PGS.TS. Vũ Đức Thi

Giáo trình cơ sở dữ liệu

Bài Giảng

Hà Nội

Lời nói đầu

Cơ sở dữ liệu là một lĩnh vực phát triển mạnh của công nghệ thông tin. Cùng với sự phát triển công nghệ thông tin ở nước ta, việc sử dụng các kiến thức về cơ sở dữ liệu vào thực tiễn ngày càng trở lên cần thiết.

Trong bài giảng này chúng tôi cung cấp cho sinh viên những kiến thức cơ bản nhất về cơ sở dữ liệu. Mục tiêu chính là với số kiến thức cơ bản này sinh viên có thể ứng dụng các kiến thức về cơ sở dữ liệu vào thực tiễn và tiếp tục nghiên cứu học tập được các môn tin học khác.

Giáo trình gồm 4 chương chính (Ngoài chương mở đầu và tài liệu tham khảo).

Chương 2 cung cấp cho sinh viên những kiến thức cơ bản về cơ sở dữ liệu, mà cụ thể là về cơ sở dữ liệu quan hệ. Trong chương này, chúng tôi trình bày những khái niệm cơ bản nhất của cơ sở dữ liệu quan hệ, cũng như những thuật toán thiết kế chúng.

Chương 3 trình bày các kiến thức liên quan đến các dạng chuẩn.

Chương 4 giới thiệu các phép toán xử lí các bảng (quan hệ).

Chương 5 và chương 6 là các chương trình bày các ứng dụng của cơ sở dữ liệu vào thực tiễn.

Trong chương 5 chúng tôi nêu một số các ứng dụng của cơ sở dữ liệu trong các hệ quản trị cơ sở dữ liệu hiện có. Trong đó có những vấn đề liên quan đến các thực thể, các khoá, các dạng chuẩn trong các hệ quản trị cơ sở dữ liệu.

Chương 6 trình bày một số các công đoạn xây dựng các dự án thiết kế tổng thể các hệ thống thông tin.

Trong chương 7, chúng tôi trình bày một số các kiến thúc cơ bản về thuật toán và độ phức tạp thuật toán. Những kiến thức này giúp cho bạn đọc tiếp thu các kiến thức của các chương trên.

Giáo trình này phục vụ cho các sinh viên ngành công nghệ thông tin hoặc các cán bộ đang công tác trong lĩnh vực tin học muốn bổ xung kiến thức cho mình.

Tại tất cả các trường đại học có giảng dạy về tin học, cơ sở dữ liệu là môn học chính cho các sinh viên khoa công nghệ thông tin. Vì thế giáo trình này có thể làm tư liệu học tập cho sinh viên hệ cử nhân tin học, cử nhân cao đẳng tin học, kĩ sư tin học, hoặc có thể làm tài liệu tham khảo cho các học viên cao học, nghiên cứu sinh và các giảng viên tin học.

PGS.TS. Vũ Đức Thi

Chương mở đầu

Cơ sở dữ liệu (CSDL) là một trong những lĩnh vực được tập trung nghiên cứu và phát triển của công nghệ thông tin, nhằm giải quyết các bài toán quản lí, tìm kiếm thông tin trong những hệ thống lớn, đa dạng, phức tạp cho nhiều người sử dụng trên máy tính điện tử. Cùng với sự ứng dụng mạnh mẽ công nghệ thông tin vào đời sống xã hội, kinh tế, quốc phòng ...Việc nghiên cứu CSDL đã và đang phát triển ngày càng phong phú và hoàn thiện. Từ những năm 70, mô hình dữ liệu quan hệ do E.F. Codd đưa ra với cấu trúc hoàn chỉnh đã tạo lên cơ sở toán học cho các vấn đề nghiên cứu lí thuyết về CSDL. Với ưu điểm về tính cấu trúc đơn giản và khả năng hình thức hoá phong phú, CSDL quan hệ dễ dàng mô phỏng các hệ thống thông tin đa dạng trong thực tiễn, tạo điều kiện lưu trữ thông tin tiết kiệm, có tính độc lập dữ liệu cao, dễ sửa đổi, bổ sung cũng như khai thác dữ liệu. Mặt khác, việc khai thác và áp dụng các kĩ thuật tổ chức và sử dụng bộ nhớ cho phép việc cài đặt các CSDL quan hệ đưa lại hiệu quả cao và làm cho CSDL quan hệ chiếm ưu thế trên thị trường.

Nhiều hệ quản trị CSDL đã được xây dựng và đưa vào sử dụng rộng rãi như : DBASE, **FOXBASE,**

FOXPRO, PARADOX, ORACLE, MEGA, IBM DB2, SQL for WINDOWS NT...

Mô hình dữ liệu quan hệ đặt trọng điểm hàng đầu không phải là khai thác các tiềm năng của máy mà ở sự mô tả trực quan dữ liệu theo quan điểm của người dùng, cung cấp một mô hình dữ liệu đơn giản, trong sáng, chặt chẽ, dễ hiểu và tạo khả năng tự động hoá thiết kế CSDL quan hệ. Có thể nói lí thuyết thiết kế và cài đặt CSDL, nhất là mô hình dữ liệu quan hệ đã phát triển ở mức độ cao và đạt được những kết quả sâu sắc. Hàng loạt vấn đề đã được nghiên cứu giải quyết như:

- Lí thuyết thiết kế CSDL, các phương pháp tách và tổng hợp các lược đồ quan hệ theo tiêu chuẩn không tổn thất thông tin hay bảo toàn tính nhất thể của các ràng buộc trên dữ liệu .

- Các loại ràng buộc dữ liệu, cấu trúc và các tính chất của chúng, ngữ nghĩa và khả năng áp dụng phụ thuộc dữ liệu ví dụ như phụ thuộc hàm, phụ thuộc đa trị, phụ thuộc kết nối, phụ thuộc lôgic...

- Các vấn đề tối ưu hoá: ở mức vật lí trong việc tổ chức quản lí các tệp; ở mức đường truy nhập với các tệp chỉ số hay các danh sách sắp xếp; ở mức lôgic trên cơ sở rút gọn các biểu thức biểu diễn các câu hỏi, ...vv

.....

Trong Giáo trình này sẽ trình bày một số kiến thức cơ bản nhất về CSDL bao gồm các kiến thức liên quan đến phụ thuộc hàm, khoá và dạng chuẩn, các thuật toán nhận dạng và thiết kế chúng, việc xây dựng các khái niệm này trong các hệ CSDL lớn như MEGA, ORACLE...., việc nghiên cứu và áp dụng chúng để xây dựng các dự án thiết kế tổng thể các hệ thống CSDL hiện nay.

Chương 2
Các kiến thức cơ bản về cơ sở dữ liệu

2.1. Khát quát về mô hình dữ liệu

Thông thường đối với việc thiết kế và xây dựng các hệ thông tin quản lí, chúng ta cần xử lí các file dữ liệu. Những file này bao gồm nhiều bản ghi (record) có cùng một cấu trúc xác định (loại bản ghi). Đồng thời, mỗi bản ghi được phân chia thành các trường dữ liệu (fild). Một cơ sở dữ liệu là một hệ thống các file dữ liệu, mỗi file này có cấu trúc bản ghi khác nhau, nhưng về mặt nội dung có quan hệ với nhau. Một hệ quản trị cơ sở dữ liệu là một hệ thống quản lí và điều hành các file dữ liệu. Nói chung một hệ quản trị cơ sở dữ liệu thường có những đặc tính sau :

- Có tính độc lập với các công cụ lưu trữ,
- Có tính độc lập với các chương trình phần mềm của người sử dụng (có nghĩa là các ngôn ngữ lập trình khác nhau có thể được dùng trong hệ này),
- Có khả năng tại một thời điểm truy nhập vào nhiều nơi trong hệ này ,
- Có khả năng khai thác tốt tiềm năng của máy,

- Người dùng với kiến thức tối thiểu cúng có thể xử dụng được hệ này,

- Bảo đảm an toàn dữ liệu và bảo mật dữ liệu,

- Thuận lợi và mềm dẻo trong việc bổ xung, loại bỏ, thay đổi dữ liệu

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

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

- Thực thể (Entity): Đó là đối tượng có trong thực tế mà chúng ta cần mô tả các đặc trưng của nó.

- Thuộc tính: Đó là các dữ liệu thể hiện các đặc trưng của thực thể.

- Ràng buộc: Đó là các mối quan hệ lôgic của các thực thể.

Tuy vậy, ba thành phần cơ bản trên được thể hiện ở hai mức :

- Mức loại dữ liệu (Type): Đó là sự khái quát hoá các ràng buộc, các thuộc tính, các thực thể cụ thể.

- Mức thể hiện: Đó là một ràng buộc cụ thể, hoặc là các giá trị thuộc tính, hoặc là một thực thể cụ thể

Thông thường chúng ta sẽ nhận được các loại dữ liệu (Type) của các đối tượng cần khảo sát trong quá trình phân tích các thể hiện cụ thể của chúng.

yếu tố quan trọng nhất của cấu trúc cơ sở dữ liệu là dạng cấu trúc dữ liệu mà trong đó các mối quan hệ giữa các dữ liệu lưu trữ được mô tả. Có thể thấy rằng loại dữ liệu nền tảng của việc mô tả các mối quan hệ là loại bản ghi (Record type). Bởi vì các ràng buộc giữa các loại bản ghi tạo ra bản chất cấu trúc của cơ sở dữ liệu. Vì thế, dựa trên việc xác định các ràng buộc giữa các loại dữ lịêu được cho như thế nào mà chúng ta phân loại các mô hình dữ liệu. Có nghĩa là từ cách nhìn của người xử dụng việc mô tả các dữ liệu và các ràng buộc giữa các dữ liệu được thực hiện như thế nào. Trên thực tế chúng ta phân biệt hai loại mô hình dữ liệu:

- Mô hình dữ liệu mạng: Trong đó chúng ta thể hiện trực tiếp các ràng buộc tuỳ ý giữa các loại bản ghi,

- Mô hình dữ liệu quan hệ: Trong mô hình này các ràng buộc trên được thể hiện qua các quan hệ (bảng).

Mô hình dữ liệu quan hệ là một công cụ rất tiện lợi để mô tả cấu trúc lôgic của các cơ sở dữ liệu. Như vậy, ở mức lôgic mô hình này bao gồm các file được biểu diễn dưới dạng các bảng. Do đó đơn vị của CSDL quan hệ là một bảng (Một quan hệ được thể hiện trong Định nghĩa 1), trong đó các dòng của bảng là các bản ghi dữ liệu cụ thể (Đó là các thể hiện cụ thể của loại bản ghi), còn tên các cột là các thuộc tính.

Theo cách nhìn của người xử dụng thì một cơ sở dữ liệu quan hệ là một tập hợp các bảng biến đổi theo thời gian.

2.2. Các khái niệm cơ bản và hệ tiên đề Armstrong:

Trong mục này, chúng ta trình bày những khái niệm cơ bản nhất về mô hình dữ liệu quan hệ của E.F. Codd. Những khái niệm cơ bản này gồm các khái niệm về quan hệ, thuộc tính, phụ thuộc hàm, hệ tiên đề Armstrong, khóa, dạng chuẩn....

Những khái niệm này đóng vai trò rất quan trọng trong mô hình dữ liệu quan hệ. Chúng được áp dụng nhiều trong việc thiết kế các hệ quản trị cơ sở dữ liệu hiện nay.

Những khái niệm này có thể tìm thấy trong [1,2,3,4,7,9,10,15,16,17].

Định nghĩa 1. (Quan hệ)

Cho R = $\{a_{1, \dots, a_n}\}$ là một tập hữu hạn và không rỗng các thuộc tính. Mỗi thuộc tính a_i có miền giá trị là D_{ai}. Khi đó r là một tập các bộ $\{h_1, ..., h_m\}$ được gọi là một quan hệ trên R với h_j (j = 1,...m) là một hàm :

$h_j : R \rightarrow \cup D_{ai}$

$a_i \in R$

sao cho: $h_j (a_i) \in D_{ai}$

Chúng ta có thể biểu diễn quan hệ r thành bảng sau:

	a_1	a_2		a_n
h_1	$h_1(a_1)$	$h_1(a_2)$	.....	$h_1(a_n)$
h_2	$h_2(a_1)$	$h_2(a_2)$	.....	$h_2(a_n)$
.	.....			

.

.

h_m $h_m(a_1)$ $h_m(a_2)$ ..... $h_m(a_n)$

Ví dụ: Trong một cơ quan, chúng ta quản lý nhân sự theo biểu gồm các thuộc tính sau:

Nhân sự

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

Chúng ta quy định kích thước cho các thuộc tính (các trường) như sau:

Tên thuộc tính	Kiểu	Kích thước
STT	Kí tự	3
HOTEN	Ký tự	30
GIOITINH	Ký tự	3
NAMSINH	Số	4
TRINHDO	Ký tự	10
LUONG	Số	7

Có nghĩa là qui định cho thuộc tính STT là các dãy gồm 3 kí tự, thuộc tính HOTEN là các dãy gồm 30 kí tự, ..., cho thuộc tính LUONG là các số có nhiều nhất 7 chữ số.

Như vậy chúng ta có tập thuộc tính

NHANSU = {STT, HOTEN, GIOITINH, NAMSINH, TRINHDO, LUONG}

ở đây D_{STT} là tập các dãy gồm 3 kí tự,...., D_{LUONG} là tập các số có nhiều nhất 7 chữ số.

Khi đó chúng ta có quan hệ r = {h_1, h_2,..., h_{120}}, ở đây ví dụ như đối với bản ghi thứ 2 (dòng thứ 2) chúng ta có:

h_2(STT) = 002, h_2 (HOTEN) = Nguyễn Kim ánh

h_2(GIOITINH) = Nữ, h_2 (NAMSINH) = 1971

h_2(TRINHDO) = Trung cấp, h_2 (LUONG) = 240000

Định nghĩa 2. (Phụ thuộc hàm)

1. Cho R = $\{a_1,...,a_n\}$ là tập các thuộc tính, r = $\{h_1,...,h_m\}$ là một quan hệ trên R, và A, B $\subseteq$ R.

2. Khi đó chúng ta nói A xác định hàm cho B hay B phụ thuộc hàm vào A trong r (Kí pháp A $\overset{f}{\underset{r}{>}}$ B) nếu

$(\forall h_i,h_j \in r)((\forall a \in A)(h_i(a)= h_j(a)) \Rightarrow(\forall b \in B)(h_i(b)=h_j(b)))$

Đặt F_r = { (A,B): A,B $\subseteq$ R, A $\overset{f}{\underset{r}{>}}$ B }. Lúc đó F_r được gọi là họ đầy đủ các phụ thuộc hàm của r.

Khái niệm phụ thuộc hàm miêu tả một loại ràng buộc (phụ thuộc dữ liệu) xảy ra tự nhiên nhất giữa các tập thuộc tính. Dù hiện nay đã có nhiều loại phụ thuộc dữ liệu được nghiên cứu, xong về cơ bản các hệ quản trị cơ sở dữ liệu lớn sử dụng phụ thuộc hàm.

Định nghĩa 3.

Phụ thuộc hàm (PTH) trên tập các thuộc tính R là một dãy kí tự có dạng A $\rightarrow$B, ở đây A,B $\subseteq$ R. Chúng

ta nói PTH $A \rightarrow B$ đúng trong quan hệ r if $A \overset{f}{\underset{r}{>}} B$. Chúng ta cũng nói rằng r thoả mãn

$A \rightarrow B$.

Dễ thấy, F_r là tập tất cả các PTH đúng trong r.

Chú ý: Trong giáo trình này chúng ta có thể viết (A,B) hoặc $A \rightarrow B$ thay cho $A \overset{f}{\underset{r}{>}} B$ mà không bị lẫn về mặt kí pháp.

Định nghĩa 4. (Hệ tiên đề của Armstrong)

Giả sử R là tập các thuộc tính và kí pháp P(R) là tập các tập con của R. Cho $Y \subseteq P(R) \times P(R)$. Chúng ta nói Y là một họ f trên R nếu đối với mọi $A, B, C, D \subseteq R$

(1) $(A,A) \in Y$,

(2) $(A,B) \in Y, (B,C) \in Y \Rightarrow (A,C) \in Y$,

(3) $(A,B) \in Y, A \subseteq C, D \subseteq B \rightarrow (C,D) \in Y$,

(4) $(A,B) \in Y, (C,D) \in Y \Rightarrow (A \cup C, B \cup D) \in Y$.

Rõ ràng, F_r là một họ f trên R.

Trong [l] A. A. Armstrong đã chứng minh một kết quả rất quan trọng như sau : Nếu Y là một họ f

bất kì thì tồn tại một quan hệ r trên R sao cho F_r = Y.

Kết quả này cùng với định nghĩa của phụ thuộc hàm chứng tỏ rằng hệ tiên đề Armstrong là đúng đắn và đầy đủ.

Mặt khác, hệ tiên đề này cho ta những đặc trưng của họ các phụ thuộc hàm, mà các đặc trưng này không phụ thuộc vào các quan hệ (bảng) cụ thể. Nhờ có hệ tiên đề này các công cụ của toán học được áp dụng để nghiên cứu làm sáng tỏ cấu trúc lôgic của mô hình dữ liệu quan hệ. Đặc biệt chúng ta xử dụng công cụ thuật toán để thiết kế các công đoạn xây dựng các hệ quản trị cơ sở dữ liệu.

Chúng ta đưa ra ví dụ chỉ ra có nhiều quan hệ khác nhau xong các họ đầy đủ các phụ thuộc hàm của chúng lại như nhau.

Cho r_1 và r_2 là các quan hệ sau:

r_1 =

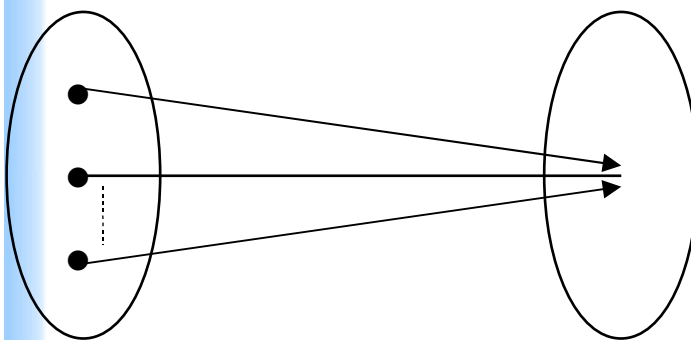
a	b
0	0
1	1
2	1
3	2

r_2 =

a	b
0	0
1	1
2	1
3	1

Có thể thấy r_1 và r_2 khác nhau nhưng $F_{r1} = F_{r2}$.

Như vậy, tương quan giữa lớp các quan hệ với lớp các họ phụ thuộc hàm có thể được thể hiện bằng hình vẽ sau.



Lớp các quan hệ Lớp các phụ thuộc hàm

Định nghĩa 5.

Một hàm L : P(R) $\rightarrow$ P(R) được gọi là một hàm đóng trên R nếu với mọi A, B $\in$ P(R) thì :

- A $\subseteq$ L(A),
- Nếu A $\subseteq$ B thì L(A) $\subseteq$ L(B),
- L(L(A)) = L(A).

Định lí 6.

Nếu F là một họ f và chúng ta đặt

$L_F = \{a : a \in R$ và $(A, \{a\}) \in F\}$

thì L_F là một hàm đóng. Ngược lại, nếu L là một hàm đóng thì tồn tại duy nhất một họ f F trên R sao cho $L = L_F$, ở đây

$F = \{ (A,B) : A, B \subseteq R , B \subseteq L(A) \}$.

Như vậy, chúng ta thấy có một tương ứng 1-1 giữa lớp các hàm đóng và lớp các họ f . Chúng ta có hình vẽ sau

Lớp các họ phụ thuộc hàm
Lớp các hàm đóng

Định lí 6 chỉ ra rằng để nghiên cứu phân tích các đặc trưng của họ các phụ thuộc hàm chúng ta có thể dùng công cụ hàm đóng.

Sau này trong mục 2.3 chúng tôi sẽ trình bày nhiều công cụ nữa để nghiên cứu cấu trúc lôgic của họ các phụ thuộc hàm.

Định nghĩa 7. (Sơ đồ quan hệ)

Chúng ta gọi sơ đồ quan hệ (SĐQH) s là một cặp <R,F>, ở đây R là tập các thuộc tính và F là tập các phụ thuộc hàm trên R. Kí pháp F^+ là tập tất cả các PTH được dẫn xuất từ F bằng việc áp dụng các qui tắc trong Định nghĩa 4.

Đặt $A^+ = \{a: A \rightarrow \{a\} \in F^+\}$. A^+ được gọi là bao đóng của A trên s.

Có thể thấy rằng $A \rightarrow B \in F^+$ nếu và chỉ nếu $B \subseteq A^+$.

Tương tự chúng ta đặt $A_r^+ = \{a : A \xrightarrow{f}_r \{a\}\}$. A_r^+ được gọi là bao đóng của A trên r.

Theo [1] chúng ta có thể thấy nếu s = <R,F> là sơ đồ quan hệ thì có quan hệ r trên R sao cho $F_r=F^+$. Quan hệ r như vậy chúng ta gọi là quan hệ Armstrong của s.

Trong trường hợp này hiển nhiên các PTH của s đúng trong r.

Định nghĩa 8. (Khoá)

Giả sử r là một quan hệ , s = <R,F> là một sơ đồ quan hệ, Y là một họ f trên R, và $A \subseteq R$. Khi đó A là một khoá của r (tương ứng là một khoá của s, một khoá của Y) nếu $A \xrightarrow{f}_{r} R$ ($A \rightarrow R \in F^+$, $(A,R) \in Y$). Chúng ta gọi A là một khoá tối tiểu của r (tương ứng của s, của Y) nếu

- A là một khoá của r (s, Y),

- Bất kì một tập con thực sự của A không là khoá của r (s, Y).

Chúng ta kí pháp K_r, (K_s, K_y) tương ứng là tập tất cả các khoá tối tiểu của r (s, Y).

Chúng ta gọi K (ở đây K là một tập con của P(R)) là một hệ Sperner trên R nếu với mọi $A,B \in K$ kéo theo $A \subseteq B$).

Có thể thấy K_r,K_s, K_y là các hệ Sperner trên R.

Định nghĩa 9.

Giả sử K là một hệ Sperner trên R. Chúng ta định nghĩa tập các phản khoá của K, kí pháp là K^{-1}, như sau:

$K^{-1} = \{A \subset R : (B \in K) \Rightarrow (B \not\subseteq A)$ and $(A \subset C) \Rightarrow (\exists B \in K)(B \subseteq C)\}$

Dễ thấy K^{-1} cũng là một hệ Sperner trên R.

Tập phản khoá đóng vai trò rất quan trọng trong quá trình nghiên cứu cấu trúc lôgic của các họ phụ thuộc hàm, khóa, dạng chuẩn, quan hệ Armstrong, đặc biệt đối với các bài toán tổ hợp trong mô hình dữ liệu quan hệ.

Trong [5] người ta đã nêu ra rằng nếu s = <R,F> là một sơ đồ quan hệ trên R, thì K_s là hệ Sperner trên R. Ngược lại, nếu K là một hệ Sperner bất kì trên R, thì tồn tại một sơ đồ quan hệ s sao cho $K_s = K$.

Ví dụ: Cho $K = \{ A_1,.....,A_m \}$ là một hệ Sperner. Khi đó s = <R,F>, ở đây $F = \{ A_1 \rightarrow R, ... , A_m \rightarrow R\}$ là sơ đồ quan hệ mà $K_s = K$.

Nhận xét :

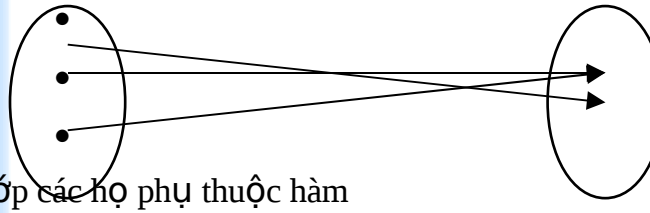
- Có thể cho ví dụ chỉ ra rằng có nhiều sơ đồ quan hệ khác nhau nhưng tập các khoá tối tiểu của chúng giống nhau. Có nghĩa là tồn tại

$s_1 = <R,F_1>, ..., s_t = <R,F_t>$ ($2 \leq t$) mà $F_1^+ \neq .... \neq F_t^+$, nhưng

$K_{s1} = ... = K_{st}$.

Tất nhiên, nếu $F_1 = F_2$ thì $K_{s1} = K_{s2}$.

Mối quan hệ giữa lớp họ phụ thuộc hàm và lớp các hệ Sperner thể hiện qua hình vẽ sau



Lớp các họ phụ thuộc hàm
Lớp các hệ Sperner

- Nếu K đóng vai trò là một tập các khoá tối tiểu của một sơ đồ quan hệ nào đó, thì theo định nghĩa K^{-1} là tập tất cả các tập không phải khoá lớn nhất.

Trong Giáo trình này, chúng ta qui ước rằng nếu hệ Sperner đóng vai trò là tập các khoá tối tiểu (tập các phản khoá), thì hệ này không rỗng (không chứa R).

Định nghĩa 10.

Cho $I \subseteq P(R)$. Khi đó I được gọi là nửa dàn giao nếu

$R \in I$ và $A,B \in I \Rightarrow A \cap B \in I$.

Giả sử $M \subseteq P(R)$. Đặt $M^+ = \{\cap M' : M' \subseteq M\}$. Khi đó chúng ta nói rằng M là một hệ sinh của I nếu $M^+ = I$.

Chú ý rằng $R \in M^+$ nhưng R không là một phần tử của M, bởi vì chúng ta theo thông lệ cho R là giao của một tập rỗng các tập con của M.

Kí pháp $N_I = \{A \in I : A \neq \cap \{A' \in I : A \subset A'\}\}$.

Trong [4] người ta đã chỉ ra rằng N_I là hệ sinh nhỏ nhất và duy nhất của I. Có nghĩa là đối với mọi hệ sinh N' của I chúng ta có $N_I \subseteq N'$.

Định nghĩa 11.

Cho r là một quan hệ trên R. Chúng ta đặt $E_r = \{E_{ij} : 1 \leq i \leq j \leq |r|\}$, ở đây $E_{ij} = \{a \in R: h_i(a) = h_j(a)\}$. E_r được gọi là hệ bằng nhau của r.

Đặt $M_r = \{ A \in P(R) : \exists E_{ij} = A, \nexists E_{pq}: A \subset E_{pq}\}$. Khi đó chúng ta gọi M_r là hệ bằng nhau cực đại của r.

Sau này ta sẽ thấy hệ bằng nhau và hệ bằng nhau cực đại được dùng rất nhiều trong các thuật toán thiết kế.

Mối quan hệ giữa lớp các quan hệ và lớp các phụ thuộc hàm đóng một vai trò quan trọng trong quá trình nghiên cứu cấu trúc lôgic của lớp các phụ thuộc hàm.

Định nghĩa 12.

Cho trước r là một quan hệ r và F là một họ f trên R. Chúng ta nói rằng r là thể hiện họ F nếu $F_r = F$. Chúng ta cũng có thể nói r là một quan hệ Armstrong của F.

Bây giờ, chúng ta đưa ra một điều kiện cần và đủ để một quan hệ là thể hiện một họ f cho trước.

Định lý 13.

Giả sử r = $\{h_1, ..., h_m\}$ là một quan hệ, F là một họ f trên R thì r thể hiện F nếu và chỉ nếu với mọi $A \subseteq R$

$$L_F(A) = \begin{cases} \cap E_{ij} \text{ nếu tồn tại } E_{ij} \in E_r : A \subseteq E_{ij}, \\ A \subseteq E_{ij} \\ R \quad \text{ngược lại.} \end{cases}$$

ở đây $L_F(A) = \{a \in R : (A, \{a\}) \in F\}$ và E_r là hệ bằng nhau của r.

Lời giải: Đầu tiên chúng ta chứng minh rằng trong một quan hệ r bất kì với mọi $A \subseteq R$

$$L_{Fr}(A) = \begin{cases} \cap E_{ij} \text{ nếu tồn tại } E_{ij} \in E_r : A \subseteq E_{ij}, \\ A \subseteq E_{ij} \\ R \quad \text{ngược lại.} \end{cases}$$

Giả sử đầu tiên chúng ta công nhận rằng A là một tập mà không có $E_{ij} \in E_r$ với $A \subseteq E_{ij}$ với mọi $h_i, h_j \in r$, $a \in A : h_i(a) = h_j(a)$. Theo định nghĩa của phụ thuộc hàm điều này kéo theo $A \rightarrow R$ và bởi định nghĩa của L_{Fr} ta thu được $L_F(A) = R$. Rõ ràng là

$$L_{Fr}(\varnothing) = \cap E_{ij}$$

$$E_{ij} \in E_r$$

Nếu $A \neq \varnothing$ và có một $E_{ij} \in E_r$ mà $A \subseteq E_{ij}$ thì chúng ta đặt

$$V = \{ E_{ij} : A \subseteq E_{ij}, E_{ij} \in E_r \}$$

và $E = \cap E_{ij}$

$$E_{ij} \in V$$

Dễ dàng nhận thấy rằng $A \subseteq E$. Nếu $V = E_r$ thì chúng ta nhận thấy rằng $(A,E) \in F_r$ nếu $V \neq E_r$ thì có thể coi như với mọi $E_{ij} \in V$ chúng ta có

(Với mọi $a \in A$) $(h_i(a) = h_j(a)) \rightarrow$ (Với mọi $b \in B$) $(h_i(b) = h_j(b))$ và với mọi $E_{ij} \notin V$ có một $a \in A$ mà $h_i(a) \neq h_j(a)$. Như vậy, $(A, E) \in F_r$.

Từ định nghĩa của L_{Fr} ta có $E \subseteq L_{Fr}(A)$. bởi vì r là một quan hệ trên R, chúng ta có $E \subset R$. Sử dụng $A \subseteq E \subseteq L_{Fr}(A)$ ta thu được $(E, L_{Fr}(A)) \in F_r$.

Bây giờ, ta giả sử rằng c là một thuộc tính mà $c \notin E$. Khi đó có một $E_{ij} \in V$ mà $c \notin E_{ij}$. Điều này kéo theo sự tồn tại của một cặp $h_i, h_j \in r$ mà với mọi $b \in E$: $h_i(b) = h_j(b)$ nhưng $h_i(c) \neq h_j(c)$. Có thể thấy rằng theo định nghĩa phụ thuộc hàm $(E \cup \{c\})$ không phụ thuộc vào E. Như vậy, với mọi thuộc tính $c \notin E$ ta có $(E, E \cup \{c\}) \notin F_r$. Bằng định nghĩa của L_{Fr} ta thu được :

$$L_{Fr}(A) = \bigcap E_{ij}$$

$$E_{ij} \in V$$

Trên cơ sở Định lí 6 chúng ta dễ dàng thấy rằng $F_r = F$ nếu và chỉ nếu $L_{Fr} = L_F$. □

Giả sử L là một hàm đóng. Đặt $Z(L) = \{A \subseteq R : L(A) = A\}$.

Rõ rằng, Z(L) là tập đóng với phép giao.

Có thể thấy là với mọi E_{ii} ($E_{ii} \in E_r$), chúng ta có $E_{ii} \in Z(L_{Fr})$, có nghĩa là $E^+_r \subseteq Z(L_{Fr})$

Nhờ Định lý 13 chúng ta có $Z(L_{Fr}) \subseteq E^+_R$. Như vậy chúng ta có

Hệ quả 14.

Giả sử r quan hệ, F là một họ f trên R. Khi đó r thể hiện F nếu và chỉ nếu $Z(L_F) = E^+_R$.

Trong [5] người ta đã chỉ ra rằng nếu cho một hệ Sperner không rỗng tuỳ ý K thì tồn tại một quan hệ r để $K = K_r$.

Bây giờ, chúng ta đưa ra một định nghĩa dưới đây

Định nghĩa 15.

Cho trước quan hệ r và hệ Sperner K trên R. Chúng ta nói rằng r thể hiện K nếu $K_r = K$.

Định nghĩa 16.

Cho F là một họ f trên R, và (A,B) là một phần tử của F. Chúng ta nói (A,B) là một phụ thuộc có vế phải cực đại của F nếu với mọi B' (B B') và (A,B') ∈ F kéo theo B = B'.

Chúng ta kí pháp M(F) là tập tất cả các phụ thuộc có vế phải cực đại của F. Chúng ta nói rằng B là vế phải cực đại của F nếu có A sao cho (A,B) ∈ M(F). Kí pháp I(F) là tập tất cả các vế phải cực đại của F.

Dưới đây chúng ta cho một điều kiện cần và đủ để một quan hệ thể hiện một hệ Sperner.

Định lý 17.

Giả sử K là một hệ Sperner không rỗng, r là một là một quan hệ trên R. Khi đó r thể hiện K nếu và chỉ nếu $K^{-1} = M_r$, ở đây M_r là hệ bằng nhau cực đại của r.

Lời giải: Có thể xem như là nếu K là một hệ Sperner không rỗng thì K^{-1} tồn tại. Mặt khác, K và K^{-1} là xác định duy nhất. Cho nên, chúng ta có $K_r = K$ nếu và chỉ nếu $K_r^{-1} = K^{-1}$.

Bây giờ chúng ta có thể chỉ cần chứng minh rằng $K_r^{-1} = M_r$. Rõ ràng, F_r là một họ f trên R. Đầu tiên chúng ta có thể giả thiết rằng A là một phản khoá của K_r. Rõ ràng A ≠ R. Nếu có một B sao cho

$A \subset B$ và $A \rightarrow B$, thì bằng định nghĩa của phản khoá, chúng ta có $B \rightarrow R$ và $A \rightarrow R$. Đây là một điều phi lý. Vì vậy $A \in I(F_r)$. Nếu có một B' sao cho $B' \neq R$, $B' \in I(F_r)$ và $A \subset B'$, thì B' là một khoá của r. Đây là một điều mâu thuẫn $B' \neq R$. Do đó $A \in I(F_r)$ - R và không tồn tại B' ($B' \in I(F_r)$ - R) để $A \subset B'$.

Mặt khác, theo định nghĩa của một quan hệ, $R \notin M_r$. Rõ ràng, $E_{ij} \in I(F_r)$. Như vậy chúng ta có $M_r \subseteq I(F_r)$. Nếu Dlà một tập sao cho $\forall C \in M_r : D \subseteq C$, thì D là một khoá của r. Bởi vậy, M_r là tập phần tử rời nhau cực đại của $I(F_R)$. Vì vậy chúng ta có $A \in M_r$

Ngược lại, nếu $A \in M_r$, thì theo định nghĩa của quan hệ và M_r Chúng ta có $A \rightarrow R$. Có nghĩa là $\forall K \in K_r : K \subseteq A$. Mặt khác, bởi vì A là một phần tử của tập bằng nhau cực đại, cho nên đối với tất cả D($A \subset D$) chúng ta có $D \rightarrow R$. Đồng thời theo định nghĩa của các phản khoá $A \in K^{-1}_r$. □

Cho trước s = <R,F> là một sơ đồ quan hệ trên R, K_s là tập tất cả các khoá tối tiểu của s. Kí pháp K_s^{-1} là tập các phản khoá của s. Từ Định lí 17 chúng ta có kết quả sau.

Hệ quả 18.

Cho trước s = <R,F> là một sơ đồ quan hệ và r là một quan hệ trên R. Khi đó $K_r = K_s$ nếu và chỉ nếu $K_s^{-1} = M_r$, ở đây M_r là hệ bằng nhau cực đại của r.

Chúng ta đưa ra một kết quả liên quan đến cả K^{-1} và K.

Định lý 19.

Giả sử K là một hệ Sperner trên R. Giả sử s(K) = min{m:K=K_r, |r|=m, r là quan hệ trên R}. Khi đó

$$\sqrt{2^{|K^{-1}|}}\ 2 \le s(K) \le |K^{-1}|+1.$$

Đánh giá này chỉ ra mối quan hệ giữa kích cỡ của quan hệ tối tiểu mà thể hiện một hệ Sperner (ở đây hệ này đóng vai trò là một hệ khoá tối tiểu) cho trước với lực lượng của hệ phản khoá tương ứng của nó.

Cho F là một họ f trên R. Theo Định nghĩa 16 thì dễ thấy I(F) là một nửa dàn giao. Khi đó

$$N_{I(F)} = \{A \in I(F) : A \neq \cap\{A' \in I : A \subset A'\}\}.$$

Trên cơ sở này chúng ta có định lí sau đánh giá quan hệ Armstrong nhỏ nhất (minimal Armstrong relation) của một họ f.

Định lý 20.

Giả sử F là một họ f trên R. Đặt

s(F) = min{m:F=F_r, |r|=m, r là quan hệ trên R}. Khi đó $\sqrt{2^{|N_{I(F)}|}} \le s(F) \le |N_{I(F)}|+1$.

Đánh giá này cho chúng ta mối quan hệ giữa kích thước của quan hệ Armstrong nhỏ nhất của họ F với lực lượng của hệ sinh nhỏ nhất của I(F).

Bây giờ, chúng ta đánh giá sâu hơn kích thước của các quan hệ Armstrong nhỏ nhất trên R, cũng như các quan hệ nhỏ nhất mà thể hiện một hệ Sperner K cho trước.

Chúng ta đặt

P(n) = max { s(K) : K là hệ Sperner tùy ý trên R = $\{a_1,..., a_n\}$ }

và Q(n) = max { s(F) : F là họ f tùy ý trên R = $\{a_1,..., a_n\}$ }

Khi đó chúng ta có

Định lý 21.

$- 1/n^2 . C_{[n/2]}^n \leq P(n) \leq C_{[n/2]}^n + 1$

$- 1/n^2 . C_{[n/2]}^n \leq Q(n) \leq (1 + C (1/n^{1/2})).C_{[n/2]}^n$

Đánh giá này cho toàn bộ các hệ Sperner (ở đây hệ này đóng vai trò là một hệ khoá tối tiểu) và các họ f có thể có trên R.

Định nghĩa 22.

Giả sử r là một quan hệ trên R và K_r là tập của tất cả các khoá tối tiểu của r. Chúng ta nói rằng a là

một thuộc tính cơ bản của r nếu tồn tại một khoá tối tiểu K ($K \in K_r$) để a là một phần tử của K.

Nếu a không thoả mãn tính chất trên thì a là thuộc tính thứ cấp.

Trong chương 3 chúng ta có thể thấy các thuộc tính cơ bản và thứ cấp đóng một vai trò quan trọng trong việc chuẩn hoá các sơ đồ quan hệ và các quan hệ.

Trong [24] đã chứng minh kết quả sau

Định lí 23.

Cho trước một sơ đồ quan hệ s = <R,F> và một thuộc tính a. Bài toán xác định a là thuộc tính cơ bản hay không là bài toán NP- đầy đủ.

Có nghĩa rằng cho đến nay không có một thuật toán có độ phức tạp thời gian đa thức để giải quyết bài toán này.

Tuy vậy, chúng ta chỉ ra rằng đối với quan hệ thì bài toán này được giải bằng một thuật toán thời gian đa thức.

Trước tiên chúng ta chứng minh kết quả sau.

Định lý 24.

Giả sử K là một hệ Sperner trên R. thì

$$\cup K = R - \cap K^{-1}.$$

Lời giải:

Nếu $c \in \cup K$, thì tồn tại một khoá tối tiểu K sao cho $c \in K$. Đặt $H = K - c$. Rõ ràng H không chứa một khoá nào. Như vậy, tồn tại một phản khoá B để B chứa H. Có thể thấy c không là phần tử của B, vì ngược lại chúng ta có B chứa K. Điều này là vô lí. Vì thế chúng ta có

$$c \in R - B \subseteq R - \cap K^{-1}.$$

Bây giờ chúng ta giả thiết $c \notin \cup K$ và $B \in K^{-1}$. Có thể thấy $c \in B$. Vì ngược lại $c \notin B$, thì $\{c\} \cup B$ hình thành một khoá chứa khoá tối tiểu K ($K \in K$). Như vậy $K \subseteq B$, và chúng ta có $c \in K$. Điều này là vô lý. □

Trên cơ sở của Định lý 17 và Định lí 24 chúng ta chỉ ra rằng đối với một quan hệ, thì vấn đề về thuộc tính cơ bản có thể là giải quyết bằng một thuật toán thời gian đa thức.

Đầu tiên chúng ta xây dựng một thuật toán xác định tập các thuộc tính cơ bản của quan hệ cho trước.

Thuật toán 25.

Vào: $r = \{h_1, ..., h_m\}$ là một quan hệ trên R

Ra: V là tập tất cả thuộc tính cơ bản của r

Bước 1: Từ r chúng ta xây dựng một tập $E_r = \{E_{ij} : m \geq j > i \geq 1\}$ và $E_{ij} = \{ a \in R : h_j(a) = h_j(a) \}$

Bước 2 : Từ E_r chúng ta xây dựng tập

$M = \{B \in P(R) : \text{Tồn tại } E_{ij} \in E_r : E_{ij} = B\}$

Bước 3: Từ M xây dựng tập $M_r = \{ B \in M : \text{Với mọi } B' \in M : B \not\subset B'\}$

Có thể thấy rằng M_r tính được bằng một thuật toán thời gian đa thức.

Bước 4: Xây dựng tập $V = R - \cap M_r$.

Rõ ràng $m.(m+1)/2 \geq |E_r| \geq |M| \geq |M_r|$. Bởi vậy thời gian tính của Thuật toán 25 là một đa thức theo số hàng và số cột của r.

Từ các Định lý 17, 24 và Thuật toán 25 chúng ta có hệ quả sau.

Hệ quả 26.

Tồn tại thuật toán đối với một quan hệ r cho trước, xác định một thuộc tính bất kì là cơ bản hay không với thời gian tính đa thức theo số hàng và cột của r.

Một vấn đề thường xuyên hay xảy ra là đối với một sơ đồ quan hệ cho trước s = <R,F>, và một phụ thuộc hàm $A \rightarrow B$, chúng ta muốn biết $A \rightarrow B$ có là phần tử của F^+ hay không. Để trả lời câu hỏi này

chúng ta cần tính bao đóng F^+ của tập các phụ thuộc hàm F. Tuy nhiên tính F^+ trong trường hợp tổng quát là rất khó khăn và tồn kém thời gian vì tập các phụ thuộc hàm thuộc F^+ là rất lớn cho dù F có thể là nhỏ. Chẳng hạn $F = \{A \rightarrow B_1, A \rightarrow B_2, .. A \rightarrow B_n\}$, F^+ khi đó còn bao gồm cả những phụ thuộc hàm $A \rightarrow Y$ với $Y \subseteq \{B_1 \cup B_2 \cup ... \cup B_n\}$. Như vậy sẽ có 2^n tập con Y . Trong khi đó, việc tính bao đóng của tập thuộc tính A lại không khó. Theo kết quả đã trình bày ở trên việc kiểm tra $A \rightarrow B \in F^+$ không khó hơn việc tính A^+.

Ta có thể tính bao đóng A^+ qua thuật toán sau:

Thuật toán 27

Vào: s = <R,F>, ở đây R ={ a_1 ,...., a_n } tập hữu hạn các thuộc tính, F tập các phụ thuộc hàm, $A \subseteq R$

Ra: A^+ bao đóng của A đối với F

Thuật toán thực hiện như sau: Tính các tập thuộc tính A_0, A_1... theo qui tắc:

1) $A_0 = A$

2) $A_i = A_{i-1} \cup \{a\}$ sao cho $\exists (C \rightarrow D) \in F, \{a\} \in Y$ và $C \subseteq A_{i-1}$

Vì $A = A_0 \subseteq ... \subseteq A_i \subseteq R$, và R hữu hạn nên tồn tại một chỉ số i nào đó mà $A_i = A_{i+1}$, khi đó thuật toán dừng và $A^+ = A_i$

Chúng ta có thể thấy độ phức tạp thời gian của thuật toán này là đa thức theo kích thước của s.

Để tiện kí pháp chúng ta thay a $\cup$ b bởi viết ab.

Ví dụ: s = <R,F>, ở đâyR = { a,b,c,d,e,g }, F bao gồm 8 phụ thuộc hàm

$$ab \rightarrow c \qquad d \rightarrow eg$$

$$c \rightarrow a \qquad be \rightarrow c$$

$$bc \rightarrow d \qquad cg \rightarrow bd$$

$$acd \rightarrow b \qquad ce \rightarrow ag$$

và A = bd.

Dùng Thuật toán 27 chúng ta có thể thấy

A_0 = bd,

A_1 = bdeg,

A_2 = bcdeg,

A_3 = abcdeg,

A^+ = abcdeg.

Để chứng minh tính đúng đắn của Thuật toán 27 chúng ta có thể dùng phương pháp quy nạp để chỉ ra rằng nếu a thuộc A_i thì a cũng thuộc A^+.

Việc chỉ ra điều ngược lại cũng bằng qui nạp nhưng khó khăn hơn là nếu a nằm trong A^+ thì a nằm trong một số A_i nào đó.

Định lý 28.

Thuật toán 27 tính chính xác A^+.

Như vậy, để xác định một phụ thuộc hàm $A \rightarrow B$ có thuộc F^+ hay không chúng ta chỉ cần kiểm tra $B \subseteq A^+$?.

Bây giờ, chúng ta đi tìm thuật toán tìm bao đóng cho một tập các thuộc tính trên một quan hệ bất kì

Đối với một quan hệ bất kì theo Định lí 13 chúng ta đã chúng minh với mọi $A \subseteq R$

$$L_{Fr}(A) = \begin{cases} \cap E_{ij} \text{ nếu tồn tại } E_{ij} \in E_r : A \subseteq E_{ij}, \\ A \subseteq E_{ij} \\ R \quad \text{ngược lại.} \end{cases}$$

Có thể thấy $L_{Fr}(A) = A_r^+$. Do vậy chúng ta có ngay thuật toán sau để tính bao đóng cho một tập bất kì trên quan hệ r.

Thuật toán 29

Vào: $r = \{h_1, ..., h_m\}$ là một quan hệ trên R

Ra: V là tập tất cả thuộc tính cơ bản của r

Bước 1: Từ r chúng ta xây dựng một tập $E_r = \{E_{ij} : m \geq j > i \geq 1\}$ và $E_{ij} = \{ a \in R : h_j(a) = h_j(a) \}$

Bước 2: Từ E_r chúng ta xây dựng một tập $M = \{B \in P(R) : \text{Tồn tại } E_{ij} \in E_r : E_{ij} = B\}$

Bước 3:

$$A_r^+ = \begin{cases} \cap B \text{ nếu tồn tại } B \in M : A \subseteq B, \\ R \text{ ngược lại.} \end{cases}$$

Dễ thấy, độ phức tạp thời gian của thuật toán này là một đa thức theo kích thước của r.

Định nghĩa 30

Giả sử s = <R,F>, t = <R,G > là hai sơ đồ quan hệ trên R. Khi đó chúng ta nói s tương đương với t nếu $F^+ = G^+$. Nếu s và t tương đương thì đôi khi chúng ta có thể nói rằng s là một phủ của t hoặc t là một phủ của s.

Dễ dàng kiểm tra rằng s và t có tương đương với nhau không.

Mỗi phụ thuộc hàm $Y \rightarrow Z$ ở trong F chúng ta kiểm tra lại $Y \rightarrow Z$ ở trong G^+ bằng vi ệc sử dụng Thuật toán 27 để tính Y^+ và kiểm tra $Z \subseteq Y^+$. Nếu có phụ thuộc hàm $Y \rightarrow Z$ không nằm trong G^+ hoặc ngược lại nếu có phụ thuộc hàm $X \rightarrow W$ ở trong G

nhưng không ở trong F^+ thì điều chắc chắn là F^+ khác G^+. Nếu mỗi phụ thuộc hàm ở trong F thì cũng nằm trong G^+ và mỗi phụ thuộc hàm nằm trong G thì cúng nằm trong F^+, khi đó chúng ta có s và t là tương đương với nhau.

Hiện nay chúng ta cho định nghĩa sau nói về sự tương đương của hai quan hệ.

Định nghĩa 31

Giả sử r và v là hai quan hệ trên R. Khi đó ta nói r và v tương đương với nhau nếu $F_r = F_v$.

Chúng tôi trình bày định lí sau liên quan đến sự tương đương của hai quan hệ.

Định lí 32

Giả sử r và v là hai quan hệ trên R. Khi đó s tương đương với v khi và chỉ khi $N_r = N_v$.

Trên cơ sở Định lí 32 chúng ta đưa ra một thuật toán kiểm tra xem r có tương đương với v hay không.

Thuật toán 33

Vào: r và t là hai quan hệ trên R

Ra: r có tương đương với v hay không

Bước 1: Từ r tính N_r,

Bước 2: Từ v tính N_v

Bước 3 : So sánh M_r với M_v.

Bây giờ, chúng ta quay lại với sơ đồ quan hệ. Chúng ta muốn gọt giũa các phụ thuộc hàm của sơ đồ quan hệ để có tập phụ thuộc hàm tốt hơn.

Định nghĩa 34

Chúng ta nói một sơ đồ quan hệ s = <R,F> là chính tắc nếu

1. Vế phải của mỗi phụ thuộc hàm trong F là thuộc tính đơn.

2. Không có $X \to a$ nào ở trong F để F- $\{X \to a\}$ tương đương với F.

3. Không có $X \to a$ và một tập con Z của X để $F-\{X \to A\} \cup \{Z \to a\}$ tương đương với F.

Chúng ta sẽ chỉ ra rằng có thuật toán để tìm một phủ chính tắc cho một sơ đồ quan hệ bất kì.

Trước tiên chúng ta đưa ra mệnh đề sau

Mệnh đề 35:

Mỗi một sơ đồ quan hệ s = <R,F> đều có một phủ tương đương t = <R,G> sao cho vế phải của mỗi phụ thuộc hàm trong G không có hơn một thuộc tính.

Chứng minh: Đặt G là tập phụ thuộc hàm có dạng $X \to a$, với $X \to Y$ nằm trong F và a là một

phần tử của Y. Trên cơ sở hệ tiên đề của Armstrong, chúng ta dễ thấy t tương đương với s. □

Chúng ta trình bày thuật toán dưới đây để tìm phủ chính tắc cho một sơ đồ quan hệ cho trước

Thuật toán 36

Vào: s= <R,F>, F = { $A_1 \rightarrow B_1$,..., $A_m \rightarrow B_m$ }

Ra: t = <R,G> là chính tắc và tương đương với s

Do Mệnh đề 35 chúng ta có thể coi s thoả mãn điều kiện 1.

Bước 1: Đặt F_0 = F, với i = 1, ..., m

$F_i = F_{i-1}$ - { $A_i \rightarrow B_i$ } nếu F_{i-1} - { $A_i \rightarrow B_i$ } tương đương với F_i . Trong trương hợp ngược lại thì $F_i = F_{i-1}$.

Bước 2: Nhờ thuật toán tính bao đóng, từ F_m chúng ta lần lượt loại bỏ các thuộc tính thừa trong mỗi vế trái của từng phụ thuộc hàm thuộc F_m.

Kết quả nhận được chính là G.

Dễ thấy rằng thuật toán trên có độ phức tạp thời gian là đa thức theo kích thước của s.

Chúng ta đưa ra một khái niệm sau

Định nghĩa 37

Giả sử s = <R,F> là một sơ đồ quan hệ trên R. Khi đó chúng ta nói rằng s là sơ đồ quan hệ tối tiểu nếu với mọi sơ đồ quan hệ t = < R,G> tương đương với s có | F | < |G|, ở đây |F| là số lượng các phụ thuộc hàm trong F.

David Maier [25] đã chứng minh định lí sau

Định lí 38

Tồn tại một thuật toán có độ phức tạp thời gian đa thức để tìm phủ tối tiểu cho một sơ đồ quan hệ cho trước.

Ví dụ:

Chúng ta xem xét tập các phụ thuộc hàm F trong ví dụ minh hoạ Thuật toán 27. Ban đầu chúng ta tách vế phải ra thành các thuộc tính đơn:

$ab \rightarrow c,$

$c \rightarrow a,$

$bc \rightarrow d,$

$acd \rightarrow b,$

$d \rightarrow e,$

$d \rightarrow g,$

$be \rightarrow c,$

$cg \rightarrow b,$

cg→d,

ce→a,

ce →g.

Rõ ràng ce →a là không cần thiết bởi vì nó suy ra được từ c →a.

cg →b là dư thừa bởi vì có cg →d , c →a và acd →b. Ngoài ra không có một phụ thuộc hàm nào là dư thừa nữa. Dễ thấy acd →b có thể thay thế bởi cd →b. Vậy chúng ta có một phủ chính tắc là :

ab →c,

c →a,

bc→d,

cd→b,

d→e,

d→g,

be→c,

cg→d,

ce→g.

2.3 Các mô tả tương đương của họ các phụ thuộc hàm

Trong mục trên chúng ta đã định nghĩa hàm đóng. Nó là một mô tả tương đương của họ các phụ thuộc hàm. Trong mục này chúng tôi cung cấp cho bạn đọc một số các mô tả tương đương khác của họ này. Chúng chính là các công cụ để chúng ta có thể nghiên cứu phong phú hơn nữa cấu trúc lôgic của họ các phụ thuộc hàm.

Định nghĩa 1

Cho R là tập các thuộc tính và P(R) là tập các tập con của R.

Một hàm C: P(R) $\rightarrow$ P(R) được gọi là một hàm chọn trên R nếu với mọi A $\in$ P(R) thì C (A) $\subseteq$ A.

Giả sử L là một hàm đóng trên R . Chúng ta đặt C(A) = R - L(R-A) (*)

Dễ thấy C là một hàm chọn trên R

Trong [6] người ta đã chứng minh được kết quả sau

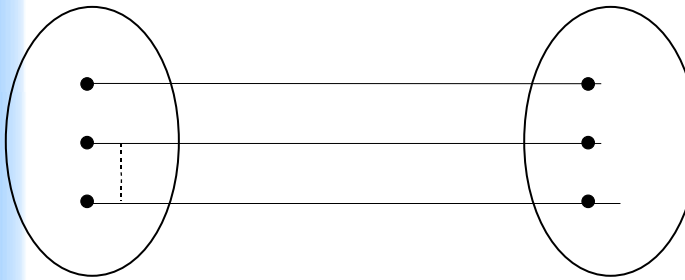
Định lí 2

Tương ứng được xác định như (*) là tương ứng 1-1 giữa tập các hàm đóng và tập các hàm chọn thoả mãn 2 điều kiện sau: với mọi A,B $\subseteq$ R

(1) C(A) $\subseteq$ B $\subseteq$ A kéo theo C(A) = C(B)

(2) A $\subseteq$ B kéo theo C(A) $\subseteq$ C(B).

Chúng ta có hình vẽ sau thể hiện mối quan hệ giữa lớp hàm đóng và lớp hàm chọn đặc biệt trên



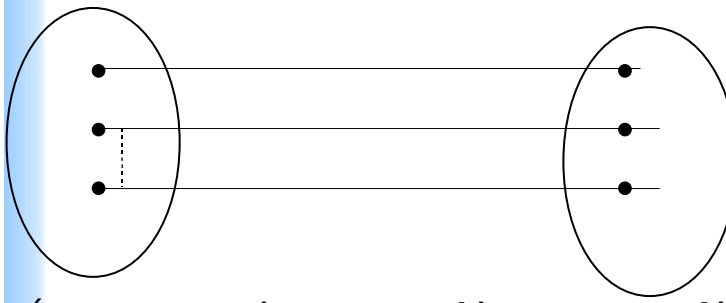
Lớp các hàm đóng Lớp các hàm chọn đặc biệt

Định lí dưới đây chỉ ra một tương ứng 1-1 giữa lớp các hàm đóng và lớp các nửa dàn giao trên R.

Định lí 4

Giả sử L là một hàm đóng trên R. Đặt Z(L) = { A : A $\in$ P(R) và L(A) = A }. Khi đó Z(L) là một nửa dàn giao trên R.

Ngược lại, nếu I là một nửa dàn giao trên R, thì tồn tại duy nhất một hàm đóng L sao cho Z(L) = I, ở đây L(A) = $\cap$ {A' $\in$ I : A $\subseteq$ A'}.



Lớp các hàm đóng Lớp các nửa dàn giao

Như vậy, từ Định lí 6 mục trên và Định lí 4, chúng ta thấy có một tương ứng 1-1 giữa lớp các nửa dàn giao và lớp các họ các phụ thuộc hàm trên R .

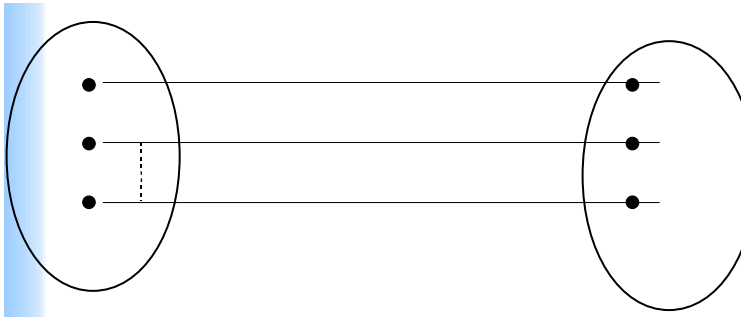
Định nghĩa 5

Giả sử $N \subseteq P(R)$. Khi đó N được gọi là tập không giao nếu với mọi $A \in N$ thì $A \neq \cap \{A' \in N : A \subset A'\}$.

Định lí 6

Nếu I là nửa dàn giao, thì N_I là tập không giao. Ngược lại nếu N là tập không giao thì tồn tại duy nhất một nửa dàn giao I sao cho $N_I = N$.

Như vậy, chúng ta thấy có một tương ứng 1-1 giữa lớp các nửa dàn giao và lớp các tập không giao.



Lớp các nửa dàn giao Lớp các tập không giao

Từ Định lí 6 mục trên và các Định lí 4 và 6 chúng ta rút ra kết luận là có một tương ứng 1-1 giữa lớp các họ phụ thuộc hàm với lớp các tập không giao.

Như vậy, để nghiên cứu phân tích các đặc trưng của họ các phụ thuộc hàm chúng ta có thể dùng công cụ nửa dàn giao hoặc tập không giao.

Bây giờ chúng tôi đưa ra khái niệm họ cực đại các thuộc tính. Đồng thời chúng ta chỉ ra rằng họ này là một mô tả tương đương của họ phụ thuộc hàm.

Định nghĩa 7

Giả sử R là tập các thuộc tính. Họ $M=\{(A,\{a\}): A \subset R, a \in R\}$ được gọi là họ cực đại các thuộc tính trên R nếu nó thoả mãn các điều kiện sau

(1) $a \notin A$,

(2) Đối với mọi (B, {b}) ∈ M, a ∉ B và A ⊆ B kéo theo A = B.

(3) ∃(B,{b}) ∈ M : a ∉ B, a ≠ b, và $L_a \cup B$ là một hệ Sperner trên R, ở đây L_a = {A: (A, {a}) ∈ M}.

Nhận xét 8.

- Có thể có (A, {a}), (B, {b}) ∈ M mà a ≠ b, nhưng A = B.

- Do (1) và (2) có thể thấy đối với a ∈ R chúng ta có L_a là một hệ Sperner trên R. Đặc biệt có thể L_a là một hệ Sperner rỗng.

- Trên cơ sở Định nghĩa 7 chúng ta có thể thấy tồn tại một thuật toán thời gian tính đa thức để xác định một tập Y ⊆ P(R) x P(R) có là một họ cực đại các thuộc tính trên R hay không.

Giả sử H là một hàm đóng trên R. Đặt Z(H) = {A : H(A) = A} và M(H) = {(A, {A}) : A ∉ A, A ∈ Z(H) và B ∈ Z(H), A ⊆ B, A ∉ B kéo theo A = B}.

Z(H) là họ các tập đóng của H. Dễ thấy, với mỗi (A, {a}) ∈ M(H), A là tập đóng cực đại mà không chứa a.

Có thể tồn tại (A, {a}), (B, {b}) ∈ M(H) mà a ≠ b, nhưng A = B.

Nhận xét 9.

Giả sử r là một quan hệ trên R và F_r là họ các phụ thuộc hàm của r. Đặt $A_r^+ = \{a: A \rightarrow \{a\} \in F_r\}$ và $Z_r = \{A: A = A_r^+\}$ và N_r là hệ sinh cực tiểu của nó. Có thể thấy $N_r \subseteq E_r$ với

$N_r = \{A \in E_r : A \neq \cap\{B: B \in E_r, A \subset B\}\}$, ở đây E_r là hệ bằng nhau r.

Chúng ta cho định lí dưới đây chỉ ra rằng giữa các hàm đóng và họ cực đạI các thuộc tính có tương ứng 1-1.

Định lí 10.

Giả sử H là một hàm đóng trên R. Khi đó M(H) là họ cực đại các thuộc tính. Ngược lại, nếu M là họ cực đạI các thuộc tính trên R thì tồn tại đúng một hàm đóng H trên R để M(H) = M, ở đây với mọi $B \in P(R)$.

$$H(B) = \begin{cases} \cap A \text{ if } \exists A \in L(M) : B \subseteq A, \\ R \quad \text{ngược lại,} \end{cases}$$

và $L(M) = \{A: (A, \{a\} \in M\}$.

Lời giải:

Giả sử H là hàm đóng trên R. Cơ sở trên định nghĩa của M(H) ta có (1) và (2). Ta đặt $L'_a = \{ A : (A,$

{a} ∈ M(H)}. Giả thiết có S (B, {b}) ∈ M(H) : a ≠ b, a ∉ B, $L'_a \cup B$ là hệ Sperner trên R (*). Khi đó ta chọn (B,{b}) ∈ M(H) sao cho B là lớn nhất cho (*). Do (2) trong Định nghĩa 7 ta có L_a' là hệ Sperner trên R. Do đó, không có một A nào mà A ∈ L_a' và B ⊆ A. Phù hợp với định nghĩa của M(H) ta có (B,{a}) ∈ M(H). Như vậy, B ∈ L_a'. Điều này là vô lí. Từ đó ta có (3) trong Định nghĩa 7. Có nghĩa là M(H) là họ cực đại các thuộc tính trên R.

Ngược lại, giả sử M là họ cực đại các thuộc tính trên R. Đặt L(M) = {A : (A, {a}) ∈ M}. Đầu tiên ta chứng tỏ rằng L(M) là tập không giao trên R. Đối với bất kì (A, {a}) ∈ M do Nhận xét 2.2 ta có A ≠ A' ∩ A" và A ≠ A' ∩ B, ở đây A', A" ∈ L_a và B ∈ L(M) : A ≠ B.

Nếu tồn tại (B, {b}), (C, {c}) ∈ M sao cho b ≠ a, c ≠ a, A ⊂ B, A ⊂ C thì bởi (2) trong Định nghĩa 2.1 ta có a ∈ B, a ∈ C. Từ đó, A ⊂ B ∩ C. Như vậy, đối với A,B,C, ∈ L(M) nếu A = B ∩ C thì A = B hoặc A = C. Do đó, L(M) là hệ không giao trên R. Chúng ta biết rằng các họ không giao và các hàm đóng xác định duy nhất lẫn nhau. Mặt khác phù hợp với Nhận xét 8 và Định lí 13 mục trên ta có H là một hàm đóng trên R và L(M) là hệ sinh tối tiểu của Z(H).

Hiện ta sẽ chứng minh M(H) = M. Nếu (A, {a}) $\in$ M thì A $\in$ L(M). Giả thiết rằng đối với mỗi b $\notin$ A có B $\in$ Z(H) : A $\subset$ B, b $\notin$ B. Có thể thấy rằng A là giao của những B như thế. Điều này mâu thuẫn với A $\in$ L(M). Nếu (A, {a}) $\in$ M thì có b $\notin$ A để (A, {b}) $\in$ M(H) (**).

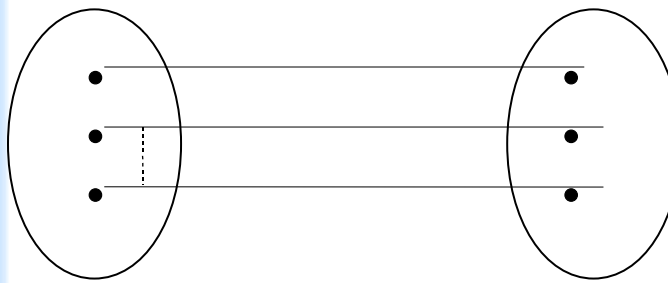
Nếu (A, {a}) $\in$ M(H) thì phù hợp với định nghĩa của M(H) B $\in$ Z(H), và A $\subset$ B kéo theo a $\in$ B. Vì a $\notin$ A, ta thấy A không là giao của các B như thế. Theo cấu trúc của H ta có A $\in$ L(M). Như vậy, nếu (A, {a}) $\in$ M(H) thì A $\in$ L(M) (***).

Bây giờ ta giả thiết là (A,{a}) $\in$ M, nhưng (A, {a}) $\notin$ M(H). Vì A là tập đóng của H,a $\notin$ A và bởi định nghĩa của M(H) thì tồn tại (B,{a}) $\in$ M(H) sao chot A $\subset$ B. Do (***) ta có B $\in$ L(M). Điều này mâu thuẫn với điều kiện (2) của Định nghĩa 7. Từ đó, ta có (A,{a}) $\in$ M(H).

Giả thiết (A,{a}) $\in$ M(H), nhưng (A,{a}) $\notin$ M. Nếu có A' $\in$ L_a để A $\subset$ A' thì bởi (**) ta có A' là một tạp đóng của H. Phù hợp với định nghĩa của M(H) ta có (A,{a}) $\notin$ M. Điều này là vô lí. Nếu A' $\subset$ A thì do (***) ta có (A',{a}) $\notin$ M, nó cũng vô lí. Nếu A $\cup$ L_a là hệ Sperner trên R thì bởi (***) ta có thể thấy nó trái với điều kiện (3) của Định nghĩa 7. Do đó tồn tại

một A' mà $A' \in L_a$ và $A = A'$. Từ đó ta có M(H) = M.

Nếu ta giả thiết rằng có H' mà M(H') = M. Đặt L(H') = {A : (A,{a}) ∈ M(H'). Theo (**) và (***) của chứng minh trên ta có L(H') là một hệ sinh nhỏ nhất của Z(H'). Do M(H') = M(H) = M ta có L(H') = L(M) = L(H). Vì các hàm đóng và các tập không giao là xã định duy nhất lẫn nhau nên H = H'. □



Lớp các hàm đóng cực đại các thuộc tính

Lớp các họ

Vì các hàm đóng và các họ các phụ thuộc hàm tương ứng xác định duy nhất lẫn mhau, từ Định lí 2.4 ta có

Hệ quả 11.

Tồn tại tương ứng 1-1 giữa lớp các họ cực đại các thuộc tính và họ các phụ thuộc hàm.

2.4. Các thuật toán liên quan đến các khoá

Khi giải quyết các bài toán thông tin quản lí, chúng ta thường sử dụng các hệ quản trị cơ sở dữ liệu mà trong đó chứa cơ sở dữ liệu quan hệ. Các phép xử lí đối với lớp bài toán này thường là tìm kiếm bản ghi sau đó thay đổi nội dung bản ghi, thêm bản ghi mới hoặc xoá bản ghi cũ. Trong các thao tác trên việc tìm kiếm bản ghi là rất quan trọng. muốn tìm được bản ghi trong một file dữ liệu chúng ta phải xây dựng khoá cho file dữ liệu đó.

Việc xây dựng khoá ở đây chính là xây dựng khoá tối tiểu. Vì thế trong mục này chúng tôi cung cấp cho bạn đọc hai thuật toán tìm khoá tối tiểu.

2.4.1 Thuật toán tìm khoá tối tiểu của một sơ đồ quan hệ

Vào : sơ đồ quan hệ s = < F, R> trong đó :

F là tập các phụ thuộc hàm và

R={ a_1,...., a_n } là tập các thuộc tính

Ra : K là một khoá tối tiểu

Thuật toán thực hiện như sau:

Tính liên tiếp các tập thuộc tính K_0, K_1....., K_n như sau:

$K_o = R = \{ a_1, ..., a_n \}$

$K_I = \begin{cases} K_{i-1} \text{ nếu } K_{i-1} - \{a_i\} \rightarrow R \not\in F^+, \\ K_{i-1} - \{a_i\} \quad \text{ngược lại} \end{cases}$

...

$K = K_n$ là khoá tối tiểu

2.4.2. Thuật toán tìm một khoá tối tiểu của một quan hệ

Cho trước $r = \{h_1,.... h_m\}$ là một quan hệ trên tập thuộc tính $R = \{ a_1 ,...., a_n \}$

Hệ bằng nhau của quan hệ r được định nghĩa ở phần trên như sau:

$E_r = \{E_{ij} : 1 \leq i \leq j \leq m\}$, ở đây $E_{i\,j} = \{a \in R : h_i (a) = h_j (a)\}$.

Dễ dàng thấy rằng E_r được tính bằng thời gian đa thức từ r

Thuật toán tìm một khoá tối tiểu của một quan hệ r:

Vào: $r = \{h_1,.... h_m\}$ là một quan hệ trên tập thuộc tính $R= \{ a_1 ,...., a_n \}$

Ra: K là một khoá tối tiểu của r

Thuật toán thực hiện như sau

Bước 1: Tính $E_r = \{ A_1 ....A_t\}$

Bước 2 : Tính $M_r = \{ A :$ có $A_j \in E_r : A = A_j$ và không có $A_i : A_i \in E_r : A \subset A_i \}$

Bước 3 : Lần lượt tính các tập thuộc K_1, K_2, ...K_n tính theo qui tắc :

$K_0 = R = \{ a_1, ..., a_n \}$ hoặc K_0 = một khoá đã biết

$K_i = K_{i-1} - \{ a_i \}$ nếu không tồn tại $A \in M_r$ sao cho $K_{i-1} - \{a_i\} \subseteq A$ hoặc $K_i = K_{i-1}$ trong trường hợp ngược lại

Bước 4: Đặt $K = K_n$, khi đó K là khoá tối tiểu

2.5. Mối quan hệ giữa quan hệ Armstrong và sơ đồ quan hệ

Việc xây dựng quan hệ Armstrong của một sơ đồ quan hệ cho trước và ngược lại từ quan hệ cho trước ta xây dựng một SĐQH sao cho quan hệ cho trước này là quan hệ Armstrong của nó có vai trò rất quan trọng trong việc phân tích cấu trúc lôgic của mô hình dữ liệu quan hệ cả trong thiết kế lẫn trong ứng dụng. Đã có nhiều tác giả nghiên cứu vấn đề này. Trong mục này chúng tôi trình bày hai thuật toán giải quyết bài toán trên và đưa ra việc đánh giá các thuật toán này cũng như đánh giá độ phức tạp của bài toán trên.

Trước tiên, chúng ta cho một thuật toán tìm tập tất cả các phản khoá của hệ Spernner cho trước.

Thuật toán 1 (Tìm tập phản khoá)

Vào: $K = \{B_1,...,B_n\}$ là hệ Sperner trên R.

Ra: K^{-1}.

Bước 1: Ta đặt $K_1 = \{R - \{a\}: a \in B_1\}$. Hiển nhiên $K_1 = \{B_1\}^{-1}$.

Bước q + 1: (q < m). Ta giả thiết rằng $K_q = F_q \cup \{X_1... X_{tq}\}$, ở đây $X_{1,}....,X_{tq}$ chứa B_{q+1} và $F_q = \{ A \in K_q : B_{q+1} \subseteq A \}$. Đối với mỗi I (I = 1,..., t_q) ta tìm các phản khoá của { B_{q+1} } trên X_i tương tự như K_1. Kí pháp chúng là A^i_1,..., $A_{ri}{}^i$. Đặt

$K_{q+1} = F_q \cup \{A^i_p : A \in F_q$ kéo theo $A^i_p \not\subset A, 1 \leq i \leq t_q. 1 \leq p \leq r_i\}$

Cuối cùng ta đặt $K^{-1} = K_m$

Định lí 2.

Với mọi q ($1 \leq q \leq m$), $K_q = \{ B_{1,}.... B_q\}^{-1}$, có nghĩa là $K_m = K^{-1,,}$

Rõ ràng, K và K^{-1} là xác định duy nhất lẫn nhau và từ định nghĩa của K^{-1} có thể thấy thuật toán của chúng ta không phụ thuộc vào thứ tự của dãy B_1,..., B_m. Đặt $K_q = F_q \cup \{X_1,..., X_{tq}\}$ và l_q ($1 \leq q \leq$ m-1) là số các phần tử của K_q. Khi đó ta có

Mệnh đề 3

Độ phức tạp thời gian tồi nhất của Thuật toán **2.1** là

m-1

$$O\left(|R|^2 \sum_{q=1} t_q.u_q\right).$$

ở đây $l_q - t_q$, nếu $l_q > t_q$,

$u_q =$ —

1 nếu $l_q = t_q$

Rõ ràng trong mỗi bước thuật toán ta có K_q là hệ Sperner trên R. Ta biết rằng[5] kích thước của hệ Sperner bất kì trên R không vượt quá $C_n^{[n/2]}$, ở đây $n = |R|$. Có thể thấy $C_n^{[n/2]}$ xấp xỉ bằng $2^{n+1/2}/(\Pi . n^{1/2})$. Từ đó độ phức tạp thời gian tồi nhất của thuật toán trên không nhiều hơn hàm số mũ theo n. Trong trường hợp mà $l_q \leq l_m$ (q = 1,..., m-1), dễ thấy rằng độ phức tạp thuật toán không lớn hơn $O(|R|^2 |K| |K^{-1}|^2)$. Như vậy, trong các trường hợp này độ phức tạp của Thuật toán 1 tìm K^{-1} là đa thức theo $|R|, |K|$, and $|K^{-1}|$. Có thể thấy nếu số lượng các phần tử của K là nhỏ thì Thuật toán 1 là rất hiệu quả. Nó chỉ đòi hỏi thời gian đa thức theo |R|

Định nghĩa 4

Cho s = (R,F) là SĐQH trên R và $a \in R$. Đặt

$K_a = \{ A \subseteq R: A \rightarrow \{a\}, \nexists B: (B \rightarrow \{a\})(B \subset A)\}$. K_a được gọi là họ các tập tối tiểu của thuộc tính a.

Rõ ràng, $R \notin K_a$, $\{a\} \in K_a$ và K_a là hệ Sperner trên R.

Thuật toán 5. (Tìm tập tối tiểu của thuộc tính a)

Vào: Cho s = (R = {a_1,..., a_n}, F) là SĐQH, a = a_1.

Ra: A ∈ K_a.

Bước 1: Ta đặt L(0) = R.

Bước i + 1: Đặt

$$L(i+1) = \begin{cases} L(i) - a_{i+1} & \text{nếu } L(i) - a_{i+1} \to \{a\}, \\ L(i), & \text{ngược lại.} \end{cases}$$

Khi đó A = L(n).

Bổ đề 6.

L(n) ∈ K_a

Lời giải: Bằng phương pháp chứng minh qui nạp có thể thấy L(n) → {a}, và L(n) ⊆ ... ⊆ L(0) (1). Nếu L(n) = a, thì bởi định nghĩa của tập tối tiểu của thuộc tính a ta thu được L(n) ∈ K_a. Bây giờ ta giả thiết là tồn tại một B sao cho B ⊂ L(n) và B ≠ 0. Như vậy sẽ có a_j sao cho a_j ⊄ B, a_j ∈ L(n). Theo các xây dựng thuật toán này ta có L(j-1) - a_j → {a}. Rõ ràng bởi (1) ta thu được L(n) - a_j ⊆ L(j-1) - a_j (2). Dễ thấy B ⊆ L(n) - a_j.

Từ (1), (2) ta có B → {a}. □

Dễ thấy, vì thuật toán xác định một phụ thuộc hàm bất kì có phải là phụ thuộc hàm của một SĐQH hay không là có độ phức tạp thời gian đa thức, nên độ phức tạp của Thuật toán 5 là $O(|R|^2|F|)$.

Bổ đề 7.

Cho s = (R,F) là SĐQH trên R và $a \in R$, K_a là họ các tập tối tiểu của a, $L \subseteq K_a$, $\{a\} \in L$. Khi đó $L \subset K_a$ nếu và chỉ nếu tồn tại C, $A \rightarrow B$ sao cho $C \in L$ và $A \rightarrow B \in F$ và $\forall E \in L \Rightarrow E \subseteq A \cup (C - B)$.

Lời giải. =>: Ta giả thiết rằng $L \subset K_a$. Do đó, tồn tại $D \in K_a - L$. Bởi $\{a\} \in L$ và K_a là hệ Sperner trên R, chúng ta có thể xây dựng một tập cực đại Q sao cho $D \subseteq Q \subset U$ và $L \cup Q$ là hệ Spernner. Từ định nghĩa của K_a, chúng ta thu được $Q \rightarrow \{a\}$ (1) và $a \notin Q$ (2). Nếu $A \rightarrow B \in F$ kéo theo (A tb Q, $B \subseteq Q$) hoặc $A \subseteq Q$ thì $Q^+ = Q$. Bởi (2) ta có $Q \rightarrow \{a\}$. Điều này mâu thuẫn với (1). Do đó, tồn tại một phụ thuộc hàm $A \rightarrow B$ sao cho $A \subseteq Q$, $B \subseteq Q$. Từ cách xây dựng của Q có C sao cho $C \in L$, $A \subseteq Q$, $C - B \subseteq Q$. Hiển nhiên rằng $A \cup (C-B) \subseteq Q$.

Rõ ràng, $E \subseteq A \cup (C-B)$ đối với mọi $E \in L$.

<=: Ta giả thiết rằng có C, và $A \rightarrow B$ sao cho $C \in L$, $A \rightarrow B \in F$ và $E \subseteq A \cup (C-B)$ đối với mọi $E \in L$ (3). Bởi định nghĩa của L chúng ta thu được $A \cup$

U(C-B) $\rightarrow$ {a}. Bởi {a} thuộc L nên có D sao cho D $\in$ K_a, a $\notin$ D, D thuộc A $\cup$ (C-B). Bởi (3) ta có D $\in$ K_a - L. □

Cơ sở trên bổ đề này và thuật toán 5, chúng ta xây dựng một thuật toán sau đây bằng qui nạp.

Thuật toán 8. Tìm họ các tập cực tiểu của thuộc tính a.

Vào: Cho s = (R, F) là một sơ đồ quan hệ và a thuộc R.

Ra: K_a

Bước 1: Đặt L(1) = E_1 = {a}

Bước i + 1: Nếu có C và A $\rightarrow$ B mà C $\in$ L(i), A $\rightarrow$ B $\in$ F, $\forall$ E $\in$ L(i) $\rightarrow$ E $\notin$ A $\cup$ (C - B), thì bởi thuật toán 5 chúng ta xây dựng E_{i+1}, ở đây $E_{i+1} \subseteq$ A $\cup$ (C - B), $E_{i+1} \in K_a$. Chúng ta đặt K(i+1) = K(i) $\cup$ E_{i+1} . Trong trường hợp ngược lại ta đặt K_a = L(i).

Bởi bổ đề 7 hiển nhiên rằng tồn tại một số tự nhiên t để K_a = L(t)

Có thể thấy rằng độ phức tạp thời gian tồi nhất của thuật toán là O (| U| |F| |K_a| (| U| + ||K_a |)). Như vậy, độ phức tạp thời gian của thuật toán này là đa thức theo| U | , | F | , và| K_a | .

Rõ ràng, nếu số lượng các phần tử của K_a đối với sơ đồ quan hệ s = <R, F> là đa thức theo kích thước của s, thì thuật toán này là rất hiệu quả. Đặc biệt khi $|K_a|$ là nhỏ.

Hiển nhiên rằng nếu đối với mỗi $A \to B \in F$ kéo theo $a \in A$ hoặc $a \notin B$, thì $K_a = \{a\}$

Nhận xét 9

Biết rằng [27] nếu s = <R, F> là một sơ đồ quan hệ, $Z(F) = \{A : A^+ = A\}$ và N(F) là hệ sinh nhỏ nhất của Z(F), thì

$$N(F) = MAX(F^+) = \bigcup_{a \in R} MAX(F^+, a)$$

ở đây $MAX(F^+,a) = \{A \subseteq U : A \to \{a\} \notin F^+, A \subset B \to B \to \{a\} \in F^+$. Rõ ràng rằng, K_a là một hệ Sperner trên R. Có thể thấy MAX (F^+, a) là tập các phản khoá của K_a đối với mọi $a \in R$. Như vậy, $MAX(F^+, a) = K_a^{-1}$.

Định lý 10

Cho $r = \{h_1,...h_m\}$ là một quan hệ, và F là một họ f trên R. Khi đó $F_R = F$ nếu và chỉ nếu với mọi $A \in P(R)$

$$\bigcap E_{ij} \quad \text{nếu } \exists E_{ij} \in E_r ; A \subseteq E_{ij}$$

$L_F(A) = \quad A \subseteq E_{ij}$

R ngược lại,

ở đây $L_F(A) = \{a \in R : (A, \{a\}) \in F\}$ và E_r là hệ bằng nhau của r.

Trên cơ sở nhận xét 9, định lý 10, các thuật toán 1, 8, chúng ta xây dựng một thuật toán tìm quan hệ Armstrong từ một sơ đồ quan hệ cho trước như sau:

Thuật toán 11 (Tìm quan hệ Armstrong)

Vào: Cho s = <R, F> là một sơ đồ quan hệ

Ra: r là quan hệ sao cho $F_r = F^+$

Bước 1: Đối với mỗi $a \in R$ bởi thuật toán 2.8 chúng ta tính K_a, và từ thuật toán 2.1 xây dựng tập các phản khoá K_a^{-1} .

Bước 2: $N = \bigcup_{a \in R} K_a^{-1}$

Bước 3: Giả sử các phần tử của N là $A_1,...,A_t$, chúng ta xây dựng quan hệ $r = \{ h_0, h_1, ..., h_0 \}$ như sau: Với mỗi $a \in R$, $h_0(a) = 0$, $\forall i = 1,...,t$

$h_i(a) = 0$ nếu $a \in A_i$, hoặc $h_i(a) = 1$ trong trường hợp ngược lại.

Do nhận xét 9 rõ ràng rằng, nếu chúng ta có N(F), thì chúng ta có thể trực tiếp xây dựng r. Độ phức tạp của việc xây dựng này phụ thuộc vào

N(F) |. Dễ thấy độ phức tạp của thuật toán 11 là độ phức tạp của bước 1. Bởi bổ đề 3 và đánh giá của thuật toán 8, dễ thấy rằng độ phức tạp tồi nhất của thuật toán 11 là

$$O\left(n \sum_{i=1}^{n} \left(\sum_{q-1}^{m_i - 1} t_{iq}\, u_{iq} + |F|\, m_i (m_i + n) \right)\right).$$

ở đây $R = \{a_1,...,a_n\}$, $m_i = |K_a|$, and

$u_{iq} = l_{iq}$ nếu $l_{iq} > t_{iq}$ hoặc $u_{iq} = 1$ nếu $l_{iq} = t_{iq}$

Trong trường hợp $l_{iq} \leq (\forall i, \forall q : 1 \leq q \leq m_i)$, độ phức tạp thuật toán của chúng ta là

$$O\left(n \sum_{i=1}^{n} |K_{ai}| \, (n|F| + |K_{ai}|F| + n|K_a^{-1}|^2)\right).$$

Như vậy, độ phức tạp thuật toán 2.11 là đa thức theo $|R|$, $|F|$, $|K_{ai}|$, $|K_a^{-1}|$. Rõ ràng, trong các trường hợp này nếu $|K_{ai}|$ và $|K_a^{-1}|$ là đa thức (đặc biệt nếu chúng là nhỏ) theo $|R|$ và $|F|$, thì thuật toán của chúng ta là hiệu quả.

Bây giờ chúng ta sử dụng thuật toán 11 để xây dựng quan hệ Armstrong cho sơ đồ quan hệ trong ví dụ dưới đây.

Ví dụ 13

Cho s = <R, F> là một sơ đồ quan hệ, ở đây R = {a,b,c,d} và F = {{a,d} $\rightarrow$R, {a} $\rightarrow${a,b,c},{b,d} $\rightarrow$ {b,c,d}}.

Bởi thuật toán 8, chúng ta thu được K_a = {a}.K_b = {{a},{b}},K_c ={{a},{b,d},{c}}, K_d = {d}.

Trên cơ sở thuật toán 1, ta có K_a^{-1} = {b,c,d }, K_b^{-1} = {c,d}, K_c^{-1} = {{b},{d}}, K^{-1}_d = {a,b,c}.

Do đó, N(F) = {{a,b,c }, {b,c,d}, {c,d}, {b}, {d}}. Khi đó ta xây dựng quan hệ r như sau:

a	b	c	d
0	0	0	0
0	0	0	1
2	0	0	0
3	3	0	0
4	0	4	4
5	5	5	0

Bây giờ chúng ta xây dựng một thuật toán tìm một sơ đồ quan hệ s từ một quan hệ cho trước sao cho quan hệ này là quan hệ Armstrong của s.

Thuật toán 14 (Tìm một khoá tối thiểu từ tập các phản khoá).

Vào: Cho K là một hệ Sperner, H là một hệ Sperner, và $C = \{b_1,...,b_m\} \subseteq R$ sao cho $H^{-1} = K$ và $\exists B \in K : B \subseteq C$.

Ra : $D \in H$

Bước 1: Đặt $T(0) = C$

Bước i+1: Đặt $T = T(i) - b_{i+1}$

$$T(i+1) = \begin{cases} T & \text{nếu } \forall B \in K : T \not\in B \\ T(i) & \text{ngược lại} \end{cases}$$

Cuối cùng đặt D = T(m)

Bổ đề 15. Nếu K là tập các phản khoá thì $T(m) \in H$.

Bổ đề 16. Cho H là một hệ Sperner trên R, và $H^{-1} = \{ B_1,...,B_m\}$ là tập các phản khoá của H, $T \subseteq H$ .Khi đó $T \subset H$, $T \neq \emptyset$ nếu và chỉ nếu tồn tại $B \subseteq U$ sao cho $B \in T^{-1}$, $B \not\in B_i$ ($\forall i : 1 \leq i \leq m$).

Cơ sở trên bổ đề 16 và thuật toán 14 chúng ta xây dựng thuật toán sau.

Thuật toán 17. Tìm tập các khoá tối thiểu từ tập các phản khoá.

Vào: Cho $K = \{B_1,...,B_k\}$ là một hệ Sperner trên R

Ra: H mà $H^{-1} = K$

Bước 1: Nhờ thuật toán 2.14 chúng ta tính A_1, đặt $K(1) = A_1$

Bước i+1: Nếu có $B \in K_i^{-1}$ sao cho $B \notin B_j$ ($\forall$ j: $1 \leq j \leq k$), thì bởi Thuật toán 2.14 chúng ta tính A_{i+1} , ở đây $A_{i+1} \in H$, $A_{i+1} \subseteq B$. Đặt $K(i+1) = K(i) \cup A_{i+1}$. Trong trường hợp ngược lại ta đặt H=K(i).

Mệnh đề 18. Độ phức tạp của Thuật toán 17 là

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

ở đây $|R| = n$, $|K| = k$, $|H| = m$, ý nghĩa của l_q , t_q, u_q, xem trong mệnh đề 3.

Rõ ràng , trong các trường hợp mà $l_q \leq k$ ($\forall$ q : $1 \leq q \leq m-1$) độ phức tạp thời gian của thuật toán là $O (|R|^2 |K|^2 |H|)$. Dễ thấy trong các trường hợp này thuật toán 2.17 tìm tập các khoá tối tiểu có độ phức tạp thời gian là đa thức trong kích thước của R, K, H.

Nếu $|H|$ là đa thức theo $|R|$ và $|K|$, thì thuật toán là hiệu quả. Có thể thấy rằng nếu số lượng các phần tử của H là nhỏ thì thuật toán 17 là rất hiệu quả.

Bổ đề 19.

Cho F là một họ f trên R, $a \in R$. Đặt $L_F(A) = \{a \in R: (A , \{a\}) \in F\}$, $Z_F = \{A : L_F (A) = A\}$. Rõ ràng,

$R \in Z_F$, $A,B \in Z_F \rightarrow A \cap B \in Z_F$. Kí pháp N_F là hệ sinh tối tiểu Z_F. Đặt $M_a = \{ A \in N_F : a \notin A, \exists B \in N_F: a \notin B, A \subset B\}$. Khi đó $M_a = MAX(F,a)$, ở đây $MAX(F,a) = \{A \subseteq U : A$ là một tập cực đại không rỗng mà $(A,\{a\}) \notin F\}$.

Lời giải:

Biết rằng [27] $MAX(F,a) \subseteq N_F$ (1). Giả thiết rằng $A \in M_a$. Bởi $A \in N_F$, có nghĩa là $L_F(A) = A$, và $a \notin A$, ta thu được $(A, \{a\}) \notin F$. Từ (1) và phù hợp với định nghĩa của M_a ta có $A \in MAX(F,a)$.

Ngược lại, Nếu $A \in MAX(F,a)$ thì do (1) ta có $A \in N_F$ (2). Do $(A,\{a\}]) \notin F$ và từ (2) ta thu được $a \notin A$. Phù hợp với định nghĩa của $MAX(F,a)$ ta có $A \in M_a$.□

Trên cơ sở Thuật toán 17 và Bổ đề 19, ta xây dựng thuật toán dưới đây để tìm SĐQH s = <R,F> cho một quan hệ r cho trước sao cho $F^+ = F_r$.

Thuật toán 20. (Tìm SĐQH)

Vào: r là quan hệ trên R

Ra: s = <R,F> mà $F^+ = F_R$

Bước1: Từ r ta tính hệ bằng nhau E_r

Bước 2: Đặt $N_r = \{ A \in E_r : A \neq \cap \{ B \in E_r : A \subset B\}\}$

Bước 3: Với mỗi $a \in R$ ta xây $N_a = \{A \in N_r : a \notin A \ \exists \ B \in N_r : a \notin B , A \in B \}$. Sau đó, bởi Thuật toán 17 ta xây họ H_a($H_a^{-1} = N_a$)

Bước 4: Xây s = <R,F>, ở đây $F = \{A \rightarrow \{a\} : \forall a \in R , A \in H_a, A \neq \{a\}\}$

Mệnh đề 21.

$F_R = F^+$

Lời giải: Vì F_R là một họ f trên R, có thể thấy $N_{F\,r} \subset E_r$, ở đây N_{Fr} là hệ sinh nhỏ nhất của Z_{Fr}. Do định nghĩa của hệ sinh nhỏ nhất ta có $N_r = N_{Fr}$. Do đó ta có $N_a = M_a$. Từ định nghĩa của tập phản khoá và định nghĩa của tập K_a ta có $H_a = K_a$. Tù đó ta thu được $F^+ \subseteq F_R$.

Ngược lại, nếu $A \rightarrow B = \{b_1,...,b_t\} \in F_R$ thì bởi việc xây dựng của F ta thu được $A \rightarrow \{b_i\} \in F^+$ với mỗi i=l,...,t. Vì không có phụ thuộc hàm tầm thường $\{a\} \rightarrow \{a\}$ trong F, dễ thấy với mọi i=1,...,t, nếu không có phụ thuộc hàm $B \rightarrow \{b_i\} \in F$, ở đây $B \subseteq U - b_i$, thì $b_i \in A$. Từ đó ta có $A \rightarrow B \in F^+$. □

Có thể thấy E_r, N_r, N_a với $a \in R$ được xây trong thời gian đa thức theo kích thước của r. Rõ ràng, việc xây dựng F phụ thuộc vào kích thước của H_a($\forall a \in R$). Do đó, độ phức tạp thời gian tồi nhất của Thuật toán 20 là

n $m_i - 1$

$$O\left(n \sum_{i=1}^{n} \left(\sum_{q=1}^{m_i - 1} (k_i l_{iq} + n t_{iq} u_{iq}) + k_i^2 + n \right)\right)$$

ở đây $R = \{a_1, ..., a_n\}$, $|N_{ai}| = k_i$, $|H_{ai}| = m_i$, ý nghĩa của các l_{iq}, t_{iq}, u_{iq} xem các Mệnh đề 3 và 18.

Dễ thấy, nếu $l_{iq} \le k_i$ ($\forall i, \forall q : 1 \le q \le m_i - 1$), thì độ phức tạp thời gian của thuật toán của chúng ta là

$$\mathbf{O}\left(n^2 \sum_{i=1}^{n} k_i^2 m_i\right)$$

Bởi vì k_i là đa thức theo kích thước của r, trong các trường hợp nếu is m_i là đa thức theo kích thước của r, thì thuật toán của chúng ta là hiệu quả. Lúc đó độ phức tạp của nó là đa thức theo kích thước của r. Nếu $|H_a|$ là nhỏ thì thuật toán của ta rất hiệu quả.

Bây giờ, nhờ Thuật toán 20 chúng ta xây dựng SĐQH s = <R,F> cho quan hệ sau đây.

Ví dụ 22 r là quan hệ sau đây trên R = {a, b, c, d}:

a	b	c	d
6	6	6	0
0	2	0	2
0	0	0	0
0	0	0	3
4	4	0	0
5	0	5	5
1	0	0	0

Dễ thấy là

E_R = {{a,b,c}, {b,c,d}. { a,c}, { b,c}, {c,d}.{b}, {c},{d}, ∅},

N_R = {[a,b,c}, {b,c,d},{a,c},{c,d},{b},{d}},

N_a = {b,c,d}, N_b= {{a,c},{c,d}},

N_c = {{b},{d}}, N_d ={a,b,c}

Ta có H_a = {a}, H_b = {{b}, {a,d}}, H_c = {{a}, {b,d},{c}}, H_d = {d}.

Ta xây dựng s = (R,F) như sau:

R = {a,b,c,d}, F = {{a,d} $\rightarrow${b},{a} $\rightarrow${c},{b,d} $\rightarrow${c}}.

Chúng ta trình bày hai kết quả cơ bản về độ phức tạp thuật toán cho việc xây dựng quan hệ Armstrong cho một SĐQH cho trước và ngược lại.

Định lí 23

Độ phức tạp thời gian cho việc tìm kiếm một quan hệ Armstrong của một SĐQH cho trước là hàm số mũ theo số lượng của các thuộc tính.

Định lí 24

Độ phức tạp thời gian cho việc tìm kiếm một SĐQH s = <R,F> từ một quan hệ r cho trước sao cho $F_r = F^+$ là hàm số mũ theo số lượng các thuộc tính.

Chương 3

Các dạng chuẩn và các thuật toán liên quan

Việc chuẩn hoá các quan hệ cũng như các sơ đồ quan hệ đóng một vai trò cực kì quan trọng trong việc thiết kế các hệ quản trị cơ sở dữ liệu trên mô hình dữ liệu của Codd. Nhờ có chuẩn hoá các quan hệ và các sơ đồ quan hệ chúng ta tránh được việc dư thừa dữ liệu và tăng tốc độ của các phép toán xử lí quan hệ.

3.1 Các khái niệm cơ bản

Chúng ta định nghĩa các dạng chuẩn như sau.

Cho r = $\{h_1,...,h_m\}$ là quan hệ trên R = $\{a_1 ...., a_n\}$

Định nghĩa 1. (Dạng chuẩn 1 - 1NF):

r là dạng chuẩn 1 nếu các phần tử của nó là sơ cấp.

Khái niệm sơ cấp hiểu ở đây là giá trị $h_i(a_j)$ (i=1,...,m; j=1,...,n) không phân chia được nữa.

Định nghĩa 2 (Dạng chuẩn 2 - 2NF)

r là dạng chuẩn 2 nếu:

- r là dạng chuẩn 1

- $A \rightarrow \{a\} \notin F_r$ đối với mọi khoá tối thiểu K, $A \subset K$ và a là thuộc tính thứ cấp.

Định nghĩa 3. (Dạng chuẩn 3 - 3NF):

r là dạng chuẩn 3 nếu:

$A \rightarrow \{a\} \notin F_r$ đối với A mà $A^+ \neq R$, $a \notin A$, $a \notin \cup K$

Có nghĩa rằng :

- K là một khoá tối tiểu

- a là thuộc tính thứ cấp

- A không là khoá

- $A \rightarrow \{a\}$ không đúng trong r

Định nghĩa 4. (Dạng chuẩn Boye-Codd - BCNF)

r là dạng chuẩn của Boye-Codd nếu:

$A \rightarrow \{a\} \notin F_r$ đối với A mà $A^+ \neq R$, $a \notin A$

Nhận xét 5

Qua định nghĩa, ta có thể thấy dạng chuẩn

BCNF là 3NF và 3NF là 2NF. Tuy vậy, chúng ta có thể đưa ra các ví dụ chứng tỏ có quan hệ là 2NF nhưng không là 3NF và có quan hệ là 3NF nhưng không là BCNF.

Nói cách khác là lớp các quan hệ BCNF là lớp con thực sự của lớp các quan hệ 3NF và lớp các quan hệ 3NF này lại là lớp con thực sự của lớp các quan hệ 2NF.

Đối với s = <F,R> thì các dạng chuẩn 2NF, 3NF, BCNF trong đó ta thay F_r bằng F^+.

Chú ý là đối với sơ đồ quan hệ ta không có dạng 1NF.

Nhận xét 5 cũng đúng cho các dạng chuẩn của sơ đồ quan hệ. Chúng ta xem ví dụ sau

Ví dụ 6.

Cho s = <R, F>, s' = <R, F' > là hai SĐQH trên R = {a, b, c, d} và

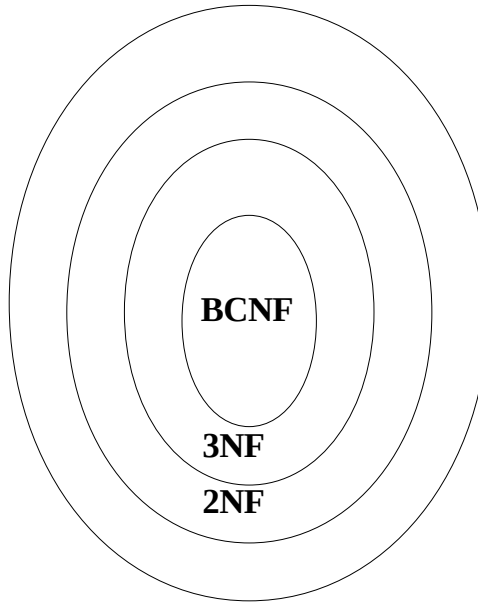
F = {{a} $\rightarrow${c}, {b} $\rightarrow${d}, {c} $\rightarrow${a, b, d}}.

F' = {{a, b} $\rightarrow${c}, {d} $\rightarrow${b}, {c} $\rightarrow${a, b, d}}.

Dễ thấy {a}, {c} là các khoá tối tiểu của s, {b} là thuộc tính thứ cấp. Do đó, s là 2NF, nhưng không là 3NF.

Rõ ràng, {a, b}, {c} là các khoá tối tiểu của s'. Hiển nhiên s là 3NF. Vì ta có {d} $\rightarrow${b}, nên s không là BCNF.

Như vậy việc phân lớp các dạng chuẩn có thể được thể hiện quan hình vẽ sau



2NF

3.2 Dạng chuẩn 2NF

Bây giờ chúng ta nêu ra loại phụ thuộc hàm đặc biệt, mà phụ thuộc dữ liệu này đóng vai trò quan trọng trong dạng chuẩn 2.

Định nghĩa 1.

Một phụ thuộc hàm $A \rightarrow B$ được gọi là sơ cấp nếu không tồn tại một tập hợp $A' \subset A$ sao cho $A' \rightarrow B$. Trong trường hợp này ta cũng nói B phụ thuộc hoàn toàn vào A. Như vậy nếu A là một thuộc tính sơ cấp thì phụ thuộc hàm $A \rightarrow B$ cũng là sơ cấp. Trong trường hợp ngược lại, ta nói B phụ thuộc bộ phận vào A

Định lí 2.

Cho r là một quan hệ trên R. Khi đó r là 2NF khi và chỉ khi

- r là 1NF

- Mỗi thuộc tính thứ cấp của r đều phụ thuộc hoàn toàn vào mọi khoá tối tiểu.

Vì SĐQH không có dạng chuẩn 1, từ Định lí 2 ta có mệnh đề sau

Mệnh đề 3.

Cho s là một sơ đồ quan hệ trên R. Khi đó s là 2NF khi và chỉ khi mọi thuộc tính thứ cấp của s đều phụ thuộc hoàn toàn vào khoá tối tiểu bất kì.

Có thể thấy, bản chất dạng chuẩn 2NF là loại bỏ các phụ thuộc bộ phận giữa các thuộc tính thứ cấp với các khoá tối tiểu.

Định lí 4.

Giả sử s = <R, F> là sơ đồ quan hệ. Đặt M_s = {A - a; a ∈ A, A ∈ K_s}, và F_n là tập tất cả các thuộc tính thứ cấp của s. Đặt l_s = {B : B = C^+ , C ∈ M_s}. Khi đó ta có các tương đương sau:

(1) s là 2NF.

(2) Với mỗi C ∈ M_s: $C^+ \cap F_n = \varnothing$;

(3) Với mỗi B ∈ l_s và a ∈ F_n : $(B - a)^+ = B - a$.

Lời giải: Giả sử s là 2NF. Nếu $F_n = \varnothing$ thì (2) là rõ ràng. Giả thiết rằng $F_n \neq \varnothing$. Do định nghĩa của F_n và của M_s, (3) là hiển nhiên.

Giả sử rằng chúng ta có (2) và $F_n \neq \varnothing$. Nếu có B ∈ l_s, và a ∈ F_n : B - a ⊂ ($B - a^+$). Từ định nghĩa của l_s có C ∈ M_s : $C^+ = B$. Rõ ràng rằng a ∈ $(B - a)^+$. Phù hợp với định nghĩa của bao đóng ta có $(B - a)^+ = C^+ = B$. Từ đó ta thu được a ∈ C^+. Do vậy, $C^+ \cap F_n \neq \varnothing$. Điều này là vô lý. Do đó ta thu được (3).

Bây giờ, giả sử chúng ta có (3) và $F_n \neq \varnothing$. Giả thiết rằng tồn tại D ⊂ A, A ∈ K_s(*) và a ∈ F_n, a ∉ D, nhưng D →{a} ∈ F^+. Do (*) và phù hợp với việc xây

dựng của M_s và l_s thì có $C \in M_s : D \subseteq C$. Hiển nhiên rằng $a \notin C$. Rõ ràng, $D^+ \subseteq C^+$ và $a \in C^+$. Đặt $B = C^+$. Có thể thấy $C \subseteq B - a$. Do đó, $B - a \subset C^+ = (B - a)^+$. Điều này mâu thuẫn với $(B - a)^+ = B - a$. Như vậy chúng ta có (1). □

Từ định lý 4 trực tiếp suy ra kết quả sau:

Hệ quả 5.

Giả sử s = (R, F) là một sơ đồ quan hệ. Ký pháp F_n là tập tất cả những thuộc tính thứ cấp của s, và $G_s = \{B - F_n : B \in K_s^{-1}\}$. Khi đó nếu đối với mọi $C \in G_s : C^+ = C$ thì s là 2NF.

Cho s = <R, F> là một sơ đồ quan hệ trên R. Đặt $Z(s) = \{X^+ : X \subseteq R\}$. Chúng ta nói rằng s là đơn nếu F chứa chỉ các phụ thuộc hàm dạng $\{a\} \rightarrow \{b\}$. Biết rằng [16] s là đơn nếu và chỉ nếu đối với mọi $A, B \in Z(s) : A \cup B \in Z(s)$. Rõ ràng, từ điều đó ta có $(A \cup B)^+ = A^+ \cup B^+$.

Mệnh đề 6.

Cho s = <R, F> là một sơ đồ quan hệ đơn. Đặt F_n là tập tất cả những thuộc tính thứ cấp của s, và $G_s = \{B - F_n : B \in K_s^{-1}\}$. Khi đó s là 2NF nếu và chỉ nếu với mọi $C \in G_s : C^+ = C$.

Lời giải: Giả sử rằng s là 2NF. Nếu $F_n = \phi$ thì bởi định nghĩa của phản khoá ta có $C^+ = C$. Nếu $F_n \neq$

ϕ thì giả thiết rằngcó $C \in G_s : C^+ \neq C$. Bởi định nghĩa của G_s thì tồn tại $B \in K_s^{-1} : C \cup F_n = B$. Biết rằng [9] F_n là giao của tất cả các phản khoá chúng ta có $C \subset B$. Do đó, $C^+ \subseteq B$, $C^+ \cap F_n \neq \phi$. Ta ký pháp các phần tử của C là $c_1,..., c_l$. Bởi vì s là đơn chúng ta thu được $\{c_1\}^+ \cup ... \cup \{c_l\}^+ = C^+$. Do đó tồn tại $c_1 \in C$ sao cho $\{c_1\}^+ \cap F_n \neq \varnothing$. Hiển nhiên c_1 là thuộc tính cơ bản. Điều này trái với việc s có 2NF. Do đó, $C^+ = C$.

Ngược lại bởi hệ quả 5 nếu ta có $C^+ = C$ đối với mọi $C \in G_s$, thì s là 2NF. □

Cho r là một quan hệ trên R. Đặt $A^+_r = \{a : a \in R, A \rightarrow^f_r \{a\}\}$, r là quan hệ đơn nếu đối với mọi $A, B \subseteq R : (A \cup B)^+_r = A^+_r \cup B^+_r$.

Biết rằng đối với một quan hệ r cho trước thì tập hợp tất cả các phản khoá của r được xây dựng trong thời gian đa thức. Trong [9] chúng ta đã chỉ ra rằng giao của tất cả các phản khoá đúng bằng tập tất cả các thuộc tính thứ cấp. Mặt khác biết rằng [6] nếu sơ đồ quan hệ s = <R, F> là đơn thì độ phức tạp thời gian tìm quan hệ r sao cho $F^+ = F_r$ là đa thức. Từ điều này và mệnh đề 6 ta có

Mệnh đề 7.

Cho s là một sơ đồ quan hệ đơn và r là một quan hệ đơn trên R. Khi đó tồn tại một thuật toán đa thức xác định rằng s (r, tương ứng) có là 2NF.

3.3 Dạng chuẩn 3NF

Trong mục này, chúng ta đưa ra một khái niệm quan trọng thường dùng để mô tả dạng chuẩn 3NF

Định nghĩa 1.

Một phụ thuộc hàm $A \rightarrow C$ được gọi là trực tiếp nếu không có B ($B \neq A$ và $B \neq C$) sao cho $A \rightarrow B$ và $B \rightarrow C$ nhưng B không phụ thuộc hàm vào A hoặc C không phụ thuộc hàm vào B. Trong trường hợp nếu có B như vậy thì B được gọi là tập thuộc tính bắc cầu và $A \rightarrow C$ là phụ thuộc bắc cầu.

Ta có một đặc trưng sau cho dạng chuẩn 3.

Định lí 2.

Cho r là một quan hệ trên R. Khi đó r là 3NF nếu và chỉ nếu

- r là 2NF

- Không có một thuộc tính thứ cấp nào của r phụ thuộc bắc cầu vào một khoá tối tiểu.

Từ Định lí 2 đối với SĐQH ta cũng có kết quả sau

Mệnh đề 3.

Giả sử s là một sơ đồ quan hệ trên R. Khi đó s là 3NF nếu và chỉ nếu

- s là 2NF

- Mọi thuộc tính thứ cấp của s phụ thuộc trực tiếp vào mỗi khoá tối tiểu.

Trong dạng chuẩn 3NF chúng ta loại bỏ các phụ thuộc bộ phận, phụ thuộc bắc cầu giữa các thuộc tính thứ cấp với các khoá tối tiểu.

Chúng ta trình bày thêm một số đặc trưng của các SĐQH dạng 3NF.

Cho s = <R, F> là một sơ đồ quan hệ trên R. Từ s chúng ta xây dựng $Z(s) = \{X^+ : X \subseteq R\}$, và tính hệ sinh N_s của Z(s). Chúng ta đặt

$T_s = \{A \in N_s : \exists\ B \in N_s : A \subset B\}$

Biết rằng [1] đối với một sơ đồ quan hệ s cho trước thì tồn tại một quan hệ r là quan hệ Amstrong của s. Mặt khác bởi Định lý 17 và Hệ quả 18 mục 2.2, mệnh đề dưới đây là rõ ràng

Mệnh đề 4.

S là một SĐQH. Khi đó $K_s^{-1} = T_s$.

Định lí 5.

Cho s = <R, F> là một sơ đồ quan hệ. Đặt F_n là tập tất cả các thuộc tính thứ cấp của s. Khi đó s là

3NF nếu và chỉ nếu $\forall B \in K_s^{-1}, a \in F_n : (B - a)^+ = B - a$.

Lời giải: Giả sử $F_n \neq \emptyset$. Có thể thấy rằng s là 3NF thì đối với mỗi $B \in K_s^{-1}, a \in F_n : B - a = (B - a)^+$.

Ngược lại, nếu s không là 3NF thì tồn tại một tập A và $a \in F_n : a \notin A$ sao cho $A \rightarrow \{a\} \in F^+$ và $A^+ \neq R$. Phù hợp với Mệnh đề 4 có $B \in K^{-1}_r$ sao cho $A^+ \subseteq B$. Từ $a \in A^+$ chúng ta có $a \in B$. Do $a \notin A$ ta có $A \subseteq B - a$. Do đó ta thu được $B - a \subset (B - a)^+$. □

Định lí 6.

Giả sử r là một quan hệ trên R. Khi đó r là 3NF nếu và chỉ nếu với mọi $A \in E_r$, $a \in A$ và a là thuộc tính thứ cấp thì $\{A- a \}_r^+ = A- a$, ở đây E_r là hệ bằng nhau của r.

Từ Định lí 5 ta có hệ quả sau

Hệ quả 7.

Giả sử s là một sơ đồ quan hệ trên R. Khi đó s là 3NF nếu và chỉ nếu với mọi A : $A^+ = A$, $a \in A$ và a là thuộc tính thứ cấp thì $\{A - a \}^+ = A- a$.

3.4. Dạng chuẩn BCNF

Trong mục này, chúng ta đưa ra một số các đặc trưng của dạng chuẩn BCNF cho sơ đồ quan hệ và quan hệ.

Định nghĩa 1.

Giả sử r là một quan hệ trên R, $A,B \subseteq R$ và $A \rightarrow B$.

Khi đó ta nói A là tập sinh của B nếu

- $|A| < |B|$,

- Không tồn tại tập con thực sự của A mà xác định hàm cho B.

Tập C là tập sinh của quan hệ r nếu có một tập D nào đó để C là tập sinh của D.

Định lí 2

Giả sử r là quan hệ trên R. Khi đó r là BCNF khi và chỉ khi mọi tập sinh của r đều là khoá.

Mệnh đề 3

Cho s = <R, F> là một sơ đồ quan hệ. Đặt F_n là tập tất cả các thuộc tính thứ cấp của s. Khi đó

s là BCNF nếu và chỉ nếu $\forall B \in K_s^1, a \in B : (B - a)^+ = B - a$.

Lời giải: Dễ thấy nếu s là BCNF thì $(B - a)^+ = B - a$ đối với $B \in K_s^{-1}$ và $a \in B$.

Ngược lại giả thiết s không là BCNF. Khi đó tồn tại $A \rightarrow \{a\} \in F^+$ ở đây $A^+ \neq R$ và $a \notin A$. Bởi mệnh đề 4 mục trên, ta có $B \in K^{-1}$ sao cho $A^+ \subseteq B$. Rõ ràng, $a \in B$ và $A \subseteq B - a$. Từ đó, ta có $(B - a)^+ = B$. □

Định lí 4.

Giả sử r là một quan hệ trên R. Khi đó r là BCNF nếu và chỉ nếu với mọi $A \in M_r$, $a \in A$ thì $\{A- a\}_r^+ = A- a$, ở đây M_r là hệ bằng nhau cực đại của r.

Giả sử $A \rightarrow B$ là một phụ thuộc hàm. Chúng ta gọi phụ thuộc hàm này là tầm thường nếu $B \subseteq A$. Ngược lại trong trường hợp này, chúng ta gọi nó là phụ thuộc hàm không tầm thường.

Định lí 5

Giả sử s = < R,F> là một sơ đồ quan hệ trên R. Khi đó s là BCNF nếu và chỉ nếu với mọi $A \rightarrow B \in F$ và $A \rightarrow B$ không tầm thường thì $A^+ = R$.

3.5. Các thuật toán liên quan

Trên cơ sở các định lí đã trình bày ở các mục trên, chúng ta xây dựng các thuật toán để xác định dạng chuẩn cho các quan hệ hoặc sơ đồ quan hệ cho trước.

Đầu tiên chúng ta xây dựng thuật toán xác định một quan hệ cho trước có là 3NF hay không.

Thuật toán 1.

Đầu vào: r = {h_1, ..., h_m }là một quan hệ trên R

Đầu ra : r là 3NF ?

Bước 1: Từ r chúng ta xây dựng một tập E_r = {E_{ij} : m ≥ j > i ≥ 1}, ở đây E_{ij} = { a ∈ R : $h_j(a) = h_j(a)$}.

Bước 2: Từ E_r chúng ta xây dựng một tập M = {B ∈ P(R) : Tồn tại $E_{ij} \in E_r$: E_{ij} = B}.

Bước 3: Từ M xây dựng tập M_r = { B ∈ M : Với mọi B' ∈ M : B ⊄ B'}.

Có thể thấy rằng M_r tính được bằng một thuật toán thời gian đa thức.

Bước 4: Xây dựng tập V = ∩M_r.

Bước 5: r là 3NF nếu với mọi B ∈ M_r , a ∈ V : $\{B - a\}_r^+$ = B - a. Ngược lại r không là 3NF.

Ví dụ : Cho quan hệ r sau

A	B	C	D	E
0	0	1	0	1
1	1	0	0	1
2	2	0	1	3
1	2	3	1	0

1 1 1 0 2

Khi đó E_{12}= DE, E_{13}= ∅, E_{14}= ∅, E_{15}= D, E_{23}= C, E_{24}= A, E_{25}=AB,

E_{34}=BD, E_{35}=∅, E_{45}=A.

Như vậy ta có M_r= {DE,AB,BD,C}. Dễ thấy DE ∩ AB ∩ BD ∩ C = ∅.

Cho nên r là 3NF.

Trên cơ sở Định lí 4 mục trên chúng ta xây dựng thuật toán dưới đây

Thuật toán 2.

Đầu vào: r = $\{h_1, ..., h_m\}$ là một quan hệ trên R

Đầu ra: r là BCNF ?

Bước 1: Từ r chúng ta xây dựng một tập $E_r = \{E_{ij} : m \geq j > i \geq 1\}$ và $E_{ij} = \{a \in R : h_j(a) = h_j(a)\}$

Bước 2: Từ E_r chúng ta xây dựng một tập M = $\{B \in P(R) :$ Tồn tại $E_{ij} \in E_r : E_{ij} = B\}$

Bước 3: Từ M xây dựng tập $M_r = \{B \in M :$ Với mọi $B' \in M : B \not\subset B'\}$. Có thể thấy rằng M_r tính được bằng một thuật toán thời gian đa thức.

Bước 4: r là BCNF nếu với mọi $B \in M_r$, $a \in B : \{B - a\}_r^+ = B - a$. Ngược lại r không là BCNF.

Ví dụ : Cho quan hệ r :

A	B	C	D	E
2	0	1	1	1
1	1	0	0	1
2	2	0	3	3
1	2	3	3	0
1	1	1	0	2

Khi đó E_{12}= {E}, E_{13}= {A}, E_{14}=∅ , E_{15}= {C}, E_{23}= {C}, E_{24}= {A}, E_{25}={A,B}, E_{34}={B,D}, E_{35}=∅, E_{45}={A}.

Như vậy ta có M_r= {{A,B},{B,D},{C},{E}}. Có thể kiểm tra rằng {A,D} - A = D và $\{D\}^+_r$ = {B,D}. Vì thế r không là BCNF.

Nhờ Định lí thuật toán dưới đây được xây dựng

Thuật toán 3.

Đầu vào: s = <R,F> là một sơ đồ quan hệ trên R, với

$F = \{ A_1 \rightarrow B_1,..., A_m \rightarrow B_m \}$

Đầu ra: s là BCNF ?

Bước 1: Nếu $A_1 \rightarrow B_1$ là phụ thuộc hàm không tầm thường và A_1^+ # R thì dừng và kết luận s không là BCNF. Ngược lại thì chuyển sang bước tiếp theo.

.....

Bước m: Giống như bước 1 nhưng đối với $A_m \rightarrow B_m$.

Bước m+1: s là BCNF.

Ví dụ : Cho sơ đồ quan hệ s =<R,F>

R = {a,b,c,d,e}

F={{a,b}→{d},{b,c}→{e}, {d}→{c}}

Ta có $\{a,b\}^+ = R$, nhưng $\{b,c\}^+ \neq R$. Vậy s không là BCNF.

Vì thời gian tính bao đóng của một tập thuộc tính bất kì của một sơ đồ quan hệ hoặc một quan hệ là đa thức. Cho nên chúng ta có các kết luận sau.

Định lí 4.

Cho trước một quan hệ r và một sơ đồ quan hệ s. Khi đó đều tồn tại một thuật toán có độ phức tạp thời gian đa thức theo kích thước của r (s) để kiểm tra r (s) có là BCNF hay không.

Định lí 5

Cho trước r là một quan hệ trên R. Khi đó tồn tại một thuật toán có độ phức tạp thời gian đa thức để kiểm tra r có là 3NF hay không.

Tuy vậy, đối với đầu vào là s thì đây lại là bài toán NP đầy đủ.

Định lí 6

Cho trước s là một sơ đồ quan hệ trên R. Khi đó bài toán xác định s có là 3NF hay không là NP - đầy đủ.

Có nghĩa là cho đến nay, độ phức tạp thời gian của bài toán này không là đa thức.

- Với trường hợp 2NF, các câu hỏi tương tự cho cả r lẫn s còn là bài toán mở (Chúng tôi phỏng đoán có độ phức tạp thời gian là hàm mũ trở lên)

3.6 Dạng chuẩn của các hệ khoá

Bây giờ chúng ta khảo sát các sơ đồ quan hệ mà tập các khoá tối thiểu của nó là một hệ Sperner cho trước.

Định nghĩa 1.

Cho K là hệ Sperner trên R. Ta nó rằng K là 2NF (3NF, BCNF, tương ứng) nếu với mỗi sơ đồ quan hệ s = <R, F> mà K_s = K thì s là 2NF (3NF, BCNF tương ứng).

Bây giờ chúng ta cho một điều kiện cần và đủ để một hệ Sperner bất kỳ là 2NF.

Cho K là một hệ Sperner trên R. Đặt K_p= {a ∈ R : ∃A ∈ K : a ∈ A}, và K_n = R - K_p. K_p (K_n) được gọi là tập các thuộc tính cơ bản (thứ cấp) của K.

Cho trước sơ đồ quan hệ s = <R, F>, chúng ta nói rằng phụ thuộc hàm A →B ∈ F là thừa nếu hoặc A = B hoặc có C →D ∈ F sao cho C ⊆ A và B ⊆ D.

Định lí 2.

Cho K là hệ Sperner trên R. Khi đó K là 2NF nếu và chỉ nếu K_n = ∅.

Lời giải: Theo định nghĩa của quan hệ 2NF, hệ Sperner 2NF và K_n ta có thể thấy nếu K_n = ∅ thì K là 2NF.

Bây giờ ta giả thiết là K là 2NF. Kí pháp K^{-1} là tập các phản khoá của K. Từ K, K^{-1} ta xây một SĐQH như sau

Đối với mỗi A ⊂ R có B ∈ K^{-1} sao cho A ⊆ B . Đặt C = ∩ {B ∈ K^{-1} :A ⊆ B}. Ta lập A →C . Kí pháp T là tập tất cả các phụ thuộc hàm như thế. Đặt F = { E →R : E ∈ K} ∪ (T - Q) , ở đây Q = {X →Y ∈ T : X → Y là phụ thuộc hàm thừa}. Từ Định lí 13 phần 2.2 và định nghĩa của hệ Sperner ta thu được K_s = K .

Rõ ràng, với mỗi SĐQH bất kì s_1 = (R, F_1) sao cho K_{s1} = K và A ⊆ R ta có $A^+_{s1} \subseteq A^+_s$, ở đây A^+_{s1} = {a: A →{a} ∈ F_1^+}. Chúng ta đã chứng minh rằng [9]

K_n là giao của các phản khoá của s. Trên cơ sở việc xây dựng s = <R,F> và phù hợp với định nghĩa của hệ Sperner 2NF ta có $K_n = \emptyset$. □

Dễ thấy SĐQH 3NF là 2NF và nếu tập các thuộc tính thứ cấp là rỗng thì SĐQH này 3NF. Do đó từ Định lí 2 ta suy ra ngay hệ quả sau

Hệ quả 3.

Cho K là hệ Sperner trên R. Khi đó K là 3NF nếu và chỉ nếu $K_n=\emptyset$.

Định nghĩa 4.

Cho K là hệ Sperner trên R. Ta nói K là đơn nhất nếu K xác định duy nhất SĐQH s = <R,F>, theo nghĩa đối với mọi SĐQH s' =<R,F '> mà $K_{s'}$ = K thì ta có $F^+ = F'^+$.

Từ định nghĩa hệ Sperner BCNF và Định nghĩa 4 ta có

Mệnh đề 5.

K là BCNF nếu và chỉ nếu K là đơn nhất.

Như đã biết [5] đối với hệ Sperner cho trước K tồn tại SĐQH s (tương ứng quan hệ r) sao cho K_s =K (tương ứng K_r =K). Ta nói s (tương ứng r) là đơn nhất nếu K_s (tương ứng K_r) xác định duy nhất s

(tương ứng r). Có nghĩa là K_s (tương ứng K_r) là đơn nhất.

Bây giờ ta cho một điều kiện cần và đủ để một SĐQH là đơn nhất.

Định lí 6.

Cho s = (R,F) là SĐQH trên R. Khi đó s là đơn nhất nếu và chỉ nếu đối với mọi a ∈ A, A ∈ $K^{-1}{}_s$: A-a = ∩ {B ∈ $K^{-1}{}_s$: (A- a) ⊂ B}

Lời giải:

Chúng ta biết rằng hệ Sperner K là đơn nhất khi và chỉ khi đối với mọi B ⊆ A,A ∈ K^{-1}, B là giao của các phản khoá. Đặt P_s = {A-a: ∈ $K^{-1}{}_s$, a ∈ A}

Có thể thấy nếu s=<R,F> là đơn nhất thì B ∈ P_s kéo theo B là giao của các phản khoá. Có nghĩa là B = ∩ {A ∈ K_s^{-1}:B ⊂ A}.

Ngược lại giả sử với mỗi B ∈ P_s ta có B = ∩ {A ∈ K_s^{-1}:B ⊆ A}(*).. Do mệnh đề 3 mục 3.4 và Mệnh đề 4 mục 3.3 ta có N_s ⊆ (P_s ∪ K_s^{-1}). Có thể thấy s là BCNF. Trên cơ sở định nghĩa của N_s và Mệnh đề 4 mục 3.3 ta có K_s^{-1} ⊆ N_s. Phù hợp với (*) ta thu được K_s^{-1} = N_s . Vì s là BCNF nên đối với mọi B ⊆ A, A ∈ K_s^{-1} : B^+ = B. Như vậy B là giao của các phản khoá của s. □

Theo định nghĩa của hệ Sperner BCNF, Định lí 6 và Mệnh đề 5 ta cho một điều kiện cần và đủ để một hệ Sperner là BCNF.

Địnhlí 7.

Giả sử K là hệ Sperner trên R. Khi đó K là BCNF nếu và chỉ nếu đối với mọi $a \in A$, $A \in K^{-1} : A - a = \cap \{B \in K^{-1} : (A - a) \subset B\}$.

Phù hợp với Định lí 6 và thuật toán đa thức tìm tập các phản khoá của một quan hệ cho trước, ta có mệnh đề sau

Mệnh đề 8.

Tồn tại thuật toán xác định một quan hệ cho trước có là đơn nhất hay không. Độ phức tạp thời gian của thuật toán này là đa thức theo kích thước của R và r.

Từ Định lí 7 và Mệnh đề 8 trực tiếp kéo theo mệnh đề sau

Mệnh đề 9.

Tồn tại thuật toán đa thức xác định tập các khoá tối tiểu của một quan hệ cho trước là BCNF hay không.

Từ Định lí 6 suy ra ngay hệ quả sau

Hệ quả 10.

Cho K là hệ Sperner trên R . Khi đó có thuật toán đa thức xác định hệ Sperner H có là đơn nhất hay không, ở đây $H^{-1} = K$.

3.7. Ví dụ

Dưới đây chúng ta cho ví dụ minh hoạ việc phân tách một bảng (quan hệ) thành các bảng ở dạng chuẩn 3NF.

Trong một nhà máy, hàng ngày người ta xuất vật tư theo phiếu xuất kho như sau:

Phiếu xuất kho

Số phiếu	Ngày xuất	Người nhận	Địa chỉ người nhận	Mã vật tư
1010001	26/10/96	Phạm An	2 Phố Huế, Hà Nội	10100 20018 10703
1020004	12/01/97	Trần Hà	14 Lê Lợi, TP. HCM	30101
1170003	17/03/97	Trần Hà	14 Lê Lợi, TP. HCM	10100 20904

Trong ví dụ này có hai thuộc tính không sơ cấp. Đó là :

- 'Địa chỉ người nhận' là một thuộc tính tổng hợp những thuộc tính sơ cấp sau: 'Số và phố', 'Tên TP'.

- 'Mã vật tư' là danh sách các vật tư của một hoá đơn, có chiều dài không nhất định, cần được tách riêng ra từng vật tư.

Ta có thể biến đổi quan hệ phiếu xuất kho sang dạng chuẩn 1 như sau

Phiếu xuất kho

Số phiếu	Ngày xuất	Người nhận	Số và phố	Thành phố	Mã vật tư
1010001	26/10/96	Phạm An	2 Phố Huế	Hà Nội	10100
1010001	26/10/96	Phạm An	2 Phố Huế	Hà Nội	20018
1010001	26/10/96	Phạm An	2 Phố Huế	Hà Nội	10703
1020004	12/01/97	Trần Hà	14 Lê Lợi	TPHCM	30101
1170003	17/03/97	Trần Hà	14 Lê Lợi	TPHCM	10100
1170003	17/03/97	Trần Hà	14 Lê Lợi	TPHCM	20904

Trong quan hệ Phiếu xuất kho ta nhận thấy là tập {Số phiếu, Mã vật tư} là khoá tối tiểu. Dễ thấy các thuộc tính Ngày xuất, Người nhận, Số và phố, Thành phố phụ thuộc hàm vào thuộc tính số phiếu. Như vậy, quan hệ Phiếu xuất kho không là 2NF (Nếu lưu trữ và xử lí trên quan hệ này sẽ dẫn đến trùng lặp dữ liệu). Do đó, ta tách thành 2 quan hệ riêng biệt:

Phiếu kho

Số phiếu	Ngày xuất	Người nhận	Số và phố	Thành phố
1010001	26/10/96	Phạm An	2 Phố Huế	Hà Nội
1020004	12/01/97	Trần Hà	14 Lê Lợi	TPHCM
1170003	17/03/97	Trần Hà	14 Lê Lợi	TPHCM

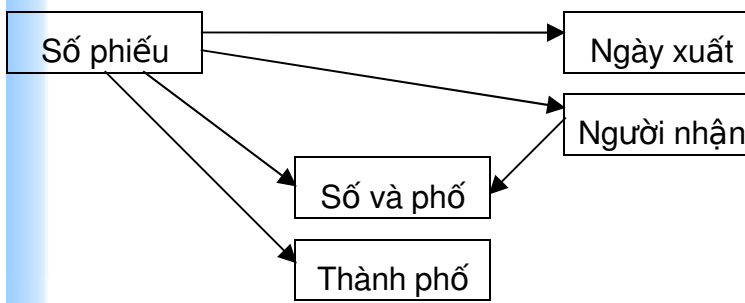
Vật tư

Số phiếu	Mã vật tư
1010001	10100
1010001	20018
1010001	10703

1020004	30101
1170003	10100
1170003	20904

Ta có thể thấy quan hệ vật tư là 3NF.

Trong quan hệ Phiếu kho là 2NF ở trên, ta thấy trên đồ thị của các phụ thuộc hàm có hai con đường để đi từ Số phiếu đến {Số và phố, Thành phố} hoặc đi qua thuộc tính Người nhận. Như vậy tồn tại phụ thuộc bắc cầu trong quan hệ này.



Điều này chứng tỏ quan hệ này chưa là 3NF, Nếu ta lưu quan hệ này thì khi xử lí sẽ dẫn đến trùng

lặp địa chỉ của người nhận. Do vậy ta tách nó thành hai thực thể riêng biệt:

Phiếu

Số phiếu	Ngày xuất	Người nhận
1010001	26/10/96	Phạm An
1020004	12/01/97	Trần Hà
1170003	17/03/97	Trần Hà

Người nhận

Người nhận	Số và phố	Thành phố
Phạm An	2 Phố Huế	Hà Nội
Trần Hà	14 Lê Lợi	TP. HCM

Như vậy, ta đã tách quan hệ phiếu xuất kho thành 3 quan hệ dạng chuẩn 3 là phiếu, người nhận, vật tư.

Chương 4
Một số phép toán xử lí bảng

Ngôn ngữ xử lí dữ liệu là một phần quan trọng trong các hệ quản trị cơ sở dữ liệu. Ngay từ năm 1970 E.F.Codd đã đưa ra hai ngôn ngữ xử lí dữ liệu chính. Đó là ngôn ngữ đại số quan hệ và ngôn ngữ tính toán quan hệ (Chủ yếu dựa vào phép toán tân từ). Hầu hết ngôn ngữ xử lí của các hệ quản trị cơ sở dữ liệu lớn hiện nay đều chứa ngôn ngữ đại số quan hệ. Trong chương này chúng tôi trình bày các phép toán cơ bản của ngôn ngữ đại số quan hệ này.

Trong Giáo trình này chúng ta coi bảng, file dữ liệu, quan hệ (theo dạng Codd) là tương đương nhau.

4.1 Các phép toán cơ bản

Để minh hoạ bằng các ví dụ làm sáng tỏ tính chất của các phép toán xử lí bảng chúng ta cho 2 quan hệ sau:

A	B	C
a	a	b
a	c	b
b	c	d
a	a	d

Quan hệ r

D	A	E
a	a	b
b	c	d
e	f	g

Quan hệ t

Hình 1

1. Phép hợp (r ∪ t)

Giả sử r và t là các quan hệ n cột

Khi đó quan hệ hợp r ∪ t là quan hệ n cột bao gồm các bản ghi (dòng) của cả r lẫn t.

Chú ý những bản ghi giống nhau chỉ giữ lại một.

Nếu r và t là các quan hệ có tên các cột khác nhau thì quan hệ hợp không có tên các cột

Với r, t trong hình 1, ta có r ∪ t =

a	a	b
a	c	b
b	c	d
a	a	d
e	f	g

2. Phép trừ (r - t)

Giả sử r và t là hai quan hệ n cột.

Quan hệ hiệu (kí hiệu là r-t) là quan hệ n cột mà bao gồm các dòng của r nhưng không có mặt trong t

Nếu r và t có các tên cột khác nhau, thì quan hệ hiệu không có tên các cột.

Ví dụ: r - t =

a c b

a a d

3. Phép giao (r ∩ t)

Giả sử t và t là hai quan hệ n cột.

Khi đó quan hệ giao của r và t là quan hệ n cột bao gồm các bản ghi có mặt cả ở trong r lẫn t.

Trong trường hợp nếu r và t có các tên cột khác nhau (tên các thuộc tính) thì các cột của quan hệ giao không có tên.

Ví dụ: r ∩ t =

a a b

b c d

4. Tích Đề các

Giả sử có quan hệ r có n cột (R_1= {a_1,...,a_n}) và t có m cột (R_2= {b_1,...,b_m})

a_1 .....a_n b_1 .....b_m

r = ... t = ...

Tích Đề các:

r x t nếu $(r_1.....r_n) \in r$ và

$(t_1.....t_m) \in t$

Thì $(r_1,....,r_n, t_1,....., t_m) \in rxt$

Ví dụ:

A	B	C
a	a	b
a	c	b
b	c	d
a	a	d

r =

D	A	E
a	a	b
b	c	d
e	f	g

t =

Tích Đề các:

r x t =

r.A	B	C	D	t.A	E
a	a	b	a	a	b
a	a	b	b	c	d
a	a	b	e	f	g
a	c	b	a	a	b

a	c	b	b	c	d
a	c	b	e	f	g
b	c	d	a	a	b
b	c	d	b	c	d
b	c	d	e	f	g
a	a	d	a	a	b
a	a	d	b	c	d
a	a	d	e	f	g

Tích Đề các là một quan hệ. Trong trường hợp có cột giống nhau (tên giống nhau) cần đánh dấu để phân biệt. Trong ví dụ trên đó là r.A và t.A

Số lượng bản ghi (dòng) là n.m.

5. Phép chiếu

Giả sử có r là một quan hệ gồm m cột ($R_1 = \{a_1...a_m\}$). Khi ấy phép chiếu (Ký hiệu là: Π) lên tập r:

$\Pi_{i1,i2...ip}$ (r)

ij là số thứ tự lấy trong tập từ 1 đến m

j = 1,...,p, ở đây chỉ số p ≤ m.

hoặc $\Pi_{a,b,...,\ t}$ (r), ở đây a, b, ... t là tên các thuộc tính.

Khi đó ta thực hiện phép chiếu như sau:

Giữ lại p cột có số hiệu là i_1, i_2, i_p, loại bỏ các dòng bị trùng nhau.

Ví dụ: (lấy ví dụ trước).

$\Pi_{1,2}$ (r) =

A	B
a	a
a	c
b	c

6. Phép chọn

Phép toán này rút ra các bản ghi thoả mãn điều kiện nào đấy.

Gọi F là một điều kiện nếu nó bao gồm:

- Các toán hạng (hằng, tên thuộc tính)

- Các quan hệ số học (<, =, >, ≤ , ≥ , #)

- Các phép toán logic: và ($\wedge$), hoặc ($\vee$) và phủ định. Riêng với hằng ta bao bằng dấu ' '

Ví dụ: F là điều kiện A = 'a' v C = 'b'

Ký hiệu phép chọn: δ_F (r) (F: điều kiện).

Khi đó δ_F là phép chọn được thực hiện bằng việc rút ra từ r (và loại bỏ các dòng trùng) các bản ghi thoả mãn F.

Ví dụ:

δ_A = 'a' $\vee$C= 'b' (r) = A B C

a a b

a c b

a a d

4.2 Các phép toán khác

Các phép toán trình bày tiếp theo đây được biểu diễn qua các phép toán ở mục trên.

1. Phép chia (r ÷ s):

Giả sử r là quan hệ n cột, t là quan hệ m cột

ở đây n > m và t ≠ ∅. Quan hệ thương của r và t là quan hệ được tạo ra như sau:

- Tích $A_1 = \prod_{1,2,..., n-m}(r)$
- Tích $A_2 = A_1$ x t
- Tích $A_3 = A_2$ - r
- Tích $A_4 = \prod_{1, 2,...., n-m}(A_3)$

Cuối cùng $r \div s = \prod_{1, 2,...., n-m}(r) - A_4$

Như vậy chúng ta có:

$r \div s = \prod_{1, 2,...., n-m}(r) - \prod_{1, 2,...., n-m}((\prod_{1, 2,...., n-m}(r)$ x t) - r)

Ví dụ:

a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	c
b	b	b	b

Quan hệ r

c	d
e	f

Quan hệ t

a	b
e	d

Quan hệ r ÷ t

2. Phép nối θ

Giả sử r là quan hệ n cột, t là quan hệ m cột, θ là một trong các quan hệ số học: < , ≤ , = , # , > , ≥

i và j là tên hoặc là số cột tương ứng của r và t

Khi đó kết quả của phép nối θ hai quan hệ r và t là quan hệ (n + m) cột bao gồm các bản ghi của tích đề các r x t, mà các bản ghi này thoả mãn quan hệ số học θ giữa các giá trị thuộc tính i và j.

Ký pháp nó là r >< t

θ

Có thể thấy r >< t = $\delta_{i\,\theta\,(n+j)}$ (r x t)

iθj

Với ví dụ như ở mục trên ta có:

r.A	B	C	D	t.A	E

a	a	b	a	a	b
a	c	b	a	a	b
a	a	d	a	a	b
b	c	d	b	c	d

r >< t

r.A=D

3. Phép nối

Giả sử có r là một quan hệ n cột (R_1 = {a_1...a_n}), r là quan hệ m cột (R_2 = {b_1,...,b_m}). Khi đó chúng ta ký hiệu r ⋈ t là phép nối.

- Nó được thực hiện bởi biểu thức trên cơ sở các phép toán sau:

r ⋈ t = ∏ i_1, i_2...i_{n+m-q} (δr.A_1 = t.A_1 ∧... ∧r.A_q = t.A_q (r x t))

A_1,...,A_q là q cột có tên trùng nhau ở hai quan hệ r và t.

- Phép toán được thực hiện:

+ Tích Đề các r và t.

+ Chọn trong tích Đề các những bản ghi thoả mãn những cột giống tên nhau phải trùng giá trị trên p cột.

+ Phép chiếu loại bỏ p cột giống nhau (Chỉ giữ lại một).

Ví dụ như hai quan hệ trong hình 1 của phần này, chúng ta có thể thấy r ⋈ t là bảng sau.

A	B	C	D	E
a	a	b	a	b
a	c	b	a	b
a	a	d	a	b

Phép nối là một phép xử lý bảng rất quan trọng. Thông thường chúng ta phân tách file dữ liệu lớn ban đầu thành các file dữ liệu nhỏ ở dạng chuẩn 3NF. Nhờ phép nối các file dữ liệu nhỏ với nhau chúng ta có thể phục hồi file dữ liệu lớn ban đầu và dung tích bộ nhớ để lưu trữ các file dữ liệu nhỏ thường nhỏ hơn dung tích dùng để lưu trữ file dữ liệu lớn ban

đầu. Như vậy, nhờ phép nối chúng ta tiết kiệm được việc lưu trữ các bảng.

4.3. Các ví dụ

Bây giờ chúng ta đưa ra ví dụ minh hoạ việc sử dụng các phép toán xử lý bảng

Một cửa hàng tổng hợp mỗi ngày có bản tổng kết bán hàng như sau:

1. Bản tổng kết bán hàng một ngày từ những hoá đơn bán ra

Đơn vị tính 1000đ

Ngày tháng	Mã hàng	Tên hàng	Đơn vị tính	Số lượng
210397	M1	Radio	1 000	1
	M3	TV	4 000	2
	M6	Xe đạp	1 000	1
			Tổng giá tiền: 10 000	
			Đã thanh toán: 6 000	

ở đây 210397 là ngày bán: Ngày 21 tháng 03 năm 1997. Trên cơ sở của bản tổng kết này chúng ta xây

dựng một quan hệ (bảng) bán hàng như sau gồm 7 cột và cho các số liệu cụ thể.

Bán hàng

Ngày tháng	Mã hàng	Tên hàng	Đơn giá	Số lượng	Tổng	Thanh toán
210397	M1	Radio	1 000	1	10 000	6 000
210397	M3	TV	4 000	2	10 000	6 000
210397	M6	Xe đạp	1 000	1	10 000	6 000
220397	M2	Máy giặt	3 000	2	6 000	2 000
230397	M1	Radio	1 000	3	15 000	11 000
230397	M4	Video	5 000	2	15 000	11 000

230397	M9	Máy ảnh	2 000	1	15 000	11 000

Có thể thấy quan hệ bán hàng có khoá tối tiểu là Ngày tháng, Mã hàng.

2. Chúng ta sẽ tách từ bảng bán hàng thành 4 bảng sau:

khối lượng

Ngày tháng	Mã hàng	Số lượng
210397	M1	1
210397	M3	2
210397	M6	1
220397	M2	2
230397	M1	3
230397	M4	2
230397	M9	1

Khoá tối tiểu của quan hệ Khối lượng là Ngày tháng, Mã hàng

Doanh số

Ngày tháng	Tổng	Thanh toán

210397	10000	6000
220397	6000	2000
230397	15000	11000

Khoá tối tiểu là Ngày tháng

Hàng

Mã hàng	Tên hàng
M1	Radio
M2	Máy giặt
M3	TV
M4	Video
M6	Xe đạp
M9	Máy ảnh

Khoá tối tiểu là Mã hàng

Mặt hàng

Tên hàng	Đơn giá
Radio	1000
Máy giặt	3000

TV	4000
Video	5000
Xe đạp	1000
Máy ảnh	2000

Khoá tối tiểu là Tên hàng

3. Có thể thấy (Nếu không kể đến thứ tự của cột và hàng):

bán hàng = khối lượng >< doanh số >< Hàng ⋈ Mặt hàng

Không khó khăn lắm chúng ta thấy 4 quan hệ khối lượng, doanh số, Hàng, mặt hàng là 3NF, còn quan hệ bán hàng chưa được chuẩn hoá. Để thực hiện việc xử lí thông tin, chúng ta lưu trữ 4 quan hệ đã được chuẩn hóa, chứ không lưu trữ quan hệ bán hàng.

Như vậy nhờ phép nối chúng ta có thể hồi phục được quan hệ Bán Hàng

Bây giờ chúng ta xử dụng các phép toán xử lí bảng để tìm kiếm và in ra các thông tin sau

- Chúng ta muốn biết doanh số bán ra sau ngày 21 tháng 03 năm 1997 đó là

$\Pi_{\text{Ngày tháng,Tổng}}(\delta_{\text{Ngày tháng' > '210397'}}(\text{doanh số}))$ và in ra bảng sau:

Ngày tháng	Tổng
220397	6 000
230397	15 000

- Chúng ta muốn biết các tên hàng và số lượng đã bán trong ngày 21 tháng 03 năm 1997.

$\Pi_{\text{Tên hàng, số lượng}} (\delta_{\text{Ngày tháng='210397'}} (\text{Khối lượng}) \bowtie \text{Hàng})$

Tên hàng	Số lượng
Radio	1
TV	2
Xe đạp	1

- Tìm các ngày mà trong các ngày đó doanh số bán ra ít nhất là 10.000.000đ

$\Pi_{\text{Ngày tháng}} (\delta_{\text{Tổng}\geq \text{'10 000'}} (\text{doanh số}))$

Ngày tháng
210197
230397

- In ra các mã và tên hàng mà đơn giá của nó nhỏ hơn 3.000.000đ

$$\Pi_{\text{Tên hàng, Mã hàng}} (\delta_{\text{Đơn giá'<'3 000'}} (\text{Mặt hàng})).$$

Tên hàng	Mã hàng
Radio	M1
Xe đạp	M6
Máy ảnh	M9

- Cho các mã hàng và đơn giá của chúng trong ngày 23 tháng 03 năm 1997.

$$\Pi_{\text{Mã hàng}} (\delta_{\text{Ngày tháng = '230397'}} (\text{khối lượng})) >< \Pi_{\text{Mã hàng, Đơn giá}} (\text{Hàng} \bowtie \text{Mặt Hàng}).$$

Mã hàng	Đơn giá
M1	1000
M4	5000
M9	2000

- Tìm tên, đơn giá của mã hàng M1 và số lượng bán ra của mặt hàng này trong ngày 23 tháng 03 năm 1997

$\Pi_{\text{Tên hàng, Đơn giá, Số lượng}}(\Pi_{\text{Mã hàng, Số lượng}}(\delta_{\text{Ngày tháng = '230397' } \wedge \text{Mã hàng = M1}}(\text{khối lượng}) >< \text{hàng} \bowtie \text{Mặt hàng}))$.

Tên hàng	Đơn giá	Số lượng
Radio	1000	3

Ví dụ: Cho 2 quan hệ

r_1 =

Chuyến bay	Máy bay
83	727
83	747
84	727
84	747
109	707

r_2 =

Phi công	Máy bay

Tuấn	707
Tuấn	727
Thành	747
Thắng	727
Thắng	747

Chúng ta cần in ra một bảng chỉ ra các phi công có thể lái cho mỗi chuyến bay. Khi đó chúng ta chỉ cần thực hiện phép nối tự nhiên giữa r_1 và r_2.

$r_3 = r_1 \bowtie r_2$

Chuyến bay	Máy bay	Phi công
83	727	Tuấn
83	727	Thắng
83	747	Thành
83	747	Thắng

84	727	Tuấn
84	727	Thắng
84	747	Thành
84	747	Thắng
109	707	Tuấn

Đơn giản hơn ta thực hiện

$\Pi_{\text{Chuyến bay, Phi công}}(r_3)$

Chuyến bay	Phi công
83	Tuấn
83	Thắng
83	Thành
84	Tuấn
84	Thắng
84	Thành

109	Tuấn

Chương 5
Một số áp dụng mô hình dữ liệu trong các hệ quản trị CSDL hiện có

5.1. Mô tả chung

Chương này mô tả việc áp dụng các khái niệm của chương 2, 3 trong các hệ QTCSDL hiện có trên thị trường. Các khái niệm này bao gồm thực thể,

thuộc tính, khoá , quan hệ, phụ thuộc hàm, các dạng chuẩn.

5.2. Những khái niệm cơ bản

Trong phần này chúng tôi nêu lại một vài khái niệm đã được trình bày sơ bộ ở chương 2.

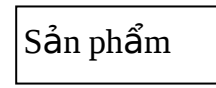
5.2.1. Thực thể

Thực thể là một hình ảnh tượng trưng cho một đối tượng cụ thể hay một khái niệm trừu tượng nhưng có mặt trong thế giới thực.

Ví dụ:

Dự án, con người, sản phẩm, ...

Thông thường khi xây dựng mô hình dữ liệu các thực thể được biểu diễn bằng những hình chữ nhật ví dụ như



5.2.2. Thuộc tính

Trong một hệ thông tin, ta cần lựa chọn một số tính chất đặc trưng để diễn tả một thực thể, các tính chất này được gọi là thuộc tính của thực thể được mô tả và đây cũng chính là các loại thông tin dữ liệu cần quản lí.

Ví dụ:

Họ tên, địa chỉ, ngày sinh của thực thể 'sinh viên'

Nhãn hiệu, giá của thực thể 'sản phẩm'

Giá trị các thuộc tính của một thực thể cho phép diễn tả một trường hợp cụ thể của thực thể, gọi là một thể hiện của thực thể đó .

Ví dụ:

('Trần Văn Sơn', '204 Triệu Việt Vương - Hà Nội', 12-5-1975) là một thể hiện của 'sinh viên'

('Máy vi tính ACER', 1349) là một thể hiện của 'sản phẩm'

Một thuộc tính là sơ cấp khi ta không cần phân tích nó thành nhiều thuộc tính khác, tuỳ theo nhu cầu xử lí trong hệ thông tin đối với một thực thể.

Thông thường một thực thể tương ứng với một bảng (hay một quan hệ của Codd).

Mỗi thực thể phải có ít nhất một thuộc tính mà mỗi giá trị của nó vừa đủ cho phép nhận diện một cách duy nhất một thể hiện của thực thể, gọi là thuộc tính nhận dạng hay là khoá. Có nhiều trường hợp chúng ta phải dùng một tập các thuộc tính để nhận diện thực thể. Khi một thực thể có nhiều khoá, người ta chọn một trong số đó làm khoá chính (khóa

tối tiểu). Giá trị của một khoá luôn luôn được xác định.

Ví dụ:

Số hoá đơn là thuộc tính nhận dạng của thực thể "Hoá đơn".

Không thể có hai hay nhiều hoá đơn có cùng số hoá đơn trong cùng một hệ thông tin.

Ví dụ:

Hoá Đơn
Số Hoá Đơn
Khách Hàng
Giá Tiền

5.2.3. Quan hệ

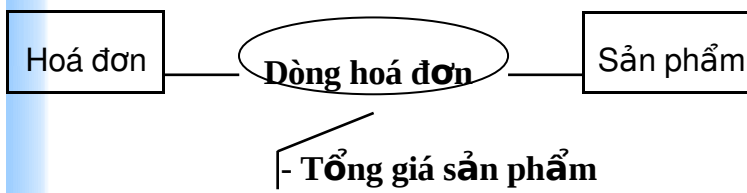
Khái niệm quan hệ ở mục này (khác với khái niệm quan hệ của Codd) được dùng để nhóm họp 2 hay nhiều thực thể với nhau nhằm biểu hiện một mối liên quan tồn tại trong thế giới thực giữa các thực thể này. Kích thước của một quan hệ là số thực thể đã cấu thành nên quan hệ, và có thể là một số

nguyên bất kỳ. Tuy vậy, trong thực tiễn, người ta luôn tìm cách tránh dùng đến những quan hệ có kích thước lớn hơn 3.

Trong một mô hình dữ liệu các quan hệ được biểu diễn bằng những hình tròn hoặc elipse. Trong một số trường hợp, mối quan hệ cũng có thể có những thuộc tính riêng.

Ví dụ:

Hoá đơn dùng để thanh toán một số sản phẩm bán ra. Mỗi dòng hoá đơn cho biết tổng giá của mỗi sản phẩm.Đây là một quan hệ có kích thước là 2, còn gọi là quan hệ nhị nguyên.



Người ta đưa ra khái niệm những vai trò khác nhau của cùng một thực thể để có thể biểu diễn mối quan hệ giữa thực thể này với chính nó. Vì loại quan hệ này ít dùng nên trong Giáo trình này chúng tôi không trình bày loại quan hệ đó.

5.2.4. Phân loại các quan hệ

Xét R là một tập quan hệ và E là một thực thể cấu thành của R, mỗi cặp (E,R) được biểu thị trên sơ đồ khái niệm dữ liệu bằng một đoạn thẳng. Với thực thể E, ta có thể xác định được:

- X là số tối thiểu các thể hiện tương ứng với E mà R có thể có trong thực tế.

Giá trị của X như vậy chỉ có thể bằng 0 hay 1.

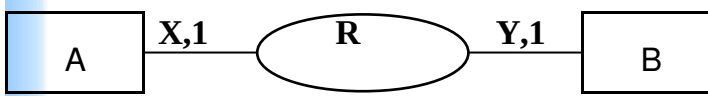
- Y là số tối đa các thể hiện tương ứng với E mà R có thể có trong thực tế.

Giá trị của Y có thể bằng 1 hay một số nguyên N > 1 .

Cặp số (X, Y) được định nghĩa là bản số của đoạn thẳng (E,R) và có thể lấy các giá trị sau: (0,1), (1,1), (0,N) hay (1,N), với N > 1.

Đối với loại quan hệ nhị nguyên R liên kết giữa hai thực thể A và B, ta phân thành ba loại quan hệ cơ bản sau:

- Quan hệ 1-1 (một - một): mỗi thể hiện của thực thể A được kết hợp với 0 hay 1 thể hiện của B và ngược lại.



X và Y có thể lấy các giá trị 0 và 1

Ví dụ:

Mỗi độc giả ở một thời điểm chỉ được đọc một quyển sách.

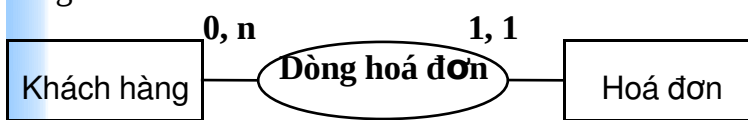
- Quan hệ 1 - N (một - nhiều): Mỗi thể hiện của thực thể A được kết hợp với 0,1 hay nhiều thể hiện của B và mỗi thể hiện của B được kết hợp với một thể hiện duy nhất của A. Đây là một loại quan hệ thông dụng và đơn giản nhất.

X có thể lấy các giá trị 0 và 1

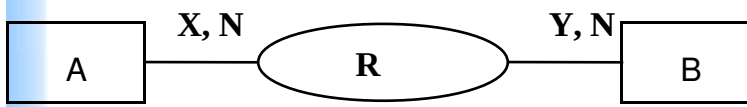
Ví dụ:

Một khách hàng có thể có nhiều hoá đơn

Một hoá đơn chỉ có thể mang tên một khách hàng.



- Quan hệ N - P (nhiều - nhiều): Mỗi thể hiện của một thực thể A được kết hợp với 0, 1 hay nhiều thể hiện của B và ngược lại, mỗi thể hiện của B được kết hợp 0, 1 hay nhiều thể hiện của A



X và Y có thể lấy các giá trị 0 hoặc 1

Ví dụ:

Một hoá đơn dùng để thanh toán một hay nhiều sản phẩm

Một sản phẩm có thể xuất hiện trong 0, 1 hay nhiều hoá đơn.

Thông thường quan hệ N - P chứa các thuộc tính. Chúng ta biến đổi loại quan hệ này thành thực thể và thực thể này cần được nhận dạng bởi một khoá chính.

5.3. Các dạng chuẩn trong các hệ QTCSDL hiện có

Trong mục này chúng tôi trình bày một số ý nghĩa của phụ thuộc hàm và mối liên hệ của nó với việc chuẩn hoá trong thực tiễn

5.3.1. Một số phụ thuộc hàm đặc biệt

Trên cơ sở của định nghĩa phụ thuộc hàm đã trình bày ở chương 2 chúng ta có thể thấy:

- Nếu B phụ thuộc hàm vào A (A→B), thì với mỗi giá trị của A tương ứng với một giá trị duy nhất của B. Hay nói cách khác, tồn tại một hàm (ánh xạ) từ tập hợp những giá trị của A đến tập hợp những giá trị của B.

Ví dụ: Trong một hoá đơn bao gồm các thuộc tính 'số hoá đơn', 'tên khách hàng', 'mã sản phẩm', 'tổng giá trị sản phẩm'....

Ta thấy 'Số hoá đơn' →'Tên khách hàng'

'Số hoá đơn', 'Mã sản phẩm' →'Tổng giá trị sản phẩm'

Chúng ta có thể mở rộng khái niệm phụ thuộc hàm khi cho phép A hoặc B là một thực thể hoặc là một quan hệ.

Ví dụ: Ta có hai thực thể 'Hoá đơn' và 'Khách hàng'

Khi đó: Thực thể 'Hoá đơn' →'Thuộc tính 'Tên khách hàng'

Thực thể 'Hoá đơn' →thực thể 'Khách hàng'

Thuộc tính 'Số hoá đơn' →thực thể 'Khách hàng'

Trong chương 3 ta trình bày phụ thuộc hàm hoàn toàn và phụ thuộc hàm trực tiếp. Hai loại phụ thuộc hàm này đóng vai trò quan trong các dạng chuẩn.

Các dạng chuẩn được đề ra với mục đích để đảm bảo tính nhất quán và tránh việc trùng lặp các thông tin. Trong mục này chúng ta sẽ quay trở lại với các dạng chuẩn. Các dạng chuẩn này có những biến đổi điều kiện ràng buộc đơn giản hơn so với các dạng chuẩn đã trình bày trong chương 3**.**

5.3.2. Dạng chuẩn 1

Chúng ta nói rằng một thực thể hay quan hệ ở dạng chuẩn 1 nếu tất cả giá trị các thuộc tính của nó là sơ cấp. Điều kiện ràng buộc giống như 1NF của chương 3. Định nghĩa của dạng chuẩn 1 mang tính mô tả. Thông thường giá trị các thuộc tính là các dãy kí tự hoặc là các số như trong FOXPRO, khi đó chúng ta cho các giá trị này là sơ cấp

Để minh họa ta đưa ra thực thể sau đã trình bày trong mục 4.3.

Bán hàng

Ngày tháng	Mã hàng	Tên hàng	Đơn giá	Số lượng	Tổng	Thanh toán
210397	M1	Radio	1000	1	10000	6 000

210397	M3	TV	4000	2	10000	6 000
210397	M6	Xe đạp	1000	1	10000	6 000
220397	M2	Máy giặt	3000	2	6 000	2 000
230397	M1	Radio	1000	3	15000	11 000
230397	M4	Video	5000	2	15000	11 000
230397	M9	Máy ảnh	2000	1	15000	11 000

Chúng ta chọn khoá chính (nó là một khoá tối tiểu) cho thực thể bán hàng là tập {Ngày tháng, Mã hàng}

5.3.3. Dạng chuẩn 2

Một thực thể hay quan hệ là 1NF được xem là dạng chuẩn 2 nếu tất cả các phụ thuộc hàm giữa khoá chính và các thuộc tính khác của nó đều là hoàn toàn .

Chú ý rằng định nghĩa dạng chuẩn 2 trong chương 2 chặt hơn vì điều kiện phụ thuộc hoàn toàn liên quan đến mọi khoá tối tiểu, chứ không chỉ liên

quan đến một khoá tối tiểu được chọn làm khoá chính.

Trong ví dụ trên, thực thể Bán hàng đã là 1NF, ta nhận thấy đối với khoá chính {số phiếu, mã vật tư} các thuộc tính Tổng và Thanh toán phụ thuộc hàm vào thuộc tính Ngày tháng, các thuộc tính Tên hàng, Đơn giá phụ thuộc hàm vào thuộc tính Mã hàng. Ngày tháng, Mã hàng là hai thuộc tính của khóa chính. Do đó dẫn đến trùng lặp dữ liệu. Thực thể Bán hàng không là 2NF. Để thoả dạng chuẩn 2NF, ta phải tách nó thành 3 thực thể riêng biệt:

hàng hoá

Mã hàng	Tên hàng	Đơn giá
M1	Radio	1 000
M2	Máy giặt	3 000
M3	TV	4 000
M4	Video	5 000
M6	Xe đạp	1 000
M9	Máy ảnh	2 000

Ta chọn khoá chính của thực thể hàng hoá là Mã hàng

khối lượng

Ngày tháng	Mã hàng	Số lượng
210397	M1	1
210397	M3	2
210397	M6	1
220397	M2	2
230397	M1	3
230397	M4	2
230397	M9	1

Ta chọn khoá chính (nó là khoá tối tiểu) cho thực thể Khối lượng là Ngày tháng, Mã hàng

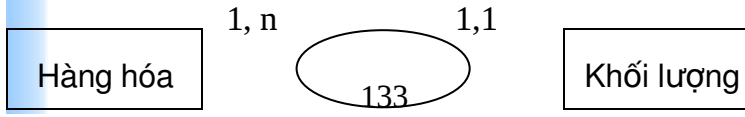
Doanh số

Ngày tháng	Tổng	Thanh toán
210397	10000	6000
220397	6000	2000
230397	15000	11000

Khoá chính là Ngày tháng

Có thể thấy thực thể Khối lượng →thực thể Hàng hoá

Ta có biểu diễn sau:



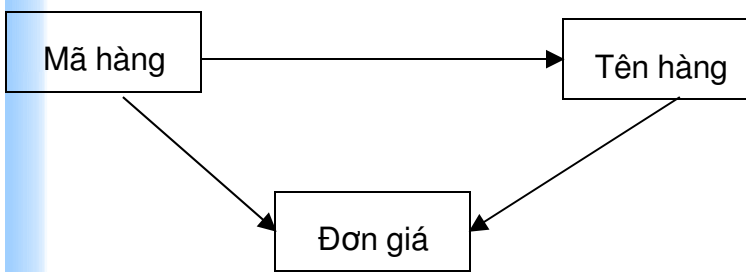
R

5.3.4. Dạng chuẩn 3 (3NF)

Một thực thể (hay quan hệ) đã là 2NF được xem là dạng chuẩn 3NF nếu tất cả các phụ thuộc hàm giữa khoá chính và các thuộc tính khác của nó đều là trực tiếp.

Hay nói cách khác, mọi thuộc tính không nằm trong khoá chính đều không phụ thuộc hàm vào một thuộc tính không phải là khóa chính. Ta có thể rút ra nhận xét: Một thực thể có nhiều khoá nhận dạng không thể thoả mãn dạng chuẩn 3NF. Mặt khác định nghĩa 3NF trong chương 2 chặt hơn vì điều kiện phụ thuộc hoàn toàn và phụ thuộc trực tiếp liên quan đến mọi khoá tối tiểu, chứ không chỉ liên quan đến một khoá tối tiểu được chọn làm khoá chính.

Trong thực thể Hàng hoá là 2NF ở trên, ta thấy trên đồ thị của các phụ thuộc hàm có hai con đường để đi từ 'Mã hàng' đến 'Đơn giá' hoặc đi qua thuộc tính 'Tên hàng'



Điều này chứng tỏ thực thể chưa là 3NF, dẫn đến trùng lặp đơn giá của tên hàng. Để là dạng 3NF, ta tách nó thành hai thực thể riêng biệt:

Hàng

Mã hàng	Tên hàng
M1	Radio
M2	Máy giặt
M3	TV
M4	Video
M6	Xe đạp
M9	Máy ảnh

Khoá chính là Mã hàng

Mặt hàng

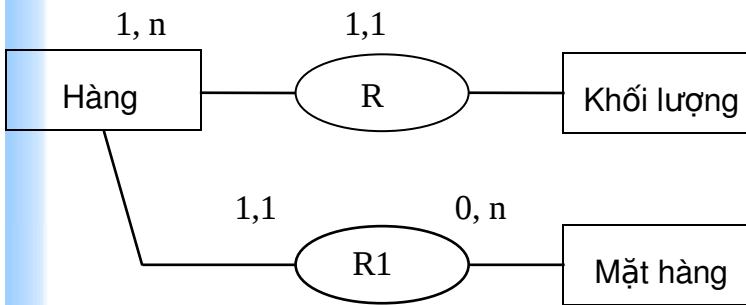
Tên hàng	Đơn giá
Radio	1000
Máy giặt	3000
TV	4000

Video	5000
Xe đạp	1000
Máy ảnh	2000

Khoá chính là Tên hàng

Có thể thấy thực thể Hàng → thực thể Mặt hàng

Tổng hợp với phần trên, ta có sơ đồ sau:



5.3.5. Dạng chuẩn của Boyce - Codd (BCNF)

Dạng chuẩn 3NF cho phép một thuộc tính thành phần của khoá chính phụ thuộc hàm vào một thuộc tính không phải là khoá

Ví dụ :

Lớp	Môn	Thầy
12	Toán	A
11	Toán	D
10	Toán	A
12	Địa	C
11	Địa	C
10	Địa	D

Thực thể này thoả dạng 3NF. Khoá chính của nó gồm các thuộc tính 'Lớp' và 'Môn'.

Nhưng do qui tắc 'Mỗi thầy chỉ dạy một môn', ta thấy có sự phụ thuộc hàm của Môn (Là một thành phần của khoá chính) vào Thầy (Là một thuộc tính bình thường):

'Thầy' $\rightarrow$'Môn'

Ta nói rằng thực thể thoả mãn dạng chuẩn Boyce-Codd (BCNF) khi tất cả các phụ thuộc hàm của nó đều thuộc dạng K $\rightarrow$ a, trong đó K là khoá chính và a là một thuộc tính bất kỳ.

Để thoả dạng BCNF, ta có thể tách thực thể trên thành hai thực thể riêng biệt như sau:

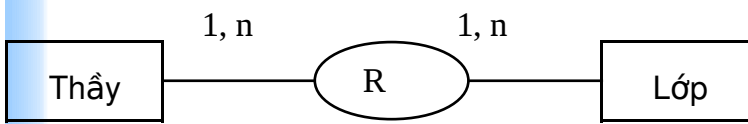
Thực thể 'Lớp':

Lớp
12
11
10

Thực thể 'Thầy':

Thầy	Môn
A	Toán
B	Toán
C	Sử Địa
D	Sử Địa

Chúng ta có biểu diễn sau:



5.3.6. Nhận xét về việc chuẩn hoá

Khi không có yêu cầu gì đặc biệt, người ta thường tìm cách chuẩn hoá mô hình dữ liệu nhằm tăng hiệu hiệu năng và giảm sơ xuất trong các giai đoạn phát triển hệ thông tin về sau.

Tuy vậy, việc chuẩn hoá không phải lúc nào cũng đạt đến mức tối đa. Thông thường chúng ta chuẩn hoá đến dạng chuẩn 2NF và 3NF.

Chương 6

Một số công đoạn xây dựng các dự án thiết kế tổng thể các hệ thống cơ sở dữ liệu hiện nay

6.1. Mô tả chung

Trong dự án thiết kế tổng thể người ta thường trình bày một số vấn đề cơ bản sau:

- Hiện trạng và tiềm lực CNTT của cơ quan chủ trì dự án.

- Hiện trạng về việc thu thập, lưu trữ và xử lí các dữ liệu liên quan hệ cơ sở dữ liệu cần xây dựng,

- Ước đoán khối lượng lưu trữ và trao đổi thông tin trong hệ cơ sở dữ liệu,

- Phân tích thiết kế hệ CSDL,

- Thiết kế hạ tầng kĩ thuật,

- Chuẩn hoá, bảo mật và an toàn thông tin,

- Tính pháp lí về quyền và nghĩa vụ trong việc thu thập, cập nhật, khai thác và bảo vệ thông tin trong hệ CSDL,

- Nội dung, đối tượng và kế hoạch triển khai công tác đào tạo,

- Tổng hợp và cân đối kinh phí

- Các biện pháp tổ chức thực hiện .

Một trong các phần quan trọng nhất của dự án là khâu phân tích thiết kế.

Thông thường để thực hiện việc phân tích thiết kế khi xây dựng dự án người ta sử dụng một số phương pháp phân tích thiết kế, ví dụ như phương pháp phân tích thiết kế có cấu trúc (Structured Analysis and Design), phương pháp GALASCI, phương pháp phân tích MCX, phương pháp phân tích MERISE (Methode pour Rassembler les Idees Sans Effort)...

Trong các phương pháp phân tích thiết kế trên MERISE có đặc tính là khi phân tích nó tách rời xử lí với dữ liệu, tổ chức theo nhiều mức, đi từ phân tích tổng thể đến chi tiết một cách tuần tự theo chiều tăng dần của mức độ phức tạp. MERISE được nhiều người xử dụng và đặc biệt tốt cho việc phân tích thiết kế cho các hệ thông tin lớn. Tại Việt Nam một số cơ quan đã được trang bị công cụ phân mềm MEGA tuân theo các chuẩn của phương pháp phân tích MERISE. Cho đến nay nhiều dự án "Thiết kế

tổng thể hệ thống cơ sở dữ liệu" đã được xây dựng trên cơ sở phương pháp phân tích MERISE.

Dựa trên phương pháp MERISE, khi phân tích thiết kế tổng thể, chúng ta tiến hành các công đoạn sau :

① Mô tả qui trình nghiệp vụ bằng lời:

Trong công đoạn này chúng ta mô tả đầy đủ, mạch lạc bằng lời qui trình nghiệp vụ của bài toán cần thiết kế. Công đoạn này thường phân biệt những yếu tố nghiệp vụ hay thay đổi với những yếu tố nghiệp vụ tương đối bất biến trong khoảng thời gian dài. Những yếu tố bất biến này thường được tập trung mô tả kĩ hơn. Trong công đoạn mô tả bằng lời này, chúng ta không chỉ mô tả những yếu tố nghiệp vụ hiện có mà còn mô tả các kiến nghị sửa đổi, các dự báo tương lai phù hợp với quá trình phát triển của hệ thống thông tin. Công đoạn này phân rã mỗi lĩnh vực nghiệp vụ thành các chức năng nghiệp vụ, và mỗi chức năng nghiệp vụ được phân rã thành các thao tác xử lí. Toàn bộ các yếu tố này đều được mô tả rõ ràng.

② Xây dựng các mô hình

Sau khi mô tả các qui trình nghiệp vụ một cách rõ ràng và mạch lạc, chúng ta tiến hành xây dựng các mô hình. Các mô hình này được phân chia theo 4 mức độ khái quát khác nhau. Đó là :

- Mức khái niệm: Mức này mô tả sự hoạt động của đơn vị theo một cấu trúc khái quát nhất. Trong mức này các chức năng hoạt động được mô tả độc lập với những bộ phận, vị trí hay nhân viên thực hiện chúng.

- Mức tổ chức: Mức này thể hiện các mục tiêu đã được khái niệm hóa lên thực tế của đơn vị trong đó có tính đến những điều kiện ràng buộc về mặt tổ chức.

- Mức lôgic: Mức này qui định các công cụ tin học mà các công cụ này được người dùng trong các thao tác xử lí.

- Mức vật lí: Mức này liên quan chặt chẽ với các trang thiết bị tin học cụ thể.

Trong mỗi mức chúng ta có 3 mô hình. Đó là mô hình trao đổi thông tin, mô hình xử lí và mô hình dữ liệu.

Cụ thể chúng ta có:

*Mô hình khái niệm trao đổi (MCC modele conceptuel de communication):

MCC phân rã lĩnh vực nghiệp vụ thành các chức năng nghiệp vụ. MCC mô tả sự trao đổi thông tin giữa các chức năng nghiệp vụ nằm trong bài toán và sự trao đổi thông tin giữa các lĩnh vực nghiệp vụ với các lĩnh vực nghiệp vụ khác và các đối tượng bên ngoài.

MCC cho ta một cái nhìn tổng thể về một lĩnh vực nghiệp vụ và các chức năng nghiệp vụ của nó cũng như các nhu cầu thông tin của nó.

*Mô hình khái niệm xử lí (MCT modele conceptuel de traitements):

MCT dùng để mô tả một lĩnh vực, qui trình, chức năng, thao tác.

MCT phân rã một chức năng nghiệp vụ thành các thao tác xử lí . Mỗi thao tác xử lí có thể được xem như một phép biến đổi thông tin. Một thao tác có thể có điều kiện khởi động là các sự kiện, các thông báo mà khi xuất hiện chúng thao tác bắt đầu được thực hiện . Trong quá trình thực hiện, thao tác cần phải truy nhập đến các thông tin đã được lưu giữ , biến đổi chúng và cập nhật lại theo một số qui luật tính toán và ràng buộc nhất định.

* Mô hình khái niệm dữ liệu (MCD modele conceptuel de donneees):

Việc mô tả dữ liệu trong mô hình MCD thông qua ngôn ngữ " Thực thể / Quan hệ " cùng với các thuộc tính của các thực thể và các quan hệ . Trên mô hình này, khóa chính, các thuộc tính chính được khai báo và lưu trữ trong hệ thống.

MCD được xây dựng cho một lĩnh vực nghiệp vụ nhằm xác định đầy đủ những dữ liệu đòi hỏi khi thực hiện các chức năng , trong đó đặc biệt là những dữ liệu cần thiết cho việc trao đổi .

* Mô hình tổ chức trao đổi (MOC):

MOC mô tả một lĩnh vực nghiệp vụ, đơn vị tổ chức.

Mô hình này mô tả các vị trí làm việc và việc luân chuyển thông tin trong đơn vị.

* Mô hình tổ chức xử lí (MOT):

MOT là mô hình tổ chức để thực hiện các thao tác của một chức năng nghiệp vụ đã được mô tả trong MCT. MOT thể hiện qui trình làm việc , trong đó nhấn mạnh tính tuần tự của các thao tác và nêu rõ những ràng buộc về thời điểm bắt đầu xử lí hay truyền thông tin. Một thao tác trong MCT có thể ứng với nhiều thao tác trong MOT và ngược lại một thao

tác trong MOT cũng có thể ứng với nhiều thao tác trong MOT tuỳ theo các hoàn cảnh cụ thể .

* Mô hình tổ chức dữ liệu (MOD):

MOD mô tả dữ liệu cần ghi nhớ trong từng địa điểm, cho từng vị trí thực hiện.

Trong khi MCD chỉ định nghĩa các khái niệm dữ liệu, thì MOC cụ thể hóa những điều kiện có thể xảy ra để một thực thể thuộc vào mô hình.

* Mô hình lôgic trao đổi (MLC):

MLC xác định sự trao đổi giữa người với người (thông qua các mẫu biểu), giữa người với máy (thông qua giao diện) và giữa các phần mềm với nhau.

* Mô hình lôgic xử lí (MLT): Thường để mô tả công cụ tin học.

* Mô hình lôgic dữ liệu (MLD):

Nhờ MLD, chúng ta có thể chuyển các MOD sang dạng quen thuộc cho các chuyên gia tin học. Thông thường, chúng ta chuyển từ ngôn ngữ thực thể - quan hệ sang dạng biểu báo.

Các mô hình vật lí trao đổi (MPC), mô hình vật lí xử lí (MPT), mô hình vật lí dữ liệu (MPD) gắn liền với các trang thiết bị tin học cụ thể.

Các mức và mô hình của MERISE có thể tóm tắt như sau:

Mức	Trao đổi	Chức năng	Dữ liệu
Khái niệm	MCC	MCT	MCD
Tổ chức	MOC	MOT	MOD
Lôgic	MLC	MLT	MLD
Vật lí	MPC	MPT	MPD

Quan trọng nhất trong các mô hình trên là các mô hình liên quan đến dữ liệu vì nó làm nền tảng cho việc xây dựng các mô hình khác.

Trong phạm vi Giáo trình này chúng tôi trình bày việc xây dựng mô hình khái niệm dữ liệu. Đó là mô hình dữ liệu thường có trong các dự án thiết kế tổng thể các hệ thống thông tin.

Quá trình xây dựng mô hình khái niệm dữ liệu có thể được chia thành các giai đoạn sau đây:

A. Khảo sát thực tế

- Thu thập và trình bày thông tin.

B. Thiết kế mô hình dữ liệu

- Kiểm kê các dữ liệu.
- Xác định các phụ thuộc hàm.
- Xây dựng mô hình khái niệm:
- Xác định tập hợp các khoá chính.
- Nhận diện các thực thể.
- Nhận diện các quan hệ.
- Phân bổ các thuộc tính còn lại.
- Vẽ sơ đồ khái niệm dữ liệu.

C. Kiểm soát và chuẩn hoá mô hình

6.2. Khảo sát thực tế

6.2. Khảo sát thực tế

Mục tiêu của giai đoạn này là qua quá trình quan sát, phỏng vấn, tìm hiểu và phân tích, chúng ta mô tả đầy đủ hiện trạng, các bài toán nghiệp vụ và các nhu cầu của người sử dụng mà hệ thống thông tin cần phải thoả mãn. Do đó, nó không chỉ giới hạn trong việc xây dựng mô hình dữ liệu mà còn là nguồn gốc các thông tin cần thiết cho việc xây dựng mô hình xử lí.

Để đạt mục tiêu này, ta cần thu thập và trình bày tất cả những thông tin dù thuộc phương diện dữ liệu

hay chức năng có thể hữu ích cho việc thiết kế hệ thông tin về sau. Các thông tin này cần được quan sát dưới dạng: tĩnh (dữ liệu sơ cấp, tài liệu, quảng cáo, đơn vị, vị trí làm việc..), dạng động (luồng luân chuyển các thông tin, tài liệu, chu kì, thời lượng), và dạng biến đổi của chúng (thủ tục, qui tắc quản lí, công thức tính toán,..).

Các thông tin thu thập được phải đầy đủ và chính xác vì chúng là nền tảng của hệ thông tin tương lai. Nhưng cũng không nên đi quá sâu vào chi tiết và phải biết gạn bỏ những thông tin không cần thiết, để không làm chệch hướng và gây khó khăn nặng nề cho việc phân tích thiết kế.

Công việc khảo sát không chỉ tập trung hoàn toàn vào giai đoạn đầu của quá trình phân tích thiết kế, mà có thể chạy dài trong suốt quá trình này để thu thập thêm thông tin, đào sâu vấn đề hay kiểm chứng một giả thiết cùng với người sử dụng khi hệ thông tin cần xây dựng quá lớn và phức tạp, ta nên chia nó thành nhiều tiểu hệ. Mỗi tiểu hệ có thể được khảo sát, phân tích hay thiết kế độc lập với nhau, trước khi được tập hợp lại .

Để chia một hệ thông tin thành nhiều tiểu hệ, người ta thường sử dụng một trong hai phương pháp tiếp cận sau đây:

- Phương pháp 1: Các tiểu hệ độc lập được định ra, dựa trên cơ sở những bài toán , chức năng nghiệp vụ chủ yếu của tổ chức. Đôi khi dựa trên một kế hoạch thực hiện theo thứ tự ưu tiên hay để thoả những yêu cầu về thời gian.

- Phương pháp 2: Một cuộc khảo sát tổng quát sơ khởi sẽ cho phép nhận diện những tiểu hệ tương đối độc lập với nhau.

Tiếp theo đó chúng ta tiến hành thu thập thông tin về hệ thống cần xây dựng

Công việc này chủ yếu là tham khảo tài liệu và tiếp xúc với những người sử dụng, đòi hỏi những khả năng như óc quan sát, kinh nghiệm, tài giao tiếp và ứng biến... Các phương pháp gò bó, cứng nhắc sẽ chẳng đem lại kết quả mong muốn. Do đó, phần này chỉ liệt kê và phân loại các thông tin có thể gặp được trong quá trình khảo sát, xem như để trợ giúp trí nhớ.

Trong quá trình thu thập thông tin thông thường ta tập hợp các dữ liệu sau

- Dữ liệu về hệ thông tin hiện tại bao gồm các dữ liệu liên quan trực tiếp đến hệ thông tin theo mọi dạng (tĩnh, động, biến đổi) và thông tin về môi trường làm việc

- Dữ liệu về hệ thống tương lai bao gồm các nhu cầu, mong muốn cho hệ thông tin của ta. Trong các dữ liệu này ta cần phân biệt những vấn đề đã được nhận thức và phát biểu rõ ràng, những vấn đề được nhận thức nhưng chưa được công nhận, những vấn đề còn chưa nhận thức và còn đang tranh luận.

Với mỗi loại dữ liệu cần thu thập đã nêu trên, nếu cần ta còn có thể tìm hiểu thêm về một số khía cạnh khác như: số lượng, độ chính xác cần có, ai là người có trách nhiệm...

Nhận xét:

- Nếu có thể, các cuộc phỏng vấn phải được tiến hành tuần tự theo cấu trúc phân cấp của tổ chức theo từng bộ phận, lĩnh vực, chức năng hay đi từ cấp lãnh đạo qua cấp quản lí đến những người thừa hành.

- Phải luôn nhớ sao chụp mẫu các hồ sơ, tài liệu, để có được cấu trúc chính xác các thông tin làm căn bản cho việc xây dựng mô hình dữ liệu sau này.

- Cần thiết nhất là luôn phân biệt những thông tin nói về hệ thông tin đang xây dựng với những thông tin thuộc về hệ này

Các thông tin thu thập, sau khi được tổng hợp sẽ được trình bày dưới hai dạng:

a. Mô tả các bài toán nghiệp vụ, các chức năng và tổ chức của cơ quan, các nhu cầu và mong muốn của người sử dụng một cách đầy đủ, nhưng ngắn gọn và mạch lạc, bằng một ngôn ngữ thông thường, gần gũi với mọi người.

b. Minh hoạ và hệ thống hoá phần trình bày trên bằng một ngôn ngữ hình thức, thường là dưới dạng phiếu mô tả, danh sách và đồ hoạ.

Trên thực tế có nhiều phương pháp trình bày thông dụng khác nhau,

6.3. Thiết kế mô hình dữ liệu

Thiết kế một mô hình khái niệm dữ liệu là liệt kê và định nghĩa chính xác những dữ liệu có liên quan đến các chức năng, hoạt động của một tổ chức. Sau đó ta nhóm chúng lại thành thực thể và quan hệ giữa các thực thể, rồi dùng một số qui ước đã định trước để trình bày dưới dạng mô hình khái niệm.

6.3.1. Kiểm kê dữ liệu

Danh sách này chủ yếu được rút từ những thông tin thu thập được trong giai đoạn khảo sát ban đầu; tài liệu thu thập được; nhu cầu, giải thích của người sử dụng.

Có thể phân biệt hai kiểu dữ liệu:

- Loại dữ liệu xuất hiện trực tiếp trên các tài liệu, màn hình, tập tin thu thập được.

- Loại dữ liệu không xuất hiện nhưng cần thiết để chứa kết quả trung gian, các thông tin đang chờ được xử lý, hay để tính toán các dữ liệu thuộc loại thứ nhất.

Một công cụ thông dụng, hữu ích cho giai đoạn này là bảng, dùng để phân tích các tài liệu thu thập và liệt kê ra danh sách các dữ liệu. Trong bảng này, ta trình bày mỗi cột là một tài liệu và mỗi hàng là một loại dữ liệu. Tại mỗi ô giao điểm, ta đánh dấu khi loại dữ liệu có xuất hiện trên tài liệu. Nên dùng hai loại dấu khác nhau để phân biệt loại dữ liệu trực tiếp với loại được tính toán thành.

Khi xây dựng bảng này, ta nên bắt đầu bằng những tài liệu cơ bản, quan trọng nhất và chỉ cần trình bày một loại tài liệu khi nó cho phép nhận dạng ít nhất một loại dữ liệu mới.

Ví dụ: Trong một công ty có:

- Kho hàng làm nhiệm vụ lưu giữ và quản lí hàng hoá và khi cần thì đề nghị mua thêm hàng

- Phòng đặt hàng có nhiệm vụ làm đơn đặt hàng gửi cho các công ty cung cấp

- Phòng kế toán lưu bản sao đơn đặt hàng để kiểm tra hàng.

Ta dùng 1 để đánh dấu dữ liệu trực tiếp và 2 cho dữ liệu được tính toán.

Tài liệu Loại dữ liệu	Phiếu đề nghị đặt hàng	Đơn đặt hàng	Phiếu giao hàng	Tệp tin về nhà cung cấp
Tên kho	1			
Địa chỉ kho	1			
Ngày đề nghị đặt hàng	1			
Số lượng hàng cần đặt	1			
Mã số sản phẩm	1	1	1	
Nhãn hiệu sản phẩm	1	1	1	
Mã số của công ty cung cấp		1	1	1
Tên công ty cung cấp		1	1	1

Địa chỉ công ty cung cấp		1	1	1
Đơn giá sản phẩm		1	1	1
Ngày đặt hàng		1		
Số lượng hàng đặt mua		1		
Tổng giá đơn đặt hàng		2		
Ngày giao hàng			1	
Số lượng hàng được giao			1	
Tổng giá hàng được giao			2	

Từ danh sách này, người ta cần kiểm tra bằng một số công tác thanh lọc như sau:

- Bỏ bớt các dữ liệu đồng nghĩa nhưng khác tên, chỉ giữ lại một.

Ví dụ: Mã số sản phẩm = danh mục mặt hàng

- Phân biệt các dữ liệu cùng tên nhưng khác nghĩa thành nhiều loại dữ liệu khác nhau.

Ví dụ: Giá bán của một cửa hiệu khác với giá bán của công ty sản xuất.

- Nhập chung các loại dữ liệu luôn luôn xuất hiện đồng thời với nhau trên mọi loại tài liệu thành một loại dữ liệu sơ cấp.

Ví dụ: số nhà và tên đường; ngày, tháng và năm sinh.

- Loại bỏ những loại dữ liệu có thể được xác định một cách duy nhất từ các loại dữ liệu khác, hoặc bằng công thức tính toán, hoặc do các qui luật của tổ chức

Ví dụ: Tổng giá đơn đặt hàng = Số lượng hàng* Đơn giá

Giả sử do qui luật của tổ chức, mọi đề nghị mua hàng phải được giải quyết nội trong ngày, ta suy ra:

Ngày đề nghị mua hàng = Ngày đặt hàng.

6.3.2. Định các phụ thuộc hàm giữa các dữ liệu

Từ danh sách dữ liệu đã được thanh lọc của hệ thông tin đạt được qua giai đoạn trên, ta phải định ra tất cả các phụ thuộc hàm có giữa chúng .

Cụ thể, ta phải tự đặt câu hỏi: Mỗi giá trị của một loại dữ liệu A có tương ứng với một giá trị duy nhất của loại dữ liệu B hay không?

Nếu câu trả lời là "Có" thì B phụ thuộc hàm vào A: $A \rightarrow B$

Ngoài các phụ thuộc hàm có vế trái A là một loại dữ liệu sơ cấp (gọi là phụ thuộc hàm sơ cấp) tương đối dễ xác định, ta còn phải nhận diện cả các phụ thuộc hàm trong đó vế trái A là một tập hợp của nhiều loại dữ liệu (gọi là phụ thuộc hàm đa phần). Trong trường hợp này, ta nên tự đặt câu hỏi: Cần ấn định giá trị của những loại dữ liệu nào để có thể suy ra một giá trị duy nhất của loại dữ liệu B? Các phụ thuộc hàm sẽ được trình bày dưới dạng một bảng như sau:

Loại dữ liệu Phụ thuộc hàm Phụ thuộc hàm

sơ cấp

đa phần

Mã số của công ty cung cấp .....

Tên công ty cung cấp.....
.....

Địa chỉ công ty cung cấp.....
.....

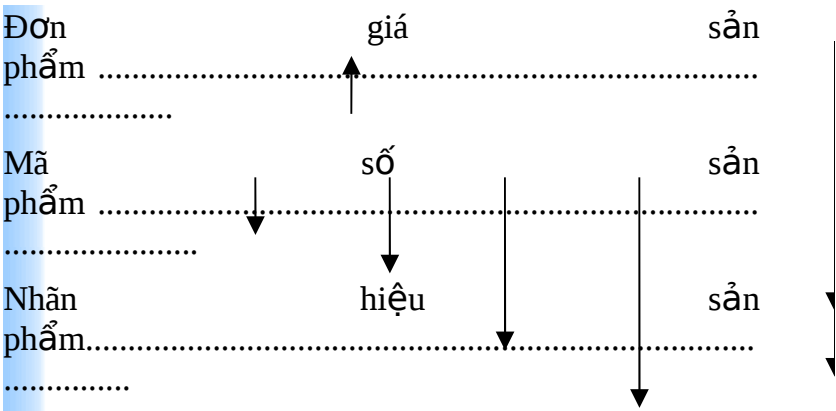
Tên kho .....
.....

Địa chỉ kho.....
.....

Ngày đề nghi đặt hàng .....
.....

Ngày đặt hàng .....
.....

Ngày giao hàng .....
.....



6.3.3. Xây dựng mô hình dữ liệu

Giai đoạn này bao gồm 5 động tác:

- Định tập hợp các khoá chính
- Nhận diện các thực thể
- Nhận diện các quan hệ

- Phân bổ các thuộc tính còn lại

- Vẽ sơ đồ khái niệm dữ liệu

6.3.3.1. Xác định tập hợp các khoá chính

Tập hợp K của những khoá chính là tập hợp tất cả những loại dữ liệu đóng vai trò nguồn (thuộc vế trái) trong ít nhất một phụ thuộc hàm.

Trong ví dụ trên ta có : K =

{'Mã số công ty cung cấp'

'Tên kho'

'Ngày đề nghi đặt hàng'

'Ngày đặt hàng'

'Ngày giao hàng'

'Mã số sản phẩm'}

6.3.3.2 Nhận diện các thực thể

Mỗi phần tử của tập hợp K sẽ là khoá chính của một thực thể

Trong ví dụ trên, ta nhận ra được 4 thực thể:

'Nhà cung cấp' với khoá chính là 'Mã số của công ty cung cấp'

'Kho' với khoá chính là 'Tên kho'

'Lịch' với khoá chính là 'Ngày'

(các thực thể 'Lịch đặt hàng', 'Lịch đề nghi mua hàng' và 'Lịch giao hàng' hoàn toàn tương đương với nhau nên được gộp thành một thực thể duy nhất là 'Lịch')

'Sản phẩm' với khoá chính là 'Mã số sản phẩm'

6.3.3.3 Nhận diện các quan hệ

Có 2 trường hợp :

a. Nếu gốc của một phụ thuộc hàm bao gồm ít nhất 2 phần tử thuộc tập hợp K thì nó tương ứng với một quan hệ N - P giữa các thực thể có khoá chính là các phần tử này.

Trong ví dụ trên, ta nhận ra được 4 quan hệ :

'Đơn giá của nhà cung cấp'

'Dòng đề nghị mua hàng'

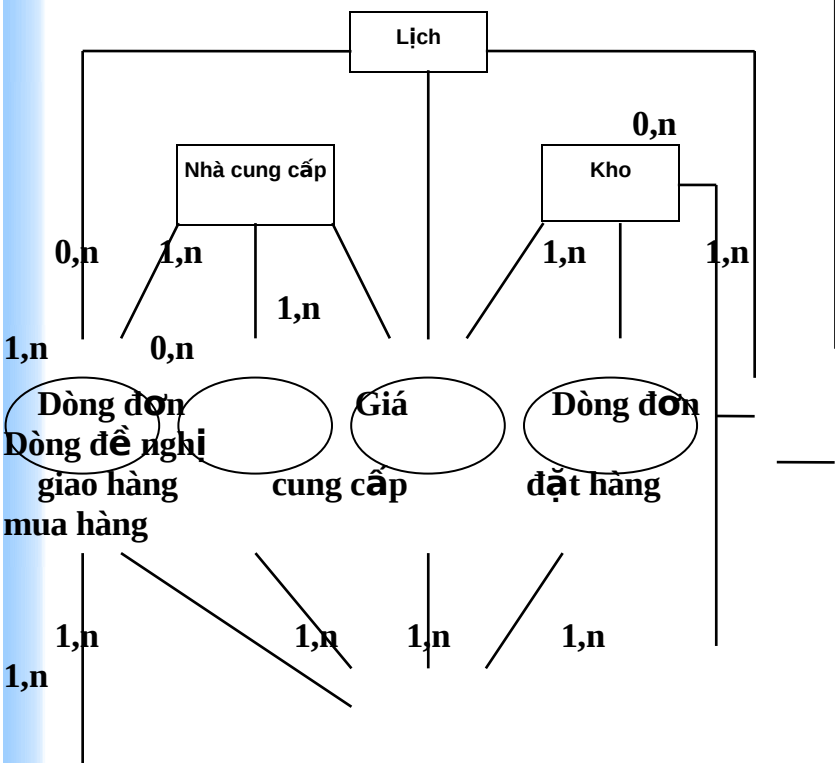
'Dòng đơn đặt hàng'

'Dòng phiếu giao hàng'

b. Một phụ thuộc hàm giữa 2 phần tử của tập hợp K xác định một quan hệ nhị nguyên kiểu 1-N giữa hai thực thể có khoá chính là các phần tử này.

6.3.3.5. Vẽ sơ đồ khái niệm dữ liệu

Từ các thực thể và quan hệ đã nhận diện, ta có thể vẽ lên một sơ đồ khái niệm dữ liệu sau:


Lịch
Nhà cung cấp
Kho
0,n
0,n
1,n
1,n
1,n
1,n
1,n
0,n
Dòng đơn giao hàng
Giá cung cấp
Dòng đơn đặt hàng
Dòng đề nghị mua hàng
1,n
1,n
1,n
1,n
1,n

6.3.4 Xây dựng mô hình dữ liệu bằng trực giác

Phương pháp phân tích hệ thống nêu trên là một công cụ hữu hiệu và chuẩn xác để xây dựng phần lớn các loại mô hình dữ liệu. Nhưng nếu áp dụng hoàn toàn trong một hệ thông tin cỡ lớn sẽ đòi hỏi nhiều thời gian và công sức. Trong thực tế, các thiết kế viên kinh nghiệm, sau khi đã nhận thức được vấn đề qua khảo sát thường chọn cách xây dựng trực tiếp một mô hình sơ khởi rồi đi thẳng vào giai đoạn sau để kiểm soát và chuẩn hoá mô hình .

Phương pháp trực giác này có ưu điểm là ít tốn thời gian và đôi khi tạo ra những mô hình đơn giản và gần thực tế hơn. Nhưng ngược lại, nó chứa nhiều rủi ro hơn.

6.4. Kiểm soát và chuẩn hoá mô hình

Để đơn giản hoá và đồng thời đảm bảo tính nhất quán của mô hình dữ liệu, ta cần kiểm soát lại mô hình vừa xây dựng bằng một số qui tắc thực tiễn sau đây:

6.4.1 Chuẩn hoá mô hình

Chú ý việc chuẩn hoá toàn bộ mô hình dữ liệu thành các dạng BCNF không phải là bắt buộc. Tuy

vậy các dạng 1FN, 2FN, 3NF nên luôn được thực hiện.

6.4.2 Tạo thêm một thực thể

Việc tạo thêm một thực thể là cần thiết khi có ít nhất một quan hệ đang được xử lý liên quan tới nó.

Việc tạo thêm một thực thể là hợp lí khi:

a) Thuộc tính sẽ được chọn làm khoá chính của thực thể này là một loại dữ liệu thông dụng trong hoạt động của tổ chức đang khảo sát.

b) Ngoài khoá chính này, quan hệ còn có chứa những thuộc tính khác .

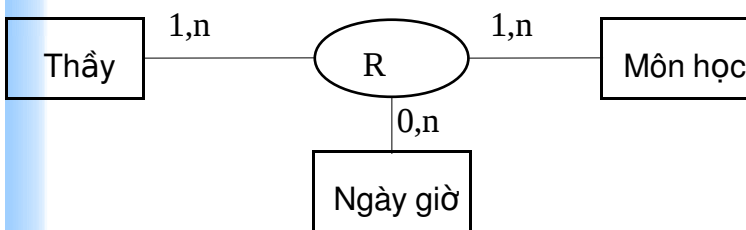
6.4.3. Biến một quan hệ thành thực thể

Một quan hệ có kích thước lớn hơn 3 nên được biến thành thực thể để đơn giản hoá.

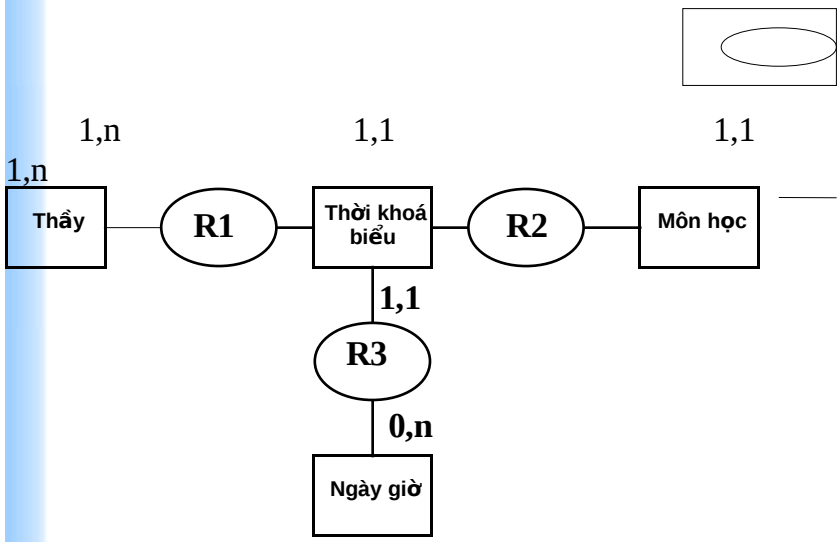
Có thể biến một quan hệ thành thực thể khi hội đủ các điều kiện sau:

- Quan hệ này có một khoá chính độc lập

- Quan hệ này tương ứng với một khái niệm quen thuộc, thông dụng trong hoạt động của tổ chức.

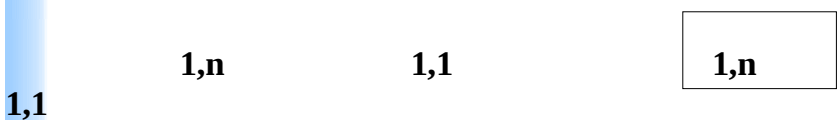


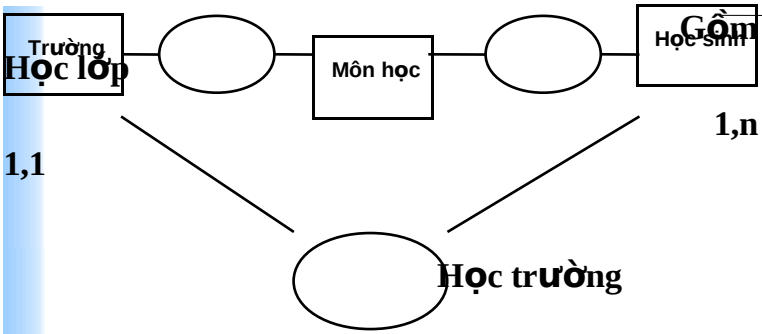
Quan hệ R được biến thành thực thể 'Thời khoá biểu':



6.4.4 Xoá một quan hệ

Một quan hệ 1 - N phải được loại bỏ khỏi mô hình dữ liệu nếu nó là tổng hợp của 2 hay nhiều quan hệ 1 - N khác.





Ta loại bỏ quan hệ Học trường

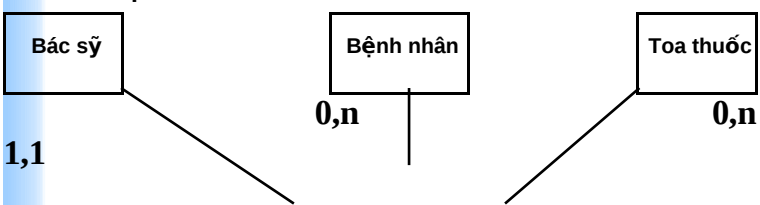
6.4.5 Phân tách một quan hệ phức tạp

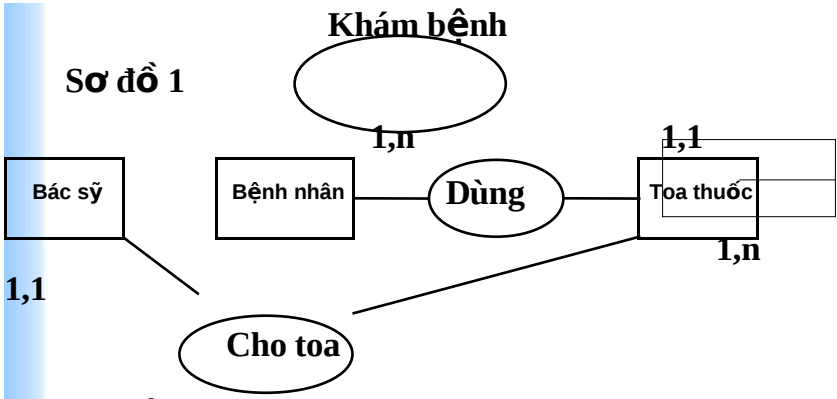
Xét một quan hệ có kích thước lớn hơn hoặc bằng 3. Quan hệ có thể được phân tách thành nhiều quan hệ khác với kích thước nhỏ hơn mà không mất thông tin nếu tồn tại ít nhất một phụ thuộc hàm giữa các thực thể cấu thành quan hệ.

6.4.5.1 Trường hợp phụ thuộc hàm ẩn

Trong trường hợp này, một trong các bản số của quan hệ bằng (1,1) hoặc (0,1). Điều này chứng tỏ sự tồn tại của một số phụ thuộc hàm ẩn.

Ví dụ:





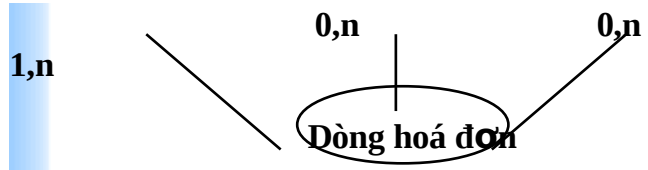
Sơ đồ 2

Chúng ta chọn sơ đồ 2

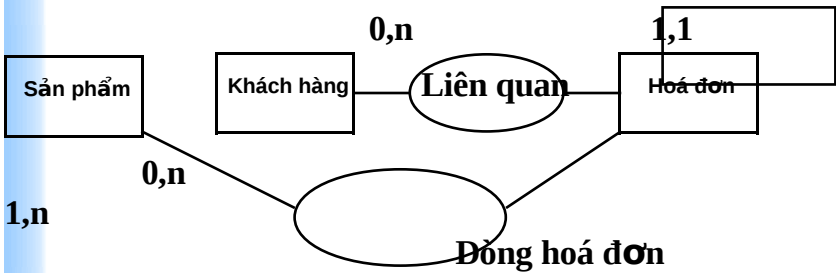
6.4.5.2. Trường hợp phụ thuộc hàm hiện

Cho một quan hệ R giữa 3 thực thể A, B, và C. Nếu tồn tại một hàm phụ thuộc A →B thì R có thể được phân thành quan hệ giữa A với B và giữa B với C.

Trong ví dụ sau đây, phụ thuộc hàm Hoá đơn → Khách hàng (mỗi hoá đơn chỉ liên quan đến một khách hàng duy nhất) cho phép ta đưa vào một quan hệ mới để diễn tả sự phụ thuộc này và đơn giản hoá mô hình.



Sơ đồ 1



Sơ đồ 2

Chúng ta chọn sơ đồ 2 đơn giản hơn.

Chương 7
Thuật toán và độ phức tạp

7.1. Khái niệm thuật toán

Nếu cho trước một bài toán, thì một cách giải bài toán được phân định ra thành một số hữu hạn bước, có kết thúc cuối cùng gọi là thuật toán.

Khi giải bài toán sẽ nẩy sinh ra các vấn đề sau:

- Độ phức tạp bài toán. Mỗi bài toán có độ phức tạp khác nhau.

- Một bài toán có nhiều thuật toán giải nó.

- Dùng nhiều ngôn ngữ lập trình để cài đặt phần mềm cho một thuật toán

- Có thể dùng nhiều cấu trúc dữ liệu cho một thuật toán.

Một số sai sót cơ bản khi giải bài toán:

- Hiểu sai bài toán.

- Tìm sai thuật toán.

- Do không hiểu ngôn ngữ lập trình nên có nhầm lẫn.

- Bản thân dữ liệu quét không hết trường hợp.

- Yêu cầu giải quyết bài toán.

- Phần mềm dễ sử dụng .

- Tốc độ tính toán nhanh .

- Bộ nhớ phù hợp .

Trong đó tính dễ sử dụng là một yêu cầu cơ bản nhất.

7.2. Độ phức tạp thuật toán

Khái niệm thuật toán chính xác liên quan đến các khái niệm máy Turing, hàm đệ qui, thuật toán Marcop, ngôn ngữ hình thức của N.Chomsky. Những khái niệm này không nằm trong khuôn khổ Giáo trình này. Chúng tôi trình bày một khái niệm quan trọng liên quan trực tiếp đến thuật toán. Đó là độ phức tạp thuật toán. Nhờ có khái niệm này chúng ta có thể đánh giá và so sánh được các thuật toán với nhau. Hay nói một cách khác, chúng ta có thể có công cụ đo, để lựa chọn một thuật toán tốt cho lời giải bài toán cần giải quyết. Thông thường chúng ta có hai loại đánh giá: Một là độ phức tạp về thời gian tính của thuật toán, hai là độ phức tạp về phạm vi bộ nhớ dùng cho thuật toán. Đối với một thuật toán, thời gian tính và phạm vi bộ nhớ cần dùng thường mâu thuẫn nhau. Có nghĩa là, nếu thời gian tính của thuật toán là ngắn thì thông thường phạm vi bộ nhớ dùng cho thuật toán đó lại lớn. Mà chúng ta lại muốn chọn một thuật toán thời gian tính thì ngắn và bộ nhớ dùng cũng nhỏ. Như vậy, trong từng trường hợp cụ thể, chúng ta sẽ quyết định chọn lựa thuật toán nào. Trong phạm vi Giáo trình này chúng ta chỉ trình bày về độ phức tạp thời gian tính. Đó là độ phức tạp thường được đề cập nhiều nhất. Đồng thời,

trong phạm vi giới hạn của giáo trình, chúng ta cũng chỉ trình bày độ phức tạp của thuật toán theo góc độ tin học.

Giả sử T là thuật toán giải quyết bài toán A. Chúng ta gọi T(n) là độ phức tạp thời gian của thuật toán T. Thông thường T(n) được biểu diễn dưới dạng sau: T(n) = O(g(n)). Trong đó hàm g(n) là cấp của T(n); n là độ dài thông tin đưa vào.

Ví dụ: $T(n) = O(n^2)$

Chúng ta hiểu f(n) = O(g(n)), nếu ∃ hằng số C và số nguyên n_0.

Sao cho:

$$\forall n \geq n_0 \text{ ta luôn có: } f(n) \leq Cg(n)$$

(Nói cách khác là g(n) là hàm chặn trên của f(n) từ một chỉ số nào đó trở đi).

Rõ ràng, trong quá trình đánh giá thuật toán, nếu có g(n) nhỏ nhất thì đó là sự đánh giá chính xác nhất.

Có thể thấy rằng, bài toán tìm g(n) nhỏ nhất khá phức tạp.

Bây giờ, chúng ta đưa ra độ phức tạp thời gian là hàm nhiều biến. Giả sử $T(n_1,... , n_k)$ là độ phức tạp thời gian của thuật toán T và $T(n_1,... , n_k) = O(g(n_1,....n_k))$. Khi đó chúng ta hiểu rằng tồn

tại các số C , $n_{01}...n_{0k}$ sao cho với mọi $n_1 \geq n_{01}$, ..., $n_k \geq n_{0k}$

$$T(n_1,...,n_k) \leq Cg(n_1,..., n_k)$$

Ví dụ: Đầu vào là R = {a_1,....., a_n},

$$r = \{h_1,..., h_m\}$$

Chúng ta có $T(n,m) = O(g(n,m))$

Trong trường hợp có nhiều đối số thì phức tạp thời gian được tính theo đối số có giá trị lớn nhất.

Ví dụ: $T(n,m) = O(n^2+2^m)$**. Khi đó độ phức tạp thời gian của thuật toán T là hàm số mũ.**

Việc đánh giá như trên gọi là độ phức tạp thời gian tồi nhất.

Trong thực tế có nhiều cách đánh giá độ phức tạp thời gian. Ví dụ như độ phức tạp thời gian trung bình. Độ phức tạp này gắn với nhiều độ đo khác nhau (độ đo xác suất). Giáo trình này đánh giá độ phức tạp thời gian theo cách tồi nhất (tìm g(n) chặn trên).

- Giả sử một độ phức tạp thuật toán chia làm nhiều đoạn, mỗi đoạn có độ phức tạp tương ứng là $T_1(n),...,T_q(n)$.

Khi đó chúng ta đặt $T(n) = O(\max(T_1(n),..., T_q(n))$.

- Nếu T_i là hàm nhiều biến: $T_1(n_1,...,n_m)$, ..., $T_q(n_1,...,n_m)$, thì lúc đó $T_1(n_1,..., n_m) = O(max(T_1(n_1,..., n_m),..., T_q(n_1,..., n_m))$.

- Giả sử T là thuật toán giải quyết bài toán A.

Ta chia hai khúc T_1 và T_2 và T_2 lồng trong T_1. Khi đó $T(n) = O(T_1(n). T_2(n))$.

Ví dụ: x:= 3;

x:= x + 1.

Dễ thấy độ phức tạp T(n) = O(c) (hay O(1)).

Ví dụ: x:= 3;

For i:= 1 to n do x:= x+1

Thì: T(n) = O(cn).

Ví dụ: For i: = 1 to n do

For j: = 1 to n do x:= x+1

$T(n) = O(c.n.n) = O(c.n^2)$

Trong trường hợp thuật toán có các đoạn lồng thắt vào nhau thì độ phức tạp là tích (Thể hiện bài toán có những toán tử lặp chu trình).

- Trong thuật toán chia thành nhiều đoạn. Có đoạn lồng thắt của đoạn khác: tính tích, có đoạn rời rạc: tính max.

Thông thường người ta chia các bài toán thành ba lớp. Đó là lớp bài toán được giải quyết bằng một thuật toán có độ phức tạp là hàm mũ, lớp bài toán NP - đầy đủ và lớp bài toán được giải quyết bằng một thuật toán có độ phức tạp là hàm đa thức.

- Đối với lớp bài toán được giải bằng thuật toán là hàm mũ và lớp bài toán NP - đầy đủ (Thường gọi là các bài toán không khả thi) thực tế trong tin học các bài toán này không có khả năng thực hiện vì thời gian tính quá lớn. Khi đó, chúng ta phải tách bài toán thành các dạng riêng biệt, và cố gắng đưa nó về lớp bài toán có độ phức tạp là hàm đa thức.

- Đối với các bài toán được giải bằng thuật toán có độ phức tạp là hàm đa thức, chúng ta cố gắng giảm số mũ k xuống (gần sát tuyến tính (mũ 1)).

Để hạ k (tăng tốc độ) thông thường người ta dùng cấu trúc dữ liệu, sử dụng ngôn ngữ gần ngôn ngữ máy.

Thông thường người ta coi các thông số vào (input) bình đẳng với nhau.

Tài liệu tham khảo

[1] Armstrong W.W. Dependency Structures of Database Relationships. Information Processing 74, Holland publ. Co. (1974), 580-583.

[2] Beeri C. Bernstein P.A. Computational problems related to the design of normal form relational schemas. ACM trans on Database Syst. 4.1 (1979), 30-59

[3] Beeri C. Dowd M., Fagin R.. Staman R. On the structure of Armstrong relations for Functional Dependencies . J.ACM 31,1 (1984), 30-46.

[4] Codd E. F. A relational model for large shared data banks. Communications ACM 13 (1970), 377-387.

[5] Demetrovics J., Logical and structural Investigation of Relational Datamodel. MTA - SZTAKI Tanulmanyok, Budapest, 114 (1980), 1-97.

[6] Demetrovics J., Libkin, L. Functional dependencies in relational databases : A Lattice point of view. Discrete Aplied Mathematics 40 (1992), 155-185.

[7] Demetrovics J., Thi V.D. Some results about functional dependencies. Acta Cybernetica 8,3 (1988), 273-278.

[8] Demetrovics J., Thi V.D. Relations and minimal keys. Acta Cybernetica 8,3 (1988), 279-285.

[9] Demetrovics J., Thi V.D. On keys in the Relational Datamodel. Inform. Process Cybern. EIK 24, 10 (1988), 515 - 519

[10] Demetrovics J., Thi V.D. Algorithm for generating Armstrong relations and inferring functional dependencies in the relational datamodel. Computers and Mathematics with Applications. Great Britain. 26,4(1993), 43 - 55.

[11] Demetrovics J., Thi V.D. Some problems concerning Keys for relation Schemes and Relationals in the Relational Datamodel. Information Processing Letters. North Holland 46,4(1993),179-183

[12] Demetrovics J., Thi. V.D. Some Computational Problems Related to the functional Dependency in the Relational Datamodel. Acta Scientiarum Mathematicarum 57, 1 - 4 (1993), 627 - 628.

[13] Demetrovics J., Thi V.D. Armstrong Relation, Functional Dependencies and Strong Dependencies. Comput. and AI. (submited for publication).

[14] Demetrovics J., Thi V.D. Keys, antikeys and prime attributes. Ann. Univ. Scien. Budapest Sect. Comput. 8 (1987), 37 - 54.

[15] Demetrovics J., Thi V.D. On the Time Complexity of Algorithms Related to Boyce-Codd Normal Forms. J. Serdica, the Bulgarian Academy of Sciencies, No.19 (1993), 134 - 144.

[16] Demetrovics J., Thi V.D. Generating Armstrong Relations for Relation schemes and Inferring Functional Dependencies from Relations.

International Jounal on Information Theories and Applications, ITA-93, 1, 4 (1993), 3 - 12.

[17] Demetrovics J., Thi V.D. Some problems concerning Armstrong relations of dual schemes and

relation schemes in the relational datamodel. Acta Cybernetica 11, 1-2 (1993), 35 - 47.

[18] Demetrovics J., Thi V.D. Normal Forms and Minimal Keys in the Relational Datamodel. Acta Cybernetica Vol. 11,3 (1994), 205 - 215.

[19] Demetrovics J., Thi, V.D. Some results about normal forms for functional dependency in the relational datamodel. Discrete Aplied Mathematics 69 (1996), 61 - 74.

[20] Garey M.R., Johnson D.S Computers and Intractability: A Guide to theory of NP - Completeness. Bell Laboratories. W.H Freeman and Company. San Francisco 1979.

[21] Gottlob G. Libkin L. Investigations on Armstrong relations dependency inference, and excluded functional dependencies. Acta Cybernetica Hungary IX/4 (1990), 385 - 402.

[22] Jou J.H, Fischer P.C. The complexity of recognizing 3NF relation schemes . IPL 14 (1982), 187 - 190.

[23] * Libkin L. Direct product decompositions of lattices, closures and relation schemes. Discrete Mathematics, North-Holland, 112 (1993), 119-138.

[24] Lucchesi C.L., Osborn S.L. Candidate keys for relations. J. Comput. Syst. Scien 17,2 (1978), 270 - 279

[25] Maier D. Minimum covers in the relational database model. JACM 27,4 (1980), 664 - 674.

[26] * Mannila H., Raiha K.J. Algorithms for inferring functional dependencies from relations. Data and Knowledge Engineering, North - Holland, V. 12, No. 1 (1994), 83 - 99.

[27] * Mannila H., Raiha K.J. On the complexity of inferring functional dependencies. Discrete Applied Mathematics, North - Holland, 40 (1992), 237 - 243.

[28] * Thalheim B. The number of keys in relational and nested relational databases. Discrete Applied Mathematics, North - Holland, 40 (1992), 265 - 282.

[29] Thi V.D. Investigation on Combinatorial Characterizations Related to Functional Dependency in the Relational Datamodel. MTA-SZTAKI Tanulmanyok. Budapest, 191 (1986), 1 - 157. Ph.D. dissertation.

[30] Thi V.D. Minimal keys and Antikeys. Acta Cybernetica 7.4 (1986), 361 - 371

[31] Thi V.D. Minimal keys and Antikeys. Acta Cybernetica 7, 4 (1986), 361 - 371.

[32] Thi V.D. Strong dependencies and s-semilattices. Acta Cybernetica 7, 2 (1987), 175 - 202.

[33] Thi V.D. Logical dependencies and irredundant relations. Computers and Artificial Intelligence 7 (1988), 165 - 184.

[34] Thi V.D., Anh N.K. Weak dependencies in the relational datamodel. Acta Cybernetica 10, 1-2 (1991), 93 - 100.

[35] Thi V.D., Thanh L.T. Some remark on Functional Dependencies in the relational Datamodel J. Acta Cybernetica, Hungary Vol. 11, 4 (1994), 345 - 352.

[36] Thi V.D. On the equivalent descriptions of family of functional dependencies in the relational datamodel. J. Computer Science and Cybernetic, Hanoi Vietnam, Vol 11, 4 (1995), 40 - 50.

[37] Thi V.D. Some Computational problems related to normal forms. J. Computer Science and Cybernetic, Hanoi Vietnam, Vol 13, 1 (1997), 53 - 65.

[38] Thi V.D. On the nonkeys. J. Computer Science and Cybernetic, Hanoi Vietnam, Vol 13, 1 (1997), 11 - 15.

[39] Thi V.D. Some results about hypergraph. J. Computer Science and Cybernetic, Hanoi Vietnam, Vol 13, 2 (1997), 8 - 15.

[40] Tsou D.M., Fischer P.C. Decomposition of a relation scheme into Boyce-Codd normal form. SIGACT NEWS 14 (1982), 23 - 29.

[41] Ullman J.D. Principles of database and knowledgw base systems. Computer Science Press, Second Edition (1992).

[42] Yannakakis M., Paradimitriou C. Algebraic dependencies. J. Comp. Syst. Scien. 25 (1982), 2 - 41.

[43] Yu C.T., Johnson D.T. On the complexity of finding the set of candidate keys for a given set of functional dependencies. IPL 5, 4 (1976), 100 - 101.

[44] Zaniolo C. Analysis and design of relational schemata for database systems. Ph. D. UCLA (1976).

[45] Collongues A., Hugues J., Laroche B. MERISE - Phương pháp thiết kế hệ thống thông tin tin học hoá phục vụ quản lí doanh nghiệp (Bản dịch). Nhà xuất bản Khoa học kĩ thuật, 1994.

[46] Hồ Sĩ Khoa. Các phương pháp xây dựng các mô hình khái niệm dữ liệu

[47] Các tài liệu hướng dẫn sử dụng hệ MEGA

[48] Demetrovics J., Denev. , Pavlov R. Cơ sở toán của khoa học tính toán. Hungary, 1985.

Mục lục

Trang

3.2. Dạng chuẩn 2 (2NF)
3.3. Dạng chuẩn 3 (3NF)
3.4. Dạng chuẩn Boyce - Codd (BCNF)
3.5. Các thuật toán liên quan
3.6. Dạng chuẩn của các hệ khoá
3.7. Ví dụ

Chương 4
Các phép toán xử lí bảng

4.1. Các phép toán cơ bản
4.2. Các phép toán khác
4.3. Các ví dụ

Chương 5
Một số áp dụng mô hình dữ liệu trong các hệ quản trị cơ sở dữ liệu (QTCSDL) hiện có

5.1. Mô tả chung
5.2. Những khái niệm cơ bản
5.3. Mối quan hệ giữa các thực thể
5.4. Các dạng chuẩn trong các hệ QTCSDL hiện có

Chương 6

Một số công đoạn xây dựng
các dự án thiết kế tổng thể các hệ thống
cơ sở dữ liệu hiện nay

6.1. Khảo sát thông tin
6.2. Thiết kế mô hình dữ liệu
6.3. Kiểm soát và chuẩn hoá mô hình

Chương 7
Thuật toán và độ phức tạp

7.1. Khái niệm thuật toán
7.2. Độ phức tạp thuật toán

Tài liệu tham khảo