

THIẾT KẾ HỆ CSDL QUAN HỆ - CHUẨN HOÁ LƯỢC ĐỒ QUAN HỆ

Nguyễn Đình Hóa

dinhhoa@gmail.com

094-280-7711

B4. Chuẩn hoá lược đồ quan hệ

Chuẩn hoá: đưa các lược đồ quan hệ về một dạng chuẩn theo yêu cầu:

- Định nghĩa về các dạng chuẩn
- Các quy tắc đưa lược đồ quan hệ về các dạng chuẩn.

Khái niệm về chuẩn hóa

- **Chuẩn hóa** là một kỹ thuật tạo ra một tập các quan hệ với các thuộc tính từ các yêu cầu cho trước về dữ liệu cần mô hình hóa của tổ chức.
- Việc chuẩn hóa thường được thực hiện như một chuỗi các kiểm tra trên một quan hệ để xác định xem nó có thỏa mãn hay vi phạm các yêu cầu của một dạng chuẩn cho trước nào đó hay không.
- **Quá trình chuẩn hóa** là một phương pháp chính thống để xác định các quan hệ dựa trên khóa chính, khóa dự bị và các phụ thuộc hàm giữa các thuộc tính.
- **Chuẩn hóa** giúp người thiết kế kiểm tra và chuyển các quan hệ về một dạng chuẩn hóa cụ thể nào đó nhằm ngăn chặn các dị thường khi thực hiện các thao tác cập nhật thông tin.

Mục đích của chuẩn hoá lược đồ

- Mục đích của việc thiết kế CSDL quan hệ: Xây dựng các lược đồ quan hệ sao cho tối thiểu hóa sự dư thừa dữ liệu => giảm không gian lưu trữ và tránh dị thường thông tin khi cập nhật dữ liệu.

staffbranch

<u>staff#</u>	sname	position	salary	branch#	baddress
SL21	Kristy	manager	30000	B005	22 Deer Road
SG37	Debi	assistant	12000	B003	163 Main Street
SG14	Alan	supervisor	18000	B003	163 Main Street
SA9	Traci	assistant	12000	B007	375 Fox Avenue
SG5	David	manager	24000	B003	163 Main Street
SL41	Anna	assistant	10000	B005	22 Deer Road

Mục đích của chuẩn hoá lược đồ

- Phân tách staffbranch thành 2 lược đồ Staff và Branch sẽ giúp tránh dữ thừa dữ liệu và tránh tạo ra các dị thường trong thao tác dữ liệu

staff

<u>staff#</u>	sname	position	salary	branch#
SL21	Kristy	manager	30000	B005
SG37	Debi	assistant	12000	B003
SG14	Alan	supervisor	18000	B003
SA9	Traci	assistant	12000	B007
SG5	David	manager	24000	B003
SL41	Anna	assistant	10000	B005

branch

<u>branch#</u>	baddress
B005	22 Deer Road
B003	163 Main Street
B007	375 Fox Avenue

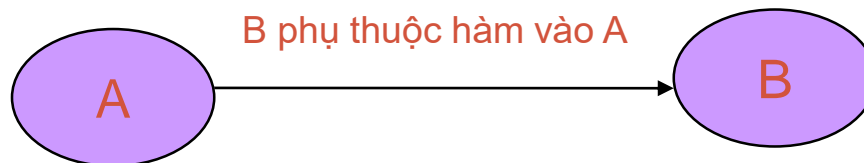
Các công cụ để chuẩn hoá lược đồ

- Phụ thuộc hàm
- Các thuộc tính khoá và các khái niệm về khoá
- Định nghĩa của các dạng chuẩn

Phụ thuộc hàm

- **Một phụ thuộc hàm** thể hiện ngữ nghĩa của các thuộc tính trong một quan hệ: một thuộc tính có quan hệ với thuộc tính khác như thế nào và xác định các phụ thuộc hàm giữa các thuộc tính đó. Phụ thuộc hàm là ràng buộc giữa các thuộc tính.

VD: A, B là các thuộc tính hoặc tập các thuộc tính trong quan hệ R. B phụ thuộc hàm vào A (ký hiệu: $A \rightarrow B$) nếu mỗi giá trị của A liên hệ tới duy nhất một giá trị của B.



Phụ thuộc hàm

- ***Đối tượng xác định của một phụ thuộc hàm*** là thuộc tính hoặc một nhóm các thuộc tính nằm bên trái của mũi tên trong một phụ thuộc hàm.
- ***Đối tượng (thuộc tính) hệ quả của phụ thuộc hàm*** là một thuộc tính hoặc nhóm thuộc tính nằm phía bên phải mũi tên trong phụ thuộc hàm.
- Với mục đích chuẩn hóa: chỉ quan tâm tới việc xác định các phụ thuộc hàm giữa các thuộc tính của quan hệ 1:1.

Phụ thuộc hàm

- Một phụ thuộc hàm là một đặc điểm chung của một lược đồ quan hệ chứ không phải là một đặc điểm riêng của một thể hiện cụ thể của lược đồ đó.
- Phụ thuộc hàm là bất biến theo thời gian, nghĩa là nó thỏa mãn tất cả các thể hiện có thể của quan hệ.

=> Thể hiện các loại ràng buộc toàn vẹn cần xác định.

- ***Phụ thuộc hàm hiển nhiên đúng***: khi và chỉ khi vế phải của phụ thuộc hàm là một tập con của vế trái.

=> Trong phạm vi chuẩn hóa, các phụ thuộc hàm này được bỏ qua.

Ví dụ

Xây dựng lược đồ ER cùng các lược đồ quan hệ và các phụ thuộc hàm cho kịch bản sau:

CSDL quản lý dạy học tại Multicampus gồm các thông tin về học viên (HV), giảng viên (GV), môn học, người quản lý (QL).

- Giảng viên dạy các môn học cho các Học viên. Quá trình dạy và học của mỗi GV và HV trên mỗi môn học phải được quản lý bởi một QL. Mỗi QL có thể quản lý nhiều quá trình dạy và học này.
- Mỗi GV dạy một môn học. Mỗi môn học có thể được dạy bởi nhiều GV và được học bởi nhiều HV. Mỗi HV học nhiều môn học
- Mỗi tập thực thể cần có ít nhất 4 thuộc tính.

Các luật suy diễn cho các phụ thuộc hàm

- IR1: Luật phản xạ: nếu $X \supseteq Y$, thì $X \rightarrow Y$
- IR2: Luật tăng trưởng: nếu $X \rightarrow Y$, thì $XZ \rightarrow YZ$
- IR3: Luật bắc cầu: nếu $X \rightarrow Y$ và $Y \rightarrow Z$, thì $X \rightarrow Z$

=> 3 luật suy diễn trên là Hệ tiên đề Amstrong: đóng vai trò là một tập luật cần thiết và đầy đủ cho việc tạo ra bao đóng của một tập các phụ thuộc hàm.

- IR4: Luật chiếu: Nếu $X \rightarrow YZ$, thì $X \rightarrow Y$ và $X \rightarrow Z$
- IR5: Luật cộng thêm: Nếu $X \rightarrow Y$ và $X \rightarrow Z$, thì $X \rightarrow YZ$
- IR6: Luật giả bắc cầu: Nếu $X \rightarrow Y$ và $YZ \rightarrow W$, thì $XZ \rightarrow W$

Các luật suy diễn cho các phụ thuộc hàm

- Gọi F là tập các phụ thuộc hàm xác định trên lược đồ quan hệ R .
- ***Tập tất cả các phụ thuộc hàm được suy diễn từ một tập phụ thuộc hàm F được gọi là bao đóng của F và được ký hiệu là F^+ .***
 - Ký hiệu: $F \models X \rightarrow Y$ cho biết phụ thuộc hàm $X \rightarrow Y$ được suy diễn từ tập phụ thuộc hàm F .
 - $F^+ \equiv \{X \rightarrow Y \mid F \models X \rightarrow Y\}$

Xác định bao đóng

- F^+ là bao đóng của tập phụ thuộc hàm F nếu nó là tập (nhỏ nhất) bao gồm tất cả các phụ thuộc hàm được sinh ra từ F nhờ hệ tiên đề Armstrong.
- Để xác định liệu một phụ thuộc hàm $X \rightarrow Y$ có thỏa mãn lược đồ quan hệ R với tập phụ thuộc hàm F hay không thì cần xem $F \models X \rightarrow Y$ không, hoặc chính xác hơn là xem $X \rightarrow Y$ có nằm trong F^+ hay không.
- **Thực hiện bằng cách:** chỉ cần sinh ra bao đóng của tập thuộc tính X là X^+ . Kiểm tra xem Y có nằm trong X^+ hay không.

Thuật toán tìm bao đóng của tập thuộc tính

Thuật toán Closure {trả về X^+ trên F }

Đầu vào: tập thuộc tính X , và một tập phụ thuộc hàm F

Đầu ra: Tính X^+ trên F

Closure (X, F)

{

$X^+ \leftarrow X$;

 repeat

$\text{old}X^+ \leftarrow X^+$;

 for mỗi phụ thuộc hàm $W \rightarrow Z$ trong F do

 if $W \subseteq X^+$ then $X^+ \leftarrow X^+ \cup Z$;

 until ($\text{old}X^+ = X^+$);

}

Thuật toán Member

Thuật toán Member {xác định các thành viên trong F^+ }

Đầu vào: một tập các phụ thuộc hàm F ,
và một phụ thuộc hàm đơn $X \rightarrow Y$

Đầu ra: Trả lại kết quả đúng (true) nếu $F \models X \rightarrow Y$,
ngược lại trả lại kết quả sai (false)

Member ($F, X \rightarrow Y$)

```
{  
    if  $Y \subseteq \text{Closure}(X, F)$   
        then return true;  
        else return false;  
}
```

Phủ và sự tương đương của phụ thuộc hàm

- Một tập phụ thuộc hàm F được phủ bởi một tập phụ thuộc hàm G (hay nói cách khác: G phủ F) nếu mọi phụ thuộc hàm trong F đều nằm trong G^+ .
- Để xác định xem G có phủ F hay không: tính X^+ trên G cho mỗi phụ thuộc hàm $X \rightarrow Y$ trong F , nếu $Y \subseteq X^+$ cho mỗi X thì G phủ F .
- Hai tập phụ thuộc hàm F và G là tương đương (ký hiệu: $F \equiv G$) nếu $F^+ = G^+$.

=> Nếu $F \equiv G$ thì mọi phụ thuộc hàm trong G có thể được suy diễn từ F và mọi phụ thuộc hàm trong F có thể được suy diễn từ G , và ngược lại.

Phủ và sự tương đương của phụ thuộc hàm

- **Ý tưởng:** Sinh ra một tập các phụ thuộc hàm G tương đương với tập F ban đầu nhưng lại có số lượng phụ thuộc hàm là nhỏ nhất (phủ tối thiểu).
- Một tập các phụ thuộc hàm F được cho là không dư thừa nếu không có tập con thực sự G nào của F mà G tương đương với F .
- G là một phủ không dư thừa của F nếu G là một phủ của F và G không dư thừa.

Thuật toán Nonredundant

Thuật toán Nonredundant {sinh ra một phủ không dư thừa}

Đầu vào: một tập phụ thuộc hàm F

Đầu ra: một phủ không dư thừa G của F

Nonredundant (F)

{

$G \leftarrow F$;

 for mỗi phụ thuộc hàm $X \rightarrow Y \in G$ do

 if Member($G - \{X \rightarrow Y\}$, $X \rightarrow Y$)

 then $G \leftarrow G - \{X \rightarrow Y\}$;

 return (G);

}

Các thuộc tính dư thừa

- Nếu F là tập các phụ thuộc hàm trên lược đồ quan hệ R và $X \rightarrow Y \in F$ thì thuộc tính A được gọi là dư thừa trong $X \rightarrow Y \in F$ nếu:
 - $X = AZ$, $X \neq Z$ and $\{F - \{X \rightarrow Y\}\} \cup \{Z \rightarrow Y\} \equiv F$, hoặc
 - $Y = AW$, $Y \neq W$ and $\{F - \{X \rightarrow Y\}\} \cup \{X \rightarrow W\} \equiv F$

\Rightarrow Một thuộc tính A dư thừa trong $X \rightarrow Y$ nếu A có thể được loại bỏ khỏi vế trái hoặc vế phải của phụ thuộc hàm mà không làm thay đổi F^+ .

Tập phụ thuộc hàm tối giản trái và tối giản phải

Cho F là một tập các phụ thuộc hàm trên lược đồ R và cho $X \rightarrow Y \in F$.

- $X \rightarrow Y$ được gọi là tối giản trái nếu X không chứa các thuộc tính dư thừa A .
 - *Một phụ thuộc hàm tối giản về trái cũng được gọi là một phụ thuộc hàm đầy đủ.*
- $X \rightarrow Y$ được gọi là tối giản phải nếu Y không chứa các thuộc tính dư thừa A .
- $X \rightarrow Y$ được gọi là tối giản nếu nó tối giản trái, tối giản phải và Y khác rỗng.

Thuật toán Tối giản trái

Thuật toán tối giản trái {trả lại tập các phụ thuộc hàm tối giản về trái F}

Đầu vào: tập các phụ thuộc hàm G

Đầu ra: một phủ tối giản trái của G

Left-Reduce (G)

{

$F \leftarrow G$;

 for mỗi phụ thuộc hàm $X \rightarrow Y$ trong G do

 for mỗi thuộc tính A trong X do

 if Member(F, $(X-A) \rightarrow Y$)

 then loại bỏ A khỏi X trong $X \rightarrow Y$ của F

 return(F);

}

Thuật toán Tối giản phải

Thuật toán tối giản phải {trả lại tập các phụ thuộc hàm tối giản về trái F}

Đầu vào: tập các phụ thuộc hàm G

Đầu ra: một phủ tối giản phải của G

Right-Reduce (G)

```
{  
    F ← G;  
    for mỗi phụ thuộc hàm  $X \rightarrow Y$  trong G do  
        for mỗi thuộc tính A trong Y do  
            if Member( $F - \{X \rightarrow Y\} \cup \{X \rightarrow (Y - A)\}$ ,  $X \rightarrow A$ )  
                then loại bỏ A khỏi Y trong  $X \rightarrow Y$  của F  
    return(F);  
}
```

Thuật toán tối giản phụ thuộc hàm

Thuật toán tối giản phụ thuộc hàm

{trả lại tập các phụ thuộc hàm tối giản F}

Đầu vào: tập các phụ thuộc hàm G

Đầu ra: một tập các phụ thuộc hàm tối giản G

Reduce (G)

```
{  
    F ← Right-Reduce( Left-Reduce(G));  
    loại bỏ tất cả các phụ thuộc hàm dạng  $X \rightarrow \text{null}$  từ F;  
    return(F);  
}
```

Nếu G chứa một phụ thuộc hàm dư thừa $X \rightarrow Y$, thì mọi thuộc tính trong Y sẽ là dư thừa, và do vậy sẽ tối giản tới $X \rightarrow \text{null}$, nên sẽ phải loại bỏ các phụ thuộc hàm này.

Ví dụ

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

Hãy tối giản F

Cho $G = \{B \rightarrow A, D \rightarrow A, BA \rightarrow D\}$

Hãy tối giản G

Phủ tối thiểu

- **Định nghĩa:** Một tập phụ thuộc hàm F là tối thiểu nếu:
 1. Mọi phụ thuộc hàm đều có vẻ phải là một thuộc tính
 2. F là không dư thừa
 3. Không có phụ thuộc hàm nào dạng $X \rightarrow A$ có thể được thay thế bởi dạng $Y \rightarrow A$ với $Y \subseteq X$ và vẫn là một tập tương đương. Nói cách khác F là một tối giản trái.

- **Ví dụ:**

$$G = \{A \rightarrow BCE, AB \rightarrow DE, BI \rightarrow J\}$$

Một phủ tối thiểu của G :

$$F = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, BI \rightarrow J\}$$

Thuật toán tìm phủ tối thiểu

Thuật toán tìm phủ tối thiểu {trả lại phủ tối thiểu của F}

Đầu vào: tập các phụ thuộc hàm F

Đầu ra: một phủ tối thiểu của F

MinCover (F)

{

$G \leftarrow F$;

 thay thế mỗi phụ thuộc hàm $X \rightarrow A_1A_2...A_n$ của G

 thành n phụ thuộc hàm $X \rightarrow A_1, X \rightarrow A_2, ..., X \rightarrow A_n$

 Left-Reduce(G);

 Nonredundant(G);

 return(G);

}

Ví dụ

Cho lược đồ $R(A, B, C, D, E, G, H)$, hãy tìm phủ tối thiểu của $F = \{CD \rightarrow AB, C \rightarrow D, D \rightarrow EH, AE \rightarrow C, A \rightarrow C, B \rightarrow D\}$

Khoá của lược đồ

- **Siêu khóa:** Là một tập các thuộc tính xác định duy nhất thực thể trong quan hệ.
- **Khóa:** là một siêu khóa mà khi loại bỏ bất kỳ thuộc tính nào từ khóa này thì nó không còn là một siêu khóa nữa. Nghĩa là, khóa có số thuộc tính là nhỏ nhất.
- **Khóa dự bị:** là một tập các thuộc tính khóa nhỏ nhất của lược đồ quan hệ.
- **Khóa chính:** là một khóa dự bị được chọn ra. Tất cả các khóa dự bị còn lại trở thành khóa phụ hay khóa thứ cấp.

Các thuộc tính khoá

- ***Thuộc tính khoá:*** là thuộc tính của quan hệ R và là thành viên của một khóa dự bị nào đó.
- ***Thuộc tính không khoá:*** là thuộc tính của quan hệ R mà không phải là thành viên của một khóa dự bị nào.

Xác định khóa cho lược đồ quan hệ

- **Định nghĩa:** Nếu R là một lược đồ quan hệ với các thuộc tính A_1, A_2, \dots, A_n và một tập các phụ thuộc hàm F trong đó $X \subseteq \{A_1, A_2, \dots, A_n\}$ thì X là một khóa của R khi:

1. $X \rightarrow A_1 A_2 \dots A_n \in F^+$, và
2. Không có tập con thực sự nào $Y \subseteq X$ mà $Y \rightarrow A_1 A_2 \dots A_n \in F^+$.

\Rightarrow Về cơ bản, định nghĩa này cho thấy phải tạo ra bao đóng của tất cả các tập con có thể có của R và quyết định xem tập nào nhỏ nhất suy diễn ra được tất cả các thuộc tính của lược đồ.

Ví dụ xác định khóa

Cho $r = (C, T, H, R, S, G)$ với tập phụ thuộc hàm
 $F = \{C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R\}$
 Xác định khóa cho r

Bước 1: Tính $(A_i)^+$ với $1 \leq i \leq n$

$$C^+ = \{CT\}, \quad T^+ = \{T\}, \quad H^+ = \{H\}$$

$$R^+ = \{R\}, \quad S^+ = \{S\}, \quad G^+ = \{G\}$$

\Rightarrow Không có thuộc tính đơn nào là khóa của r

Bước 2: Tính $(A_i A_j)^+$ với $1 \leq i \leq n, 1 \leq j \leq n$

$$(CT)^+ = \{C, T\}, \quad (CH)^+ = \{CHTR\}, \quad (CR)^+ = \{CRT\}$$

$$(CS)^+ = \{CSGT\}, \quad (CG)^+ = \{CGT\}, \quad (TH)^+ = \{THRC\}$$

$$(TR)^+ = \{TR\}, \quad (TS)^+ = \{TS\}, \quad (TG)^+ = \{TG\}$$

$$(HR)^+ = \{HRCT\}, \quad (HS)^+ = \{HSRCTG\}, \quad (HG)^+ = \{HG\}$$

$$(RS)^+ = \{RS\}, \quad (RG)^+ = \{RG\}, \quad (SG)^+ = \{SG\}$$

Tập thuộc tính (HS) là một khóa của r

Ví dụ (tiếp)

Bước 3: Tính $(A_i A_j A_k)^+$ với $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq k \leq n$

$$(CTH)^+ = \{CTHR\},$$

$$(CTR)^+ = \{CTR\}$$

$$(CTS)^+ = \{CTSG\},$$

$$(CTG)^+ = \{CTG\}$$

$$(CHR)^+ = \{CHRT\},$$

$$(CHS)^+ = \{CHSTRG\}$$

$$(CHG)^+ = \{CHGTR\},$$

$$(CRS)^+ = \{CRSTG\}$$

$$(CRG)^+ = \{CRGT\},$$

$$(CSG)^+ = \{CSGT\}$$

$$(THR)^+ = \{THRC\},$$

$$(THS)^+ = \{THSRCG\}$$

$$(THG)^+ = \{THGRC\},$$

$$(TRS)^+ = \{TRS\}$$

$$(TRG)^+ = \{TRG\},$$

$$(TSG)^+ = \{TSG\}$$

$$(HRS)^+ = \{HRSCTG\},$$

$$(HRG)^+ = \{HRGCT\}$$

$$(HSG)^+ = \{HSGRCT\},$$

$$(RSG)^+ = \{RSG\}$$

Các siêu khóa được đánh dấu màu đỏ.

Ví dụ (tiếp)

Bước 4: Tính $(A_i A_j A_k A_r)^+$ với $1 \leq i \leq n, 1 \leq j \leq n, 1 \leq k \leq n, 1 \leq r \leq n$

$$(CTHR)^+ = \{CTHR\}, \quad (CTHS)^+ = \{CTHSRG\}$$

$$(CTHG)^+ = \{CTHGR\}, \quad (CHRS)^+ = \{CHRSTG\}$$

$$(CHRG)^+ = \{CHRGTR\}, \quad (CRSG)^+ = \{CRSGT\}$$

$$(THRS)^+ = \{THRSCG\}, \quad (THRG)^+ = \{THRGC\}$$

$$(TRSG)^+ = \{TRSG\}, \quad (HRSG)^+ = \{HRSGCT\}$$

$$(CTRS)^+ = \{CTRS\}, \quad (CTSG)^+ = \{CTSG\}$$

$$(CSHG)^+ = \{CSHGTR\}, \quad (THSG)^+ = \{THSGRC\}$$

$$(CTRG)^+ = \{CTRG\}$$

Các siêu khóa được đánh dấu màu đỏ.

Ví dụ (tiếp)

Bước 5: Tính $(A_i A_j A_k A_r A_s)^+$ với $1 \leq i \leq n, 1 \leq j \leq n, 1 \leq k \leq n,$
 $1 \leq r \leq n, 1 \leq s \leq n$

$$(CTHRS)^+ = \{CTHSRG\}$$

$$(CTHRG)^+ = \{CTHGR\}$$

$$(CTHSG)^+ = \{CTHSGR\}$$

$$(CHRSR)^+ = \{CHRSRGT\}$$

$$(CTRSG)^+ = \{CTRSG\}$$

$$(THRSG)^+ = \{THRSGC\}$$

Các siêu khóa được đánh dấu màu đỏ

Ví dụ (tiếp)

Bước 6: Tính $(A_i A_j A_k A_r A_s A_t)^+$ với $1 \leq i \leq n, 1 \leq j \leq n, 1 \leq k \leq n,$
 $1 \leq r \leq n, 1 \leq s \leq n, 1 \leq t \leq n$

$$(CTHRSG)^+ = \{CTHSRG\}$$

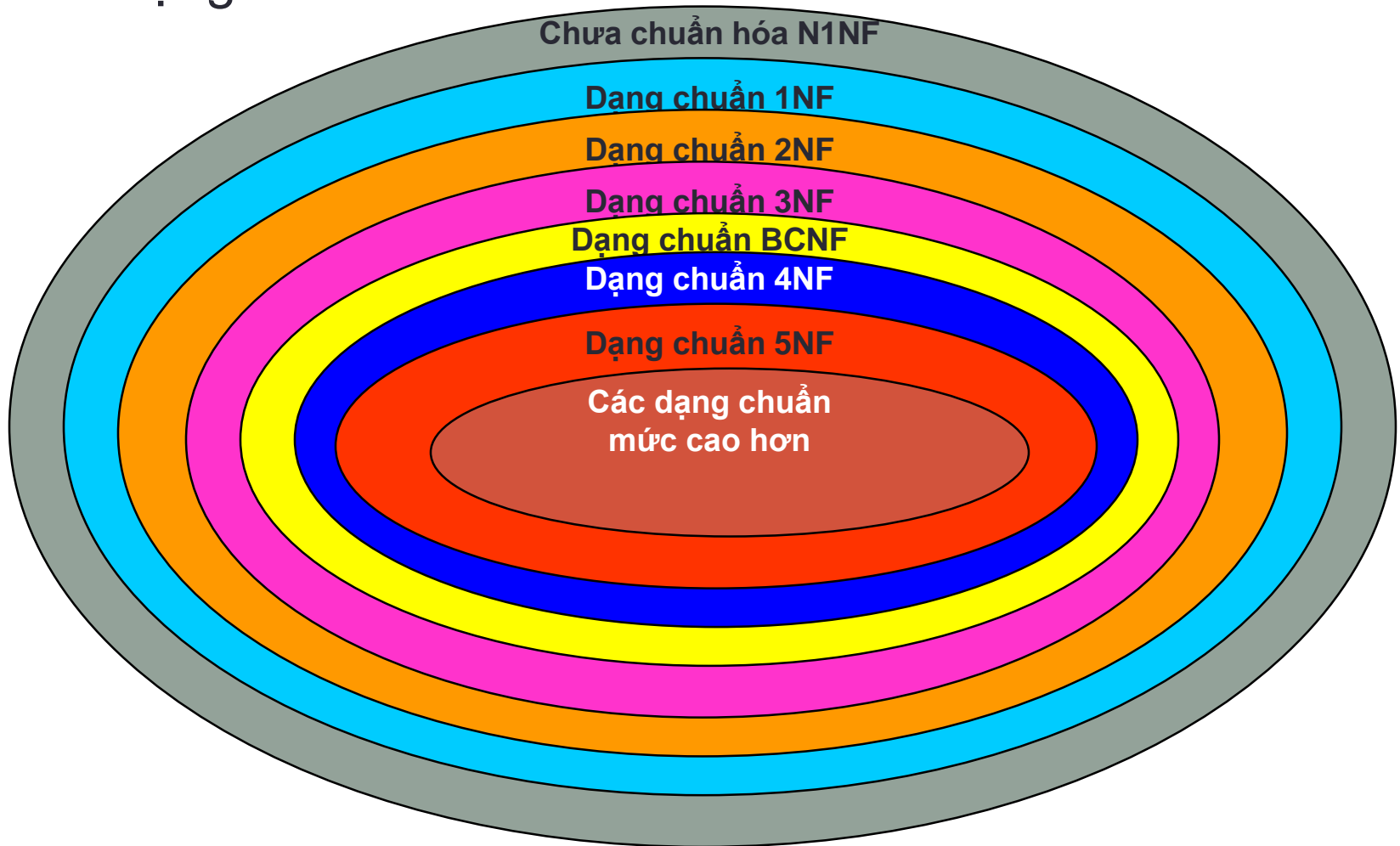
Siêu khóa được đánh dấu màu đỏ.

Với 6 thuộc tính, số trường hợp phải xét là:

$$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} + \binom{6}{4} + \binom{6}{5} + \binom{6}{6} = 6 + 15 + 20 + 15 + 1 = 63$$

Chuẩn hóa dựa trên khóa chính

- Các dạng chuẩn



Dạng chưa chuẩn hóa (N1NF)

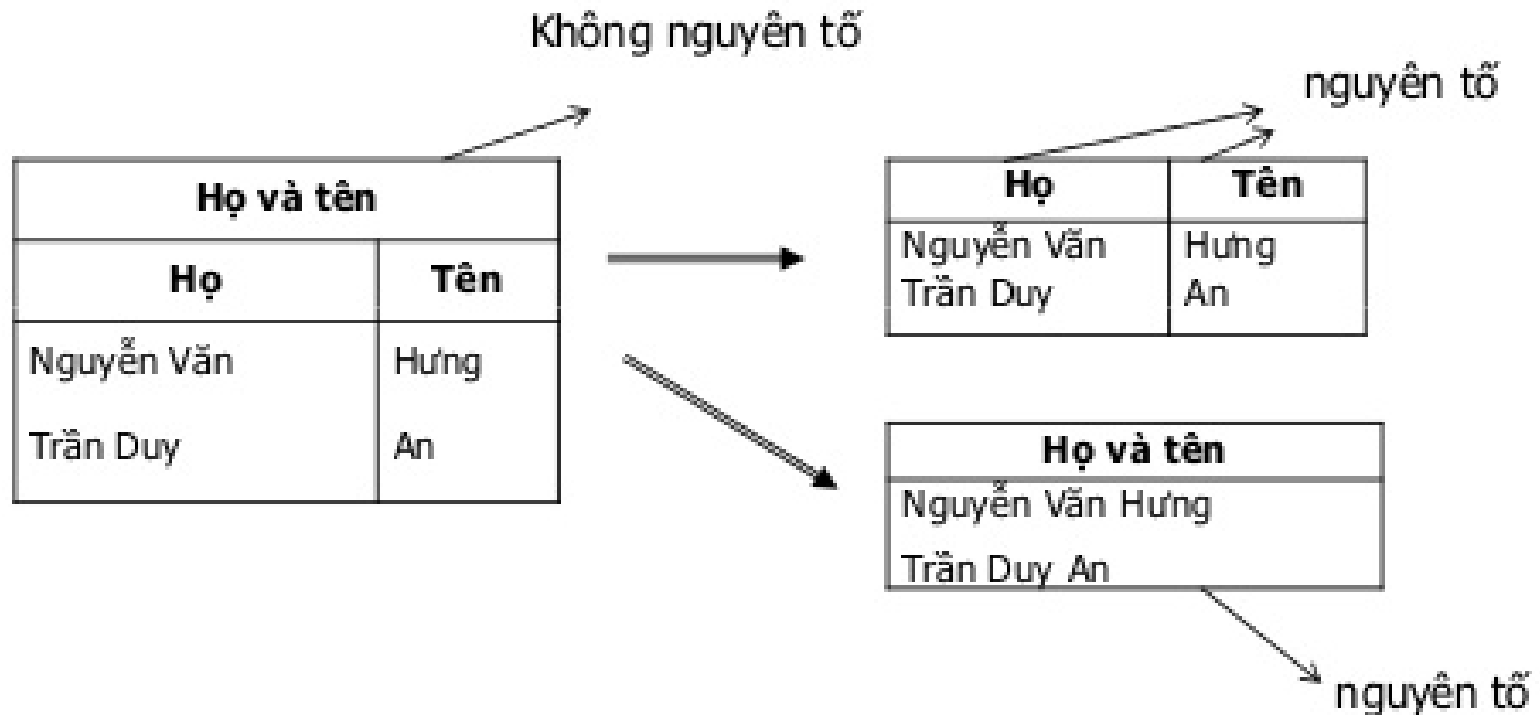
- **Các quan hệ chưa ở dạng chuẩn 1** chứa một hoặc một số thuộc tính không nguyên tố, các thuộc tính lặp, hoặc các thuộc tính dẫn xuất.
- **Thuộc tính chứa giá trị nguyên tố:** là những thuộc tính chứa giá trị không thể phân rã được nữa.

Dạng chuẩn 1 (First Normal Form 1NF)

- **Một quan hệ ở dạng chuẩn 1:**
 - Mọi giá trị thuộc tính của quan hệ đều ở dạng nguyên tố.
 - Giá trị của các thuộc tính là đơn trị.
 - Không có thuộc tính nào được suy ra trực tiếp từ các thuộc tính khác.

VÍ DỤ

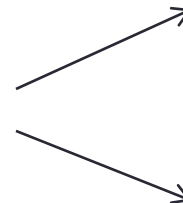
- Chuyển đổi các thuộc tính không nguyên tố về các thuộc tính nguyên tố.



VÍ DỤ

- Chuyển đổi các thuộc tính đa trị thành các thuộc tính đơn trị.

<u>SN</u>	Tên	Tuổi	Môn học
0012	Hùng	19	C++, Java
0123	Toản	20	C++
0032	Hồng	18	CSDL
0133	Anh	20	Toán, Java



<u>SN</u>	Tên	Tuổi
0012	Hùng	19
0123	Toản	20
0032	Hồng	18
0133	Anh	20

<u>SN</u>	<u>Môn học</u>
0012	C++
0012	Java
0123	C++
0032	CSDL
0133	Toán
0133	Java

Dạng chuẩn 2 (2NF)

- **Dạng chuẩn 2** dựa trên khái niệm phụ thuộc hàm đầy đủ.
- Một phụ thuộc hàm $X \rightarrow Y$ được gọi là phụ thuộc hàm đầy đủ nếu nó đã tối giản trái.
- Một lược đồ quan hệ R với tập phụ thuộc hàm F ở dạng chuẩn 2NF khi thỏa mãn:
 - Là dạng chuẩn 1NF
 - Mọi thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào mọi khóa của R.

Hoặc: không tồn tại thuộc tính không khóa mà phụ thuộc hàm một phần vào một khóa nào đó của R.

Chuẩn hoá từ 1NF về 2NF

- ❖ Loại bỏ các thuộc tính không khóa mà phụ thuộc một phần vào khóa chính để tách thành bảng riêng. Khóa chính của bảng riêng là bộ phận khóa mà chúng phụ thuộc vào.
- ❖ Các thuộc tính còn lại lập thành một bảng với khóa chính là khóa ban đầu.

Dạng chuẩn 3 (3NF)

- **Dạng chuẩn 3** dựa trên khái niệm phụ thuộc hàm bậc cầu.
- **Định nghĩa dạng chuẩn 3NF:** Một lược đồ quan hệ R ở dạng 3NF với một tập phụ thuộc hàm F nếu thỏa mãn:
 - Với bất kỳ phụ thuộc hàm $X \rightarrow A$ thì hoặc X là một siêu khóa của R, hoặc A là một thuộc tính khóa.
- **Một cách định nghĩa khác về dạng chuẩn 3NF:** Một lược đồ quan hệ R ở dạng 3NF với một tập phụ thuộc hàm F nếu thỏa mãn:
 - R ở dạng 2NF
 - Không có thuộc tính không khóa nào phụ thuộc hàm bậc cầu vào một khóa của R.

Chuẩn hóa từ 2NF về 3NF

❖ **Cách 1:**

- Tìm phủ tối thiểu của tập phụ thuộc hàm của lược đồ ban đầu
- Loại bỏ các thuộc tính phụ thuộc bắc cầu ra khỏi quan hệ, lập thành bảng mới với khóa chính là thuộc tính bắc cầu
- Các thuộc tính còn lại lập thành một bảng với khóa chính là khóa ban đầu.

❖ **Cách 2:**

- Tìm phủ tối thiểu (G) của tập phụ thuộc hàm (F) của lược đồ (R) ban đầu
- Ứng với mỗi phụ thuộc hàm trong G, lập một bảng quan hệ mới
- Nếu trong tất cả các bảng mới tạo ra tại bước 2 không có bảng nào chứa một khóa dự bị nào của R thì tạo ra thêm một bảng mới chứa một khóa dự bị của R.

Ví dụ dạng chuẩn 3NF

Cho $R = (A, B, C, D)$, khóa $K = \{AB\}$, và tập phụ thuộc hàm $F = \{AB \rightarrow CD, C \rightarrow D, D \rightarrow C\}$

R không là dạng chuẩn 3NF vì $C \rightarrow D$ và C không là một siêu khóa của R.

Cách giải thích khác:

R không ở dạng chuẩn 3NF vì $AB \rightarrow C$ và $C \rightarrow D$ và như vậy D là một thuộc tính không khóa mà lại phụ thuộc bắc cầu vào khóa AB.

Chuẩn hóa

- Chuẩn hóa về dạng 1NF: loại bỏ dữ liệu dư thừa
- Chuẩn hóa về 2NF: loại bỏ các phụ thuộc hàm bộ phận
- Chuẩn hóa về 3NF: loại bỏ các phụ thuộc hàm bắc cầu

VÍ DỤ

- Cho lược đồ (MS bệnh nhân, tên BN, MS hẹn, thời gian, bác sĩ) cùng các phụ thuộc hàm:
 - MS bệnh nhân \rightarrow tên BN
 - MS bệnh nhân, MS hẹn \rightarrow thời gian, bác sĩ
 - Thời gian \rightarrow MS hẹn

Hãy chuyển lược đồ trên sang dạng chuẩn 3NF.

Dạng chuẩn Boyce-Codd (BCNF)

- **Định nghĩa:** Một lược đồ quan hệ R được coi là ở dạng chuẩn Boyce-Codd với tập phụ thuộc hàm F nếu với bất kỳ phụ thuộc hàm $X \rightarrow A$ nào và $A \not\subseteq X$, thì X là một siêu khóa của R .
- **Ví dụ:** Cho lược đồ quan hệ $R = (A, B, C)$, tập phụ thuộc hàm $F = \{AB \rightarrow C, C \rightarrow A\}$ và khóa $K = \{AB\}$

R không ở dạng chuẩn BCNF vì có $C \rightarrow A$ và C không phải là một siêu khóa của R .

BCNF

- Sự khác nhau giữa dạng chuẩn 3NF và BCNF là BCNF cho phép bỏ đi luật với phụ thuộc hàm $X \rightarrow A$ thì X phải là thuộc tính khóa.
- Trong thực tế, hầu hết các lược đồ quan hệ ở dạng chuẩn 3NF thì cũng ở dạng chuẩn BCNF. Chỉ khi trong lược đồ có $X \rightarrow A$ mà X không phải là một siêu khóa hoặc A là một thuộc tính khóa thì lược đồ này ở dạng chuẩn 3NF mà không ở dạng chuẩn BCNF.

Phân tách lược đồ về dạng chuẩn

- Để đưa một lược đồ CSDL về dạng chuẩn 3NF hoặc BCNF, người thiết kế phải đảm bảo thỏa mãn 2 tính chất:
 1. Kết nối không mất mát thông tin
 2. Các phụ thuộc hàm được bảo toàn sau khi phân tách.
- Hiện tại, có một số thuật toán phân tách về 3NF đảm bảo cả 2 tính chất (1) và (2).
- Tuy nhiên, không có thuật toán nào phân tách về BCNF mà đảm bảo cả tính chất (1) và (2).

Phân tách về dạng chuẩn BCNF

- Kiểm tra xem lược đồ R có thuộc chuẩn BCNF hay không. Nếu không thì thực hiện tiếp các bước sau.
- Chọn một trong số các phụ thuộc hàm vi phạm chuẩn BCNF, giả sử là $X \rightarrow Y$
- Tìm X^+ . Tạo một lược đồ mới R1 gồm các thuộc tính trong X^+ . Các thuộc tính còn lại, bao gồm cả X hình thành một lược đồ mới R2.
- Tìm các phụ thuộc hàm cho R1 và R2 từ tập phụ thuộc hàm ban đầu.
- Chu trình lặp lại từ đầu cho đến khi tất cả các lược đồ thuộc dạng chuẩn BCNF.

VÍ DỤ

- Cho lược đồ (MS bệnh nhân, tên BN, MS hẹn, thời gian, bác sĩ) cùng các phụ thuộc hàm:
 - MS bệnh nhân \rightarrow tên BN
 - MS bệnh nhân, MS hẹn \rightarrow thời gian, bác sĩ
 - Thời gian \rightarrow MS hẹn

Hãy chuyển lược đồ trên sang dạng chuẩn BCNF.

Ví dụ

- Hãy chỉ ra các phụ thuộc hàm vi phạm chuẩn 3NF, BCNF trong các lược đồ sau:

R1 (A, B, C, D)

F1($AB \rightarrow C, C \rightarrow D, D \rightarrow A$)

R2 (A, B, C, D)

F2 ($AB \rightarrow C, BC \rightarrow D, CD \rightarrow A, AD \rightarrow B$)

R3 (A, B, C, D)

F3 ($A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A$)

R4 (A, B, C, D, E)

F4 ($AB \rightarrow C, DE \rightarrow C, B \rightarrow D$)

R5 (A, B, C, D, E)

F5 ($AB \rightarrow C, C \rightarrow D, D \rightarrow B, D \rightarrow E$)