợp tập PTH(Q_i) là rỗng thì không cần tính tập là LK(Q_i) , vì theo định nghĩa, tập sau nằm trong tập trước.

Các quan hệ cung là:

- cung 31 : CTDH_DDH (MAMH, SODH)
- cung 32 : CTDH_MH (MAMH, SODH)
- cung 41 : PGH_DDH (SOGH, SODH)
- cung 52 : CTGH_MH (SOGH, MAMH)
- cung 53 : CTGH_CTDH(SOGH, MAMH,SODH)
- cung 54 : CTGH_GH(SOGH, MAMH)

Kết quả bước này được trình bày trong Hình IV.6 dưới đây:



- Hình IV.5 –Kết quả của các bước từ 1 đến 4–

B5 : Không thực hiện, vì đã không tạo nút bản lề nào cả.

B6 :

- Trong quan hệ nút *PGH*, loại bỏ thuộc tính SODH.
- Trong quan hệ nút *ChitietGH*,loại bỏ thuộc tính SODH.

B7 : Không tạo được cung vô hướng nào cả.

Nhận xét :

1) 5→4→1 & 5→3→1: đây là 2 mạch đi khác nhau cùng xuất phát từ nút 5 và cùng đến nút 1; 2 mạch đi này kiểm tra rằng nếu một chi tiết giao hàng liên quan đến một chi tiết của một đơn đặt hàng (mạch đi 5→3→1) và liên quan đến một đợt giao hàng thì đợt giao hàng này phải của cùng đơn đặt hàng đã biết (mạch đi 5→4→1). Một khi cơ chế kiểm tra này hoạt động thường trực, thì dễ truy xuất thông tin sau:
   "các chi tiết giao hàng của một dơn đặt hàng cùng với ngày đặt hàng", nghĩa là thực hiện phép chiếu sau:
   Q{SOGH, MAMH, SLGH, SODH, NGDH}. Quan hệ này có thể truy xuất thông qua một trong hai chuỗi kết sau có gốc là 5 (cả hai chuỗi kết đều được cài đặt trên đồ thị quan hệ và đều cung cấp kết quả như nhau):
   (5 >< 53 >< 3 >< 31 ><3 […..] (mạch đi 5→3→1), hoặc
   (5 >< 54 >< 4 >< 41 ><1 [ ] (mạch đi 5→4→1).
2) 5→3→2 & 5→2 : hiện tượng này do có sự lồng khoá giữa các quan hệ 3&2 và 5&2.

III.3. Biến đổi ngược : từ một đồ thị quan hệ sang một cấu trúc cơ sở dữ liệu quan hệ

Thực hiện quá trình biến đổi ngược này là để kiểm chứng lại xem cấu trúc quan hệ được biểu diễn qua đồ thị quan hệ có hoàn toàn tương đương với cấu trúc quan hệ ban đầu hay không (tương đương ở đây được hiểu theo nghĩa bảo toàn thông tin và bảo toàn phụ thuộc hàm).

1. Gọi ρ^{-1}, tập quan hệ con có được sau khi áp dụng quá trình biến đổi ngược.

$\rho^{-1} = \{ Q_i \} \cup \{ Q_{ij} \}$, Q_i là quan hệ nút, Q_{ij} là quan hệ cung.

2. Gộp các quan hệ có cùng khoá lại thành một.

Quan sát ρ^{-1} và ρ_d , ta nhận thấy số lượng quan hệ, thành phần của từng quan hệ sẽ như nhau. Điều này được bảo đảm do chính nội dung thuật toán.

IV. CÁC TRƯỜNG HỢP CẦN LƯU Ý

IV.1. TRƯỜNG HỢP 1

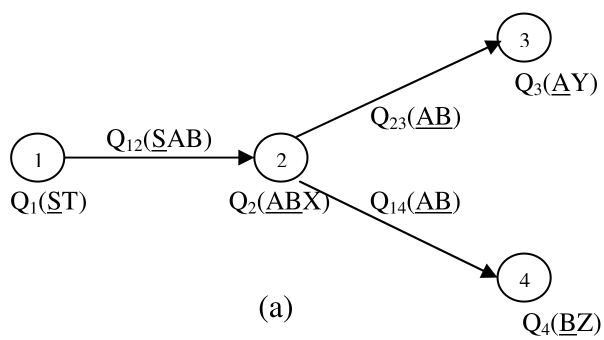
Cho cấu trúc quan hệ sau:

$C_0 = < Q_0$ (SABXYZT), $F_0 = \{S \rightarrow ABT;\ AB \rightarrow X;\ A \rightarrow Y;\ B \rightarrow Z\} >$

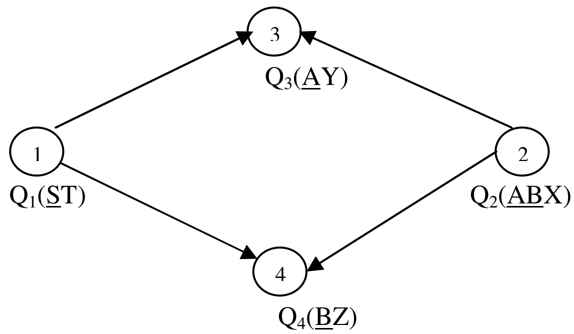
Và cấu trúc CSDL quan niệm:

$C_1 = \{< Q_1$ (SABT), $F_1 = \{S \rightarrow ABT\} >$;
$< Q_2$ (ABX) , $F_2 = \{AB \rightarrow X\} >$;
$< Q_3$ (AY) , $F_3 = \{A \rightarrow Y\} >$;
$< Q_4$ (BZ) , $F_4 = \{B \rightarrow Z\} > \}$

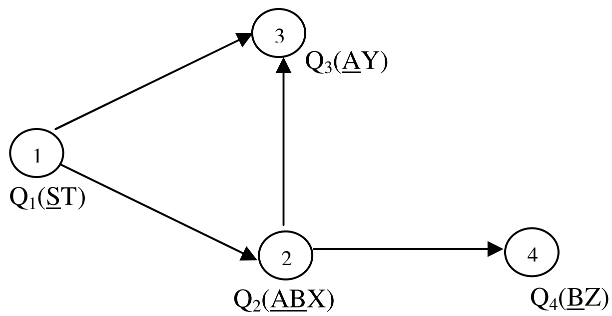
Có 4 cách biểu diễn khác nhau của C_1 ở dạng đồ thị quan hệ :



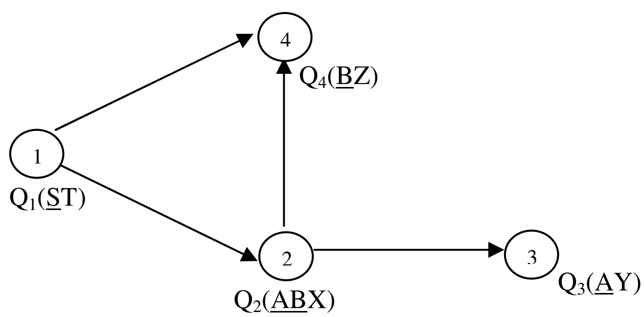
(a)



(b)



(c)



(d)

- Hình IV.7 –Bốn đồ thị quan hệ của C_1–

Dưới góc độ thao tác quan hệ, cả 4 đồ thị quan hệ đều như nhau. Nếu ta xem xét mỗi chuỗi kết dựa trên thuộc tính chung có thể đặt ra cho cấu trúc quan hệ này, thì những chuỗi kết ấy đều được cài đặt trên cả 4 đồ thị quan hệ.

Chẳng hạn như muốn truy xuất Q_0[STAY], nghĩa là:
(Q_1 >< Q_3) [STAY];

- với đồ thị (a): có chuỗi kết được cài đặt với mạch đi (12,23)
- với đồ thị (b): dùng mạch đi (13)
- với đồ thị (c): có 2 mạch đi khác nhau: (13) và (12,23)
- với đồ thị (d): dùng mạch đi (12,23)

Nhưng nếu Q_0[AB] cần được truy xuất, thì cách truy xuất sẽ có khác nhau trên 4 đồ thị :

- với (a): truy xuất chỉ ở nút 2, nghĩa là Q_2[AB], là có đầy đủ những cặp giá trị (AB) được lưu trong CSDL
- với (b): mạch đi (13, 14) và nút 2, nghĩa là (Q_{13} >< Q_{14}) [AB] $\cup$ Q_2[AB]; hình 4.8 minh hoạ một tình trạng CSDL, qua đó mạch đi (13,14) sẽ cung cấp các bộ {(a_1, b_3), (a_3, b_2) }và nút 2 sẽ cung cấp các bộ {(a_1, b_1), (a_1, b_3) }
- với (c) & (d): truy xuất chỉ ở 2 nút là đủ; tuy nhiên 2 đồ thị quan hệ này có ý nghĩa khai thác hơi khác với đồ thị quan hệ (a) ở chỗ cho phép người sử dụng khả năng tạo một cách độc lập với những bộ Q_0[SAB], những bộ Q_0[SA] thông qua việc tạo các thể hiện của cung (13) đối với đồ thị (c), hoặc những bộ Q_0[SB] thông qua việc tạo các thể hiện của cung (14) đối với đồ thị (d). Khả năng này không có được trong đồ thị (a).

TQ$_1$

S	A	B	T
s_1	a_1	b_3	t_1
s_2	a_3	b_2	t_2

TQ$_2$

A	B	X
a_1	b_1	x_1
a_1	b_3	x_2

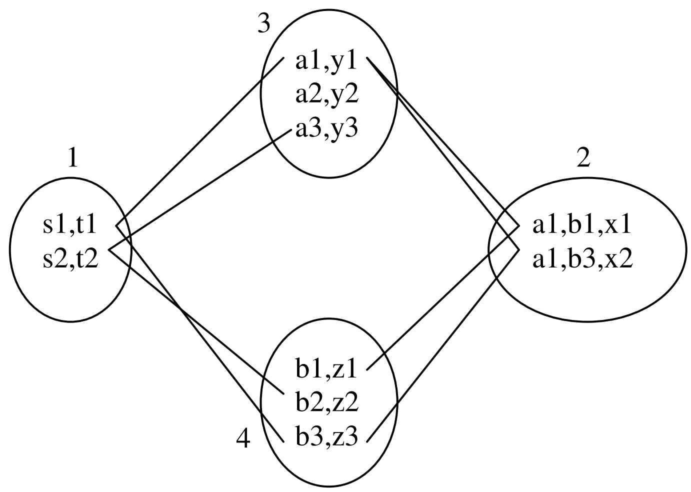
TQ$_3$

A	Y
a_1	y_1
a_2	y_2
a_3	y_3

TQ$_4$

B	Z
b_1	z_1
b_2	z_2
b_3	z_3

(4.8.a)



- Hình IV.8 –Một tình trạng CSDL được trình bày dưới dạng bảng (a) và dưới dạng đồ thị quan hệ (b)–

Nếu một hệ QTCSDLQH được chọn để cài đặt, trên quan hệ Q_1 phải khai báo:

- một chỉ mục (AB) với đồ thị (a)
- hai chỉ mục (A) và (B) với đồ thị (b), <u>không có</u> chỉ mục (AB)
- hai chỉ mục (AB) và (A) với đồ thị (c)
- hai chỉ mục (AB) và (B) với đồ thị (d)

Thuật toán trình bày ở trên đề nghị đồ thị quan hệ (a), với quan tâm hàng đầu là cung cấp một biểu diễn đồ thị không trùng lắp trên phương diện con đường truy xuất để người thiết kế có cái nhìn gọn nhất. Các dạng biến thể của đồ thị quan hệ, như các đồ thị (b), (c), (d) sẽ được cân nhắc ở một bước sau với các chỉ tiêu như ý đồ khai thác, tần suất khai thác, v.v….

IV.2. TRƯỜNG HỢP 2

Cho quan hệ :

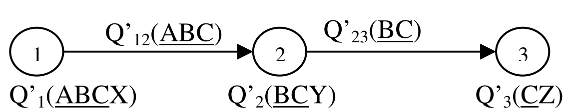
$C_0 = \langle Q_0(ABC\ XYZ),\ F_0 = \{ABC \rightarrow X,\ BC \rightarrow Y, C \rightarrow Z\} \rangle$

Và cấu trúc CSDL quan niệm:

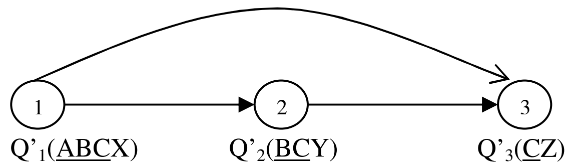
$C'_1 = \{\langle Q'_1\ (ABC\ X)\ ,\ F'_1 = \{ABC \rightarrow X\} \rangle ;$

$\langle Q'_2(BC\ Y)\ ,\ F'_2 = \{BC \rightarrow Y\} \rangle ;$

$\langle Q'_3(C\ Z)\ ,\ F'_3 = \{C \rightarrow Z\} \rangle \}$



(a)



(b)

- Hình IV.9 –Hai đồ thị quan hệ của C'_1–

Trường hợp thứ hai này tương tự như trường hợp trước trong các đồ thị quan hệ (c) và (d). Với đồ thị (IV.9.b), người sử dụng có thể tạo những bộ dạng (a,-,c, x), nghĩa là chỉ mới biết cặp giá trị (a,c) và chưa biết cặp giá trị của (A,B) tương ứng với "a". Ngược lại, với đồ thị (IV.9.a), người sử dụng chỉ có thể tạo hoặc (a,b,c,x) hoặc (a,-,-,x).

Khi khai báo cấu trúc vật lý của quan hệ Q_1, với (a) chỉ khai báo một chỉ mục (BC), với (b) phải khai báo hai chỉ mục (BC) và (C).

Thuật toán đề nghị đồ thị quan hệ (a).

IV.3. TRƯỜNG HỢP 3

Cho quan hệ:

$C''_0 = \langle Q''_0\ (ABCDE\ XYZ),$

$F''_0 = \{CDE \rightarrow X; AE \rightarrow CDB;$

$AD \rightarrow CY; C \rightarrow AB;$

$AB \rightarrow CZ\} \rangle$

Và cấu trúc CSDL quan niệm:

$C''_1 = \{\langle Q''_1\ (\underline{CDE} / \underline{AE}\ BX),$

$F''_1 = \{CDE \rightarrow ABX; AE \rightarrow CDBX;$

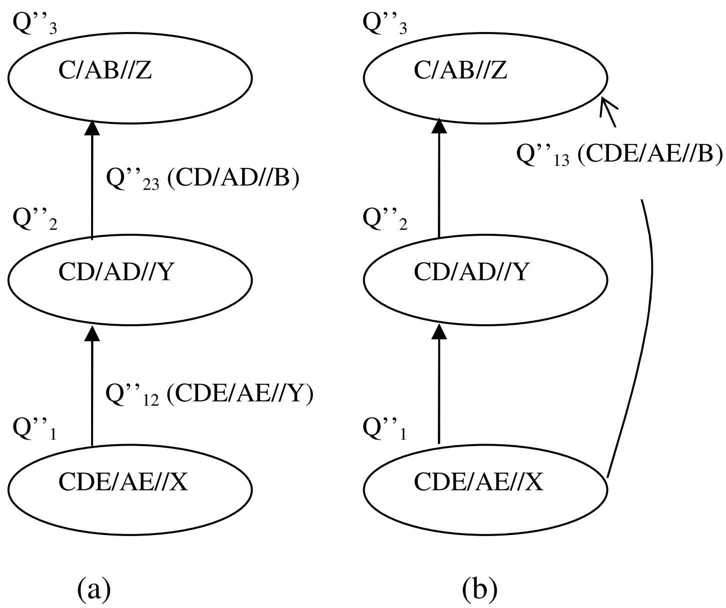
$C \rightarrow AB; AB \rightarrow C\} \rangle$

$\langle Q''_2(\underline{CD} / \underline{AD} / Y),$

$F''_2 = \{CD \rightarrow AY; AD \rightarrow CY; C \rightarrow A\} \rangle$

$\langle Q''_3(\underline{C} / \underline{AB})\ Z),$

$F''_3 = \{C \rightarrow ABZ \; ; \; A\,B \rightarrow CZ \} >\}$



- Hình IV.10 –Hai đồ thị quan hệ của C''_1–

Khác với trường hợp ở trên, các quan hệ của C''_1 không có sự lồng khoá toàn bộ, giữa hai quan hệ Q_1 và Q_3 , ngoài hai khoá lồng nhau là CDE và C, còn có phụ thuộc hàm không hiển nhiên $AE \rightarrow AB$. Cung (13) thể hiện phụ thuộc hàm này.

Thuật toán đề nghị đồ thị (b).

V. QUI TRÌNH TỔNG THỂ CỦA GIAI ĐOẠN THIẾT KẾ LOGIC

1. Từ một cấu trúc CSDL quan niệm được xác định trong giai đoạn thiết kế quan niệm, giai đoạn thiết kế lôgic sẽ biến đổi nó thành một đồ thị quan hệ. Đồ thị này phải hoàn toàn trung thực với cấu trúc quan niệm theo 3 loại tiêu chuẩn được đề ra là:

- bảo toàn nội dung CSDL.
- bảo toàn sự truy xuất trực tiếp
- bảo toàn tính hiệu quả trong việc kiểm tra ràng buộc toàn vẹn.

Đồ thị quan hệ là một dạng tóm tắt của đồ thị con đường truy xuất thô.

2. Giai đoạn thiết kế vật lý tiếp theo sẽ dựa trên đồ thị này để lựa chọn một đồ thị con đường truy xuất sẽ cài đặt.

BÀI TẬP

I. BÀI TẬP CHƯƠNG II

Bài tập II.1

Cho quan hệ R(A, B, C, D) như sau:

R(	A	B	C	D)
	a	1	X	2
	a	1	Y	2
	b	2	X	1
	b	2	Y	1

Cho biết phụ thuộc hàm nào liệt kê dưới đây được thỏa trong quan hệ ?

a) f_1: $A \rightarrow A$

b) f_2: $A \rightarrow B$

c) f_3: $A \rightarrow C$

d) f_4: $AC \rightarrow C$

e) f_5: $A \rightarrow D$

f) f_6: $D \rightarrow A$

Bài tập II.2

Cho lược đồ quan hệ (R,F) với R = ABCDE

và F = {$A \rightarrow C$, $BC \rightarrow D$, $D \rightarrow E$, $E \rightarrow A$}. Tính:

a) $(AB)^+$

b) $(BD)^+ - (D)^+$

Bài tập II.3

Cho lược đồ quan hệ (R,F) với R = ABCDEG

và F = {$B \rightarrow C$, $AC \rightarrow D$, $D \rightarrow G$, $AG \rightarrow E$}. Cho biết:

a) $AB \rightarrow G \in F^+$?

b) $BD \rightarrow AD \in F^+$?

Bài tập II.4

Cho các tập con X, Y, Z, V và một thuộc tính A của tập thuộc tính R. Hãy xác định một trong các quan hệ cao nhất các cặp tập phụ thuộc hàm sau đây bằng cách đặt dấu $\equiv$ hoặc dấu $\models$ vào chỗ dấu ? Giải thích vì sao.

a) {X→Y, Z→V} ? {XZ→YV}
b) {X→Y } ? {X→(Y-X)}
c) {X→Y} ? {XZ→Y}
d) {X→Y, Y→Z} ? {X→Z}
e) {X→Y, YZ→V} ? {XZ→V}
f) {X→Y} ? {XZ→(Y-V)}
g) {X→Y, X→Z} ? {X→YZ}
h) {X→YZ} ? {X→Y}
i) {X→YZ, Z→AV} ? {X→YZA}

Bài tập II.5

Tìm khóa của các lược đồ quan hệ (R,F) sau:

a) R = ABCD và
F = {AD→BC, B→A, D→C }

b) R = ABCDEGH và
F = {ABC→DE, BCD→G, ABH→EG, CE→GH}

Bài tập II.6

Cho lược đồ quan hệ (R,F) với R = ABCDEGH và
F = {CD→H, E→B, D→G, BH→E, CH→DG, C→A}

a) Tập ABD có phải là khóa của (R,F) không? Vì sao?
b) Tập CH có phải là khóa của (R,F) không? Vì sao?
c) Tìm một khóa của lược đồ quan hệ trên.

Bài tập II.7

Cho lược đồ quan hệ (R,F) với R = ABCDEGH và

F = {DE→G, H→C, E→A, CG→H, DG→EA, D→B}

a) Tập BCE có phải là khóa của (R,F) không? Vì sao?

b) Tìm một khóa của lược đồ quan hệ trên.

II. BÀI TẬP CHƯƠNG III

Bài tập III.1

Xét lược đồ quan hệ có các thuộc tính:

S : store

D : department

I : item

M : manager

Với các phụ thuộc hàm:

$SI \rightarrow D$, $SD \rightarrow M$

a) Tìm tất cả các khóa của SDIM

b) Chứng minh rằng SDIM đạt dạng chuẩn cấp hai nhưng không đạt dạng chuẩn cấp ba

Bài tập III.2

Xác định và giải thích dạng chuẩn cao nhất của các lược đồ quan hệ (R,F) sau:

a) R = ABCD và $F = \{A \rightarrow C, D \rightarrow B, C \rightarrow ABD\}$

b) R = ABCD và $F = \{D \rightarrow B, C \rightarrow A, B \rightarrow ACD\}$

c) R = ABCD và $F = \{CD \rightarrow B, A \rightarrow C, B \rightarrow ACD\}$

Bài tập III.3

Dùng kỹ thuật lập bảng để kiểm tra tính nối không mất của các phép phân rã sau:

a) Cho lược đồ quan hệ (R,F) với:

R = ABCD

$F = \{A \rightarrow B, AC \rightarrow D\}$

Và phân rã ρ = (AB, ACD)

b) Cho lược đồ quan hệ (R,F) với:

R = ABCDE

$F = \{A \rightarrow B, AC \rightarrow D\}$

Và phân rã ρ = (AB, ACD)

c) Cho lược đồ quan hệ (R,F) với:

R = ABCDEGH

F = {CD→H,E→B,D→G,BH→E,CH→DG,C→A}

Và phân rã ρ = (ABCDE, BCH, CDEGH)

Bài tập III.3

Đối với những tập phụ thuộc hàm khác nhau, phân rã

ρ = (AB, BC, CD)

Có thể có nối không mất hoặc không. Với mỗi tập phụ thuộc dưới đây, hãy chứng minh rằng phân rã trên có nối không mất hoặc cho một quan hệ (thể hiện) phản ví dụ

a) {A→B, B→C}

b) {B→C, C→D}

Bài tập III.4

Chứng minh rằng (AB, ACD, BCD) không phải là một phân rã có nối không mất ứng với tập phụ thuộc hàm {A→C, D→C, BD→A}

Bài tập III.5

Xét lược đồ quan hệ ABCD với các phụ thuộc

F = {A→B, B→C, D→B}

Giả sử chúng ta muốn tìm một phân rã BCNF có nối không mất

a) Nếu ở bước đầu tiên chúng ta phân rã ABCD thành ACD và BD thì những phụ thuộc hình chiếu trong hai lược đồ nầy là những phụ thuộc nào?

b) Những lược đồ nầy có dạng BCNF không? Nếu không thì cần phải phân rã tiếp như thế nào?

Bài tập III.6

Cho lược đồ quan hệ ABCD với các phụ thuộc hàm {A→B, B→C, A→, D→C}. Gọi ρ là phân rã (AB, AC, BD)

a) Tìm các phụ thuộc hình chiếu cho mỗi lược đồ của ρ

b) ρ có phải là phân rã có nối không mất ứng với các phụ thuộc đã cho hay không?

c) ρ có bảo toàn các phụ thuộc đã cho hay không?

Bài tập III.7

Cho F = {AB→C, A→B }

a) Tìm một phủ cực tiểu của F

b) Tập phụ thuộc hàm G = {A→, B→C} có phải là một phủ cực tiểu của F hay không? Nếu không hãy đưa ra một quan hệ (thể hiện) của lược đồ ABC thỏa F nhưng không thỏa G

Bài tập III.8

Cho F = {AB→C, A→D, BD→C}

a) Tìm phủ cực tiểu của F

b) Hãy đưa ra một phân rã ABCD thành hai lược đồ dạng 3NF có bảo toàn phụ thuộc (ứng với tập phụ thuộc hàm F)

c) Trình bày những phụ thuộc hình chiếu cho mỗi lược đồ tìm được

d) Kết quả của câu b) có phải là một phân rã có nối không mất hay không? Nếu không chúng ta có thể sửa đổi lược đồ như thế nào để phân rã có nối không mất và vẫn bảo toàn phụ thuộc

Bài tập III.9

Giả sử chúng ta có cơ sở dữ liệu của một công ty hoạt động đầu tư với các thuộc tính sau:

B : broker; người môi giới

O : office of a broker; văn phòng của người môi giới

I : investor; nhà đầu tư

S : stock; cổ phần

Q : quantity of stock owned by an investor; số lượng cổ phần của nhà đầu tư

D : divident paid by a stock; lãi của mỗi cổ phần

Và các phụ thuộc hàm:

$S \rightarrow D$, $I \rightarrow B$, $IS \rightarrow Q$, $B \rightarrow O$

a) Hãy tìm một khóa cho lược đồ quan hệ R=BOSQID

b) Lược đồ R có bao nhiêu khóa? Hãy chứng minh

c) Hãy tìm một phân rã R thành dạng Boyce-Codd có nối không mất

d) Hãy tìm một phân rã R thành dạng cấp ba có nối không mất và bảo toàn phụ thuộc

Bài tập III.10

Giả sử chúng ta biểu diễn lược đồ quan hệ R của bài tập III.9 bằng các lược đồ ISQ, IB, SD và ISO.

a) Tìm các hình chiếu của F trên các lược đồ con

b) Tìm một phủ cực tiểu cho hợp của các phụ thuộc được chiếu

c) Phân rã nầy có bảo toàn phụ thuộc hay không?

Bài tập III.11

Hãy phân rã lược đồ quan hệ (R,F) sau thành 3NF có nối không mất và bảo toàn phụ thuộc với:

R = MLTGSDP và

F = {M→T, GP→M, GT→P, MS→D, GS→P}

Với ngữ nghĩa sau:

M : môn học chuyên đề

L : lớp chuyên đề

T : thầy - giáo viên phụ trách chuyên đề

G : giờ học chuyên đề

S : sinh viên theo học chuyên đề

D : số đăng ký của sinh viên trong chuyên đề đó

P : phòng học dành cho chuyên đề

M→T : mỗi chuyên đề có một thầy phụ trách

GP→M : tại mỗi thời điểm, mỗi phòng học được dành cho không quá một môn

GT→P : tại mỗi thời điểm, mỗi thầy dạy trong không quá một phòng học

MS→D : mỗi sinh viên tham gia chuyên đề nào thì được cấp một mã số ghi danh theo chuyên đề đó

GS→P : tại mỗi thời điểm, mỗi sinh viên có mặt trong không quá một phòng học

Bài tập III.12

Cho lược đồ cơ sở dữ liệu 'Thựctập' :

SV (SV, HT, NS, QUE, HL)

DT (DT, TDT, CN, KP)

SD (SV, DT, NTT, KM, KQ)

Với các tập phụ thuộc hàm hình chiếu như sau:

F_{SV} = {SV→HT, NS, QUE, HL}

F_{DT} = {DT→TDT, CN, KP}

F_{SD} = {SV, DT→NTT, KQ; NTT→KM }

Hãy xác định dạng chuẩn cao nhất trong từng lược đồ quan hệ con: (SV, F_{SV}), (DT, F_{DT}) và (SD, F_{SD}). Nếu lược đồ quan hệ con nào chưa đạt 3NF, hãy phân rã thành 3NF có nối không mất và bảo toàn phụ thuộc.

Bài tập III.13

Xét cơ sở dữ liệu của Ship Voyages (công ty vận tải bằng đường thủy) có các thuộc tính sau:

S : ship name; tên tàu

T : type of ship; loại tàu

V : voyage identifier; mã số chuyến tàu

C : cargo carried by one ship on one voyage; khối lượng hàng hóa được vận chuyển

P : port; bến cảng

D : day; ngày

Giả sử rằng một chuyến tàu (voyage) có một chuỗi các sự kiện: lấy một loại hàng và phân phối hàng nầy cho các bến cảng. Một tàu trong một ngày chỉ được ghé qua một cảng, vì thế chúng ta có thể giả định có các phụ thuộc hàm sau:

$S \rightarrow T$, $V \rightarrow SC$, $SD \rightarrow PV$

a) Hãy tìm một phân rã thành dạng BCNF có nối không mất

b) Hãy tìm một phân rã thành dạng 3NF có nối không mất và bảo toàn phụ thuộc

c) Hãy giải thích tại sao không có phân rã thành BCNF vừa có nối không mất vừa bảo toàn phụ thuộc cho CSDL nầy.

TÀI LIỆU THAM KHẢO

1. D. Maier. The Theory of Relational Databases.
   Computer Science Press, Rockville, 1983
2. J. Ullman. Database and Knowledge Base Systems.
   Vol.1,2,3 Computer Science Press, 1983
   (Bản dịch tiếng Việt của Trần Đức Quang, NXB Thống kê Hà nội 1999)
3. Lê Tiến Vương. Nhập môn cơ sở dữ liệu.
   NXB Thống kê Hà nội 2000. Tái bản lần 5
4. Đỗ Phúc, Nguyễn Đăng Tỵ. Giáo trình cơ sở dữ liệu.
   Đại học Quốc gia tp.HCM.
5. Huỳnh Thị Hà, Nguyễn Đình Loan Phương. Giáo trình cơ sở dữ liệu quan hệ.
   Đại học Quốc gia tp.HCM, 2003.
6. Đồng Thị Bích Thủy. Giáo trình thiết kế cơ sở dữ liệu.
   Đại học Quốc gia tp.HCM.
7. Nguyễn Xuân Huy, Lê Hoài Bắc. Bài tập cơ sở dữ liệu.
   NXB Thống kê Hà nội 2003.