CƠ SỞ DỮ LIỆU

Chương 4

Chương 4 - Lý thuyết thiết kế cơ sở dữ liệu quan hệ

NỘI DUNG:

- Tổng quan về thiết kế CSDL quan hệ
- Phụ thuộc hàm
- Phép tách các sơ đồ quan hệ (SĐQH)
- Các dạng chuẩn đối với các SĐQH

Tổng quan về thiết kế CSDLQH

- Vấn đề của một sơ đồ quan hệ được thiết kế chưa tốt:

  Giả sử ta cần một cơ sở dữ liệu lưu trữ thông tin về các hãng cung ứng. Sơ đồ quan hệ được thiết kế trong đó tất cả các thuộc tính cần thiết được lưu trong đúng 1 quan hệ:

  Suppliers(sid, sname, city, numofemps, product, quantity)

sid	sname	city	NOE	product	quantity
S1	Smith	London	100	Screw	50
S1	Smith	London	100	Nut	100
S2	J&J	Paris	124	Screw	78
S3	Blake	Tokyo	75	Bolt	100

- **Dư thừa dữ liệu**: Hãng nào cung ứng nhiều hơn một mặt hàng thì thông tin của hãng đó sẽ bị lặp lại trong bảng (VD S1), mặt hàng được cung ứng bởi nhiều hãng cũng bị lặp lại (VD Screw)
- **Dị thường dữ liệu khi thêm**: Nếu có một hãng chưa cung cấp mặt hàng nào, vậy giá trị cho thuộc tính product và quantity trong bộ dữ liệu mới được thêm vào sẽ không được xác định
- **Dị thường dữ liệu khi xóa**: Nếu một hãng chỉ cung cấp một mặt hàng, nếu ta muốn xóa thông tin về sự cung cấp này thì ta sẽ mất thông tin về hãng cung cấp
- **Dị thường dữ liệu khi sửa đổi**: Do thông tin bị lặp lại nên việc sửa đổi một bộ dữ liệu có thể dẫn đến việc không nhất quán trong dữ liệu về một hãng nếu sơ sót không sửa đổi trên toàn bộ các bộ giá trị liên quan đến hãng đó

Đề xuất giải pháp

- Nếu sơ đồ trên được thay thế bằng 2 sơ đồ quan hệ
  - *Supp(sid, sname, city, numofemps)*
  - *Supply(sid, product, quantity)*

  thì tất cả các vấn đề nêu ở trên sẽ được loại bỏ. Tuy nhiên, khi tìm kiếm dữ liệu thì phải kết nối 2 bảng chứ không chỉ là chọn và chiếu trên một bảng như ở cách thiết kế trước.

Mục đích của chuẩn hoá

- Xác định được một tập các lược đồ quan hệ, cho phép tìm kiếm thông tin một cách dễ dàng, **đồng thời** tránh được dư thừa dữ liệu.
- Hướng tiếp cận:
  - Một trong những kỹ thuật được sử dụng là Tách các lược đồ quan hệ có vấn đề thành những lược đồ quan hệ chuẩn hơn.
  - Phụ thuộc hàm (functional dependencies) được sử dụng để nhận biết các lược đồ chưa chuẩn và đề xuất hướng cải tiến.

Phụ thuộc hàm

- Định nghĩa:
  - Cho R(U) là một sơ đồ quan hệ với U là tập thuộc tính $\{A_1, A_2, \ldots, A_n\}$. X, Y là tập con không rỗng của U.
  - Nói X xác định hàm Y, hay Y là phụ thuộc hàm vào X (viết: $X \rightarrow Y$) nếu với một quan hệ r xác định trên R(U) và với 2 bộ bất kỳ t_1, t_2 thuộc r mà $t_1[X] = t_2[X]$ thì $t_1[Y] = t_2[Y]$
- Bản chất, nếu 2 bộ giống nhau về giá trị của các thuộc tính X thì cũng giống nhau về giá trị của các thuộc tính Y.
- Phụ thuộc hàm là một trường hợp của ràng buộc toàn vẹn, tổng quát hóa khái niệm khóa.

- Ví dụ 1: AB → C

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a1	b2	c2	d1
a2	b1	c3	d1

- Ví dụ 2: trong cơ sở dữ liệu mẫu dùng trong Chương 3, ta có bảng S, với mỗi giá trị của sid đều tồn tại một giá trị tương ứng cho sname, city và status.

  Do đó, có sid → sname, sid → city, sid → status

Hệ tiên đề Amstrong đối với phụ thuộc hàm

Cho R(U) là 1 sơ đồ quan hệ, U là tập các thuộc tính.

$X, Y, Z, W \subseteq U$, Ký hiệu: $XY = X \cup Y$

- Phản xạ (*reflexivity*) : Nếu $Y \subseteq X$ thì $X \rightarrow Y$
- Tăng trưởng (*augmentation*) : Nếu $X \rightarrow Y$ thì $XZ \rightarrow YZ$
- Bắc cầu (*transitivity*) : Nếu $X \rightarrow Y$, $Y \rightarrow Z$ thì $X \rightarrow Z$

Hệ quả của hệ tiên đề Amstrong

- Luật hợp (*union*): Nếu $X \rightarrow Y$, $X \rightarrow Z$ thì $X \rightarrow YZ$.
- Luật tựa bắc cầu (*pseudo-transitivity*)

  Nếu $X \rightarrow Y$, $WY \rightarrow Z$ thì $XW \rightarrow Z$.
- Luật tách (*decomposition*): Nếu $X \rightarrow Y$, $Z \subseteq Y$ thì $X \rightarrow Z$

- Ví dụ 1:

  Cho tập phụ thuộc hàm {AB→C, C→A}

  Chứng minh: BC → ABC

C → A	BC → AB
AB → C	AB → ABC
BC → AB, AB → ABC	BC → ABC

- Ví dụ 2:

  Cho lược đồ quan hệ R(ABEIJGH) và tập phụ thuộc hàm F = {AB→E, AG→J, BE→I, E→G, GI→H}

  Chứng minh: AB → GH

Bao đóng của một tập phụ thuộc hàm

- **Định nghĩa:**
  - Cho F là một tập phụ thuộc hàm. Bao đóng của F ký hiệu là F^+ là tập lớn nhất chứa các phụ thuộc hàm có thể được suy ra từ các phụ thuộc hàm trong F.
- Đặc điểm của bao đóng của một tập phụ thuộc hàm:
  - Có thể rất lớn
  - Chi phí rất tốn kém cho việc tìm kiếm
- Vấn đề đặt ra: Kiểm tra xem một phụ thuộc hàm có được suy diễn từ một tập phụ thuộc hàm có sẵn không => sử dụng bao đóng của một tập thuộc tính đối với tập phụ thuộc hàm.

Bao đóng của một tập các thuộc tính đối với một tập các phụ thuộc hàm

- Định nghĩa:
  - Cho một sơ đồ quan hệ R(U), F là một tập phụ thuộc hàm trên U. X là tập con của U. Bao đóng của tập thuộc tính X đối với tập F, ký hiệu là X^+_F (X^+), là tập tất cả các thuộc tính được xác định hàm bởi X thông qua tập F

$$X^+ = \{A \in U |\ X \to A \in F^+\}$$

- Có thể thấy, định nghĩa về bao đóng của một tập thuộc tính dựa trên bao đóng của tập phụ thuộc hàm.

  =>Thuật toán xác định bao đóng của một tập thuộc tính

Thuật toán 1: Tìm bao đóng của một tập thuộc tính đối với tập phụ thuộc hàm

- **Vào**: Tập hữu hạn các thuộc tính U, tập các pth F trên U, $X \subseteq U$
- **Ra**: X^+_F
- **Thuật toán**

Bước 0: $X^0 = X$

Bước i: Tính X^i từ X^{i-1}

Nếu $\exists\ Y \to Z \in F$ và $Y \subseteq X^{i-1}$ và $A \in Z$ và $A \notin X^{i-1}$

thì $X^i = X^{i-1} \cup A$

ngược lại, $X^i = X^{i-1}$

Nếu $X^i \neq X^{i-1}$

thì lặp **Bước i**

ngược lai, chuyển **Bước i+1**

Bước cuối cùng: khi $X^i = X^{i-1}$ (không tăng thêm): $X^+_F = X^i$

Ví dụ

- Cho R(U) , U = {A, B, C, D, E, F}
  F = {AB→C, BC→AD, D→E, CF→B}
  Tính $(AB)^+$
- Thực hiện:
  - Bước 0: X^0 = AB
  - Bước 1: X^1 = ABC (do AB→ C)
  - Bước 2: X^2 = ABCD (do BC→AD)
  - Bước 3: X^3 = ABCDE (do D→E)
  - Bước 4: X^4 = ABCDE

Bổ đề

- $X \rightarrow Y$ được suy diễn từ tập F dựa trên hệ tiên đề Amstrong khi và chỉ khi $Y \subseteq X^{+}{}_{F}$
- Chứng minh:
  - Giả sử $Y=A_1...A_n$, với $A_1, ..., A_n$ là các thuộc tính và $Y \subseteq X^+$
  - Từ Định nghĩa X^+ ta có $X \rightarrow A_i$. Áp dụng tiên đề Amstrong cho mọi i, suy ra $X \rightarrow Y$ nhờ luật hợp.
  - Ngược lại, giả sử có $X \rightarrow Y$, áp dụng hệ tiên đề Amstrong cho mỗi i, ta có $X \rightarrow A_i$, $A_i \in Y$ nhờ luật tách. Từ đó suy ra $Y \subseteq X^+$

Khoá tối thiểu

- Định nghĩa: Cho sơ đồ quan hệ R=<U,F>, U là tập thuộc tính, F là một tập các phụ thuộc hàm xác định trên U. K được gọi là khoá tối thiểu của R nếu:
  - $K \subseteq U$
  - $K \rightarrow U \in F^+$
  - Với mọi $K' \subset K$, thì $K' \rightarrow U \notin F^+$
- Với những gì đã đề cập trong phần bao đóng ở trên, có thể nói, để thỏa mãn là một khoá tối thiểu thì $K^+ = U$ và K là tập thuộc tính nhỏ nhất có tính chất này.

Thuật toán 2: Tìm khoá tối thiểu

- **Vào**: $U = \{A_1, A_2, \ldots, A_n\}$, F
- **Ra**: khoá tối thiểu K xác định được trên U và F
- **Thuật toán**

Bước 0: $K^0 = U$

Bước i: Nếu $(K^{i-1} \setminus \{A_i\}) \rightarrow U$

thì $K^i = K^{i-1} \setminus \{A_i\}$

ngược lại, $K^i = K^{i-1}$

Bước n+1: $K = K^n$

Ví dụ

Cho U = {A, B, C, D, E}, F = {AB→C, AC→B, BC→DE}. Tìm một khoá tối thiểu của một quan hệ r xác định trên U và F

Thực hiện

B0: K^0= U = ABCDE

B1: Kiểm tra xem có tồn tại phụ thuộc hàm (K^0\{A})→U (BCDE→U) hay không. Ta cần phải sử dụng Thuật toán 1 để kiểm tra điều kiện tương đương là $(BCDE)^+$ có bằng U không. $(BCDE)^+$= BCDE , khác U. Vậy K^1 = K^0 = ABCDE

B2: Tương tự, thử loại bỏ B ra khỏi K^1 ta có $(ACDE)^+$ = ABCDE = U. Vậy K^2 = K^1 \ {B} = ACDE

B3: K^3 = ACDE

B4: K^4 = ACE

B5: K^5 = AC

B6: Vậy AC là một khoá tối thiểu mà ta cần tìm

Thuật toán khác tìm tất cả các khóa trong lược đồ quan hệ

Gọi:

- U là tập tất cả các thuộc tính CSDL, F là tập phụ thuộc hàm
- L(left): là các thuộc tính xuất hiện bên trái, R(right): là các thuộc tính xuất hiện ở vế phải
- S(superkey): là tập các siêu khóa, K(key): là tập các khóa

Tập thuộc tính nguồn (TN): gồm các thuộc tính chỉ xuất hiện ở vế trái, không xuất hiện ở vế phải của F và các thuộc tính không xuất hiện ở cả vế trái và vế phải của F.

Vậy TN = U \ R

- **Ví dụ**: Cho sơ đồ U = {A, B, C, D, E},

  F = {AB→C, AC→B, BC→DE}

  L = {A, B, C}, R = {B, C, D, E}, TN = U \ R = {A}

Tập thuộc tính đích (TĐ): gồm các thuộc tính chỉ xuất hiện ở R, không xuất hiện ở L. **Vậy TĐ = R \ L**

- **Ví dụ**: Cho L = {A, B, C}, R = {B, C, D, E}

  TĐ = {D, E}

Tập thuộc tính trung gian (TG): chứa các thuộc tính xuất hiện ở cả L và R. **Vậy TG = L ∩ R**

- **Ví dụ**: Cho L = {A, B, C}, R = {B, C, D, E}

  TG = L ∩ R = {B, C}

Thuật toán khác tìm tất cả các khóa trong lược đồ quan hệ

Thuật toán:

Bước 1: Tìm tập thuộc tính nguồn TN và tập thuộc tính trung gian TG

Bước 2: Nếu TG = Ø thì K(Key) = TN, và kết thúc thuật toán, xuất ra K của sơ đồ quan hệ <U,F>

Ngược lại, nếu TG ≠ Ø thì qua Bước 3

Bước 3: Tìm tất cả các tập con X_i của TG

Bước 4: Tìm Siêu khóa (S_i)

Với $\forall X_i$, nếu $(TN \cup X_i)^+ = U$ thì khi đó $S_i = TN \cup X_i$

Bước 5: Tìm Khóa (K_i) bằng cách loại bỏ các siêu khóa không tối thiểu

- Với mọi S_i S_j thuộc S, nếu S_i chứa trong S_j thì loại bỏ S_j ra khỏi tập siêu khóa. Khi đó, tập S còn lại chính là tập khóa cần tìm

- Ví dụ : Cho sơ đồ quan hệ R = <U, F>, với U = {A, B. C}, F = {AB→C, C→A}. Tìm tất cả các khóa thuộc tập cơ sở dữ liệu trên.
- Lời giải:
  - L = {ABC}   R = {CA}   TN = {B}
  - TG = {AC}≠∅ nên ta làm tiếp Bước 3
  - Ta có tập con X_i của tập TG = {∅, A, C, AC}
  - Lấy từng thuộc tính thuộc tập con X_i của tập TG hợp với TN ta có các thuộc tính sau:

  + $S_1 = TN \cup \emptyset = B$, có $B^+ = B \neq U$ nên $S_1 = B$ không là siêu khóa

  + $S_2 = TN \cup A = AB$ Ta có $AB^+ = ABC = U$ nên $S_2 = AB$ là siêu khóa

  + $S_3 = TN \cup C = BC$ Ta có $BC^+ = ABC = U$ nên $S_3 = BC$ là siêu khóa

  + $S_4 = TN \cup AC = ABC$ Ta có $ABC^+ = ABC = U$ nên $S_4 = ABC$ là siêu khóa

  - Vậy ta có tập siêu khóa S = {AB, BC, ABC}. Tuy nhiên, vì AB ⊂ ABC và BC ⊂ ABC nên loại bỏ ABC ra khỏi tập siêu khóa
  - Vậy ta có, tập K = {AB, BC} là khóa của lược đồ quan hệ

Nhận xét về phụ thuộc hàm

- Từ một tập các phụ thuộc hàm có thể suy diễn ra các phụ thuộc hàm khác
- Trong một tập phụ thuộc hàm cho sẵn, có thể có các phụ thuộc hàm bị coi là dư thừa

→ Làm thế nào để có được một tập phụ thuộc hàm tốt?

Hai tập phụ thuộc hàm tương đương

- **Định nghĩa**: Tập phụ thuộc hàm F là phủ của tập phụ thuộc hàm G, nếu $G \subset F^+$, G là phủ của F, nếu $F \subset G^+$, hay F và G tương đương nếu $F^+ = G^+$, ký hiệu là $F \cong G$
- Kiểm tra tính tương đương của 2 tập phụ thuộc hàm
  - Bước 1. Nếu với $\forall$ phụ thuộc hàm $f_i \in F$, f_i có dạng $X_{fi} \rightarrow Y_{fi}$, mà $f_i \in G^+$ thì $F^+ \subseteq G^+$. Kiểm tra $f_i \in G^+$ bằng cách kiểm tra $Y_{fi} \subseteq (X_{fi})^+_G$
  - Bước 2. Tương tự, nếu $\forall$ phụ thuộc hàm $g_j \in G$, mà $g_j \in F^+$ thì $G^+ \subseteq F^+$
  - Bước 3. Nếu $F^+ \subseteq G^+$ và $G^+ \subseteq F^+$ thì $F \cong G$

Ví dụ

- Cho sơ đồ quan hệ R(U) với U = {A, B, C, D, E, F}

  F = {AB→C, D→EF, C→BD}

  G = {AC→B, D→EF, B→CD}

  Hỏi F và G có phải là 2 tập phụ thuộc hàm tương đương hay không?
- Thực hiện:

  Đối với các phụ thuộc hàm trong F

  - f_1= AB→C. AB^+_G = ABCDEF = U. Vậy $f_1 \in G^+$
  - f_2= D→EF ∈ G nên chắc chắn $\in G^+$
  - f_3= C→BD. C^+_G = C không chứa BD. Vậy $f_3 \notin G^+$
  - Kết luận $F \not\equiv G$

Thuật toán 3: Tập phụ thuộc hàm không dư thừa

- **Định nghĩa**: Tập phụ thuộc hàm F là không dư thừa nếu không tồn tại $X \rightarrow Y \in F$ sao cho $F \setminus \{X \rightarrow Y\} \approx F$.
- **Thuật toán 3**: Tìm phủ không dư thừa của một tập phụ thuộc hàm
  - Vào: Tập thuộc tính U, $F = \{L_i \rightarrow R_i \mid i = 1..n\}$
  - Ra : Phủ không dư thừa F' của F
  - **Thuật toán**

    Bước 0: $F^0 = F$

    Bước i: Nếu $F^{i-1} \setminus \{L_i \rightarrow R_i\} \approx F^{i-1}$

    thì $F^i = F^{i-1} \setminus \{L_i \rightarrow R_i\}$

    ngược lại, $F^i = F^{i-1}$

    Bước n+1: $F' = F^n$

Phủ tối thiểu của một tập phụ thuộc hàm

- Định nghĩa: F_c được gọi là phủ tối thiểu của một tập phụ thuộc hàm F nếu thỏa mãn ba điều kiện:
  - (Đk1) Với $\forall f \in F_c$, f có dạng $X \rightarrow A$, trong đó A là một thuộc tính.
  - (Đk2) Với $\forall f = X \rightarrow Y \in F_c$, không tồn tại $A \in X$ (A là một thuộc tính) mà $(F_c \setminus f) \cup \{(X \setminus A) \rightarrow Y\} \cong F_c$
  - (Đk3) không tồn tại $X \rightarrow A \in F_c$ mà $F_c \setminus \{X \rightarrow A\} \cong F_c$

Thuật toán 4: Tìm phủ tối thiểu của một tập phụ thuộc hàm

- **Vào**: Tập thuộc tính U, $F = \{L_i \rightarrow R_i: i = 1..n\}$
- **Ra**: phủ tối thiểu F_c của tập phụ thuộc hàm F
- **Thuật toán**

B.1. Biến đổi F về dạng $F_1=\{L_i \rightarrow A_j\}$

trong đó A_j là một thuộc tính bất kỳ thuộc U (thoả mãn đk1)

B.2. Loại bỏ thuộc tính thừa trong vế trái của các phụ thuộc hàm

Lần lượt giản ước từng thuộc tính trong vế trái của từng phụ thuộc hàm trong F_1 thu được F_1'. Nếu $F_1' \cong F_1$ thì loại bỏ thuộc tính đang xét

Khi không có sự giản ước nào xảy ra nữa, thu được F_2 thỏa mãn đk2

B.3. Loại bỏ phụ thuộc hàm dư thừa

Lần lượt kiểm tra từng phụ thuộc hàm f. Nếu $F_2 \setminus f \cong F_2$ thì loại bỏ f

Khi không còn phụ thuộc hàm nào có thể loại bỏ thì thu đươc F_3 thoả mãn đk3

B.4. $F_c = F_3$

Ví dụ 1

- U = {A,B,C}

  F = {A→BC, B→C, A→B, AB→C}. Tìm phủ tối thiểu của F?

  - F_1 = {A→B, A→C, B→C, AB→C}
  - Xét các phụ thuộc hàm trong F_1 mà vế trái có nhiều hơn một thuộc tính, chỉ có AB→C. Giản ước A thì ta còn B→C có trong F_1, vậy A là thuộc tính thừa. Tương tự ta cũng tìm được B là thừa, vậy loại bỏ luôn AB→C khỏi F_1. F_2 = {A→B, A→C, B→C}
  - Bỏ phụ thuộc hàm thừa: A→C là thừa.

    Vậy F_c = {A→B, B→C}

Ví dụ 2

Tìm phủ tối thiểu của tập phụ thuộc hàm sau:

$F = \{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow G, ACDF \rightarrow EG\}$

- Bước 1: có $F_1=\{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow G, ACDF \rightarrow E, ACDF \rightarrow G\}$
- Bước 2: $F_2^0=F_1=\{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow G, ACDF \rightarrow E, ACDF \rightarrow G\}$
  - Loại bỏ thuộc tính thừa ở vế trái của 4 pth $ABCD \rightarrow E$, $EF \rightarrow G$, $ACDF \rightarrow E$ và $ACDF \rightarrow G$
  - Xét $ABCD \rightarrow E$: có B dư thừa, vì $(ACD)^+_{F2^0}=ACDBE$, chứa E, vậy $ACD \rightarrow E$ được suy diễn ra từ F_2^0
  - Xét $EF \rightarrow G$: không dư thừa
  - Xét $ACDF \rightarrow E$: dư thừa F, còn lại $ACD \rightarrow E$
  - Xét $ACDF \rightarrow G$: không dư thừa
  - Vậy $F_2^2=\{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G, ACDF \rightarrow G\}$
- Bước 3: $F_3^0=F_2^2=\{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G, ACDF \rightarrow G\}$
  - Loại bỏ các phụ thuộc hàm dư thừa trong F_3^0
  - Xét $ACDF \rightarrow G$: $(ACDF)^+=ACDFBEG$, chứa G. Vậy loại bỏ phụ thuộc hàm này trong F_3^0,

  Vậy $F_C= F_3^1=\{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G\}$

Phép tách các sơ đồ quan hệ

- Mục đích
  - Thay thế một sơ đồ quan hệ $R(A_1, A_2, \ldots, A_n)$ bằng một tập các sơ đồ con $\{R_1, R_2, \ldots, R_k\}$ trong đó $R_i \subseteq R$ và $R = R_1 \cup R_2 \cup \ldots \cup R_k$
- Yêu cầu của phép tách
  - Bảo toàn thuộc tính, ràng buộc
  - Bảo toàn dữ liệu

Phép tách không mất mát thông tin

- Định nghĩa: Cho sơ đồ quan hệ R(U) phép tách R thành các sơ đồ con $\{R_1, R_2, \ldots, R_k\}$ được gọi là phép tách không mất mát thông tin đối với một tập phụ thuộc hàm F nếu với mọi quan hệ r xác định trên R thỏa mãn F thì:

  $r = \Pi_{R1}(r) \bowtie \Pi_{R2}(r) \bowtie \ldots \bowtie \Pi_{Rk}(r)$

- Ví dụ: Phép tách **mất mát** thông tin

  Supplier(sid, sname, city, NOE, pid, pname, colour, quantity)

  →S1(sid, sname, city, NOE) và SP1(pid, pname, colour, quantity)
- Ví dụ: Phép tách **không mất mát** thông tin

  →S1(sid, sname, city, NOE) và

  SP2(sid, pid, pname, colour, quantity)

Định lý tách đôi

- Cho sơ đồ quan hệ R(U), tập phụ thuộc hàm F , phép tách R thành $R_1(U_1)$, $R_2(U_2)$ là một phép tách không mất mát thông tin nếu một trong hai phụ thuộc hàm sau là thỏa mãn trên F^+:

$$U_1 \cap U_2 \rightarrow U_1 - U_2$$

$$U_1 \cap U_2 \rightarrow U_2 - U_1$$

- Hệ quả: Cho sơ đồ quan hệ R(U) và phụ thuộc hàm $X \rightarrow Y$ thỏa mãn trên R(U). Phép tách R thành hai lược đồ con $R_1(U_1)$, $R_2(U_2)$ là một phép tách không mất mát thông tin với:

$$U_1 = XY$$

$$U_2 = XZ$$

$$Z = U \setminus XY$$

Thuật toán 5: Kiểm tra tính không mất mát thông tin của một phép tách

- **Vào**: $R(A_1, A_2, \ldots, A_n)$, F, phép tách $\{R_1, R_2, \ldots, R_k\}$
- **Ra**: phép tách là mất mát thông tin hay không
- **Thuật toán**

Bước 1. Thiết lập một bảng k hàng, n cột

Nếu A_j là thuộc tính của R_i thì điền a_j vào ô (i,j).

Nếu không thì điền b_{ij}.

Bước i. Xét $f = X \rightarrow Y \in F$

Nếu $\exists$ hai hàng t1, t2 thuộc bảng: t1[X] = t2[X] thì đồng nhất t1[Y] = t2[Y], ưu tiên về giá trị a.

Lặp cho tới khi không thể thay đổi được giá trị nào trong bảng

Bước cuối. Nếu bảng có một hàng gồm các kí hiệu $a_1, a_2, \ldots, a_n$ thì phép tách là không mất mát thông tin

ngược lại, phép tách không bảo toàn thông tin

Ví dụ

- R=ABCD được tách thành R_1=AB, R_2 =BD, R_3=ABC, R_4=BCD, F = {A→C, B→C, CD→B, C→D}
- Bước 1: Tạo bảng gồm 4 hàng, 4 cột
- Bước 2: Từ A→C, thay hàng 1, cột 3 bằng a_3
- Bước 3: Từ B→C, thay hàng 2, cột 3 bằng a_3
- Bước 4: Từ C→D, thay ô (1,4) và (3,4) bằng a_4
- Vậy, có hai hàng có toàn các giá trị a_j, chứng tỏ phép tách đã cho không mất mát thông tin

	A	B	C	D
R_1	a_1	a_2	b_{13}	b_{14}
R_2	b_{21}	a_2	b_{23}	a_4
R_3	a_1	a_2	a_3	b_{34}
R_4	b_{41}	a_2	a_3	a_4

	A	B	C	D
R_1	a_1	a_2	a_3	b_{14}
R_2	b_{21}	a_2	a_3	a_4
R_3	a_1	a_2	a_3	b_{34}
R_4	b_{41}	a_2	a_3	a_4

	A	B	C	D
R_1	a_1	a_2	a_3	a_4
R_2	b_{21}	a_2	a_3	a_4
R_3	a_1	a_2	a_3	a_4
R_4	b_{41}	a_2	a_3	a_4

Phép tách bảo toàn tập phụ thuộc hàm

- Hình chiếu của tập phụ thuộc hàm

  Cho sơ đồ quan hệ R, tập phụ thuộc hàm F, phép tách $\{R_1, R_2, \ldots, R_k\}$ của R trên F.

  Hình chiếu F_i của F trên R_i là tập tất cả $X \rightarrow Y \in F^+$:

$$XY \subseteq R_i$$

- Phép tách sơ đồ quan hệ R thành $\{R_1, R_2, \ldots, R_k\}$ là một phép tách bảo toàn tập phụ thuộc hàm F nếu

$$(F_1 \cup F_2 \ldots \cup F_k)^+ = F^+$$

  hay hợp của tất cả các phụ thuộc hàm trong các hình chiếu của F lên các sơ đồ con sẽ suy diễn ra các phụ thuộc hàm trong F.

Ví dụ

- **Ví dụ 1**: R = {A, B, C} F = { A→B, B→C, C→A} được tách thành R_1 = AB, R_2 = BC. Phép tách này có phải là bảo toàn tập phụ thuộc hàm không?
- **Ví dụ 2**: R = {A, B, C} , F = {AB→C, C→B} được tách thành R_1 = AB, R_2 = BC. Phép tách này có bảo toàn tập phụ thuộc hàm không, có mất mát thông tin không?
- **Ví dụ 3**: R = { A, B, C, D} , F = {A→B, C→D} được tách thành R_1 = AB, R_2 = CD. Phép tách này có bảo toàn tập phụ thuộc hàm không, có mất mát thông tin không?
- Vậy một phép tách có bảo toàn tập phụ thuộc hàm thì không đảm bảo là nó sẽ không mất mát thông tin và ngược lại

Các dạng chuẩn đối với SĐQH

- Vấn đề thiết kế CSDL quan hệ:
  – Có cần thiết phải tinh chỉnh thiết kế?
  – Thiết kế có được đã là tốt hay chưa?

  => các dạng chuẩn: có một vài dạng chuẩn, có thể coi như là một số các vấn đề về dư thừa dữ liệu hay dị thường dữ liệu đã được ngăn ngừa hay tối thiểu hóa
- Các dạng chuẩn mà chúng ta quan tâm
  – Dạng chuẩn 1 (1NF)
  – Dạng chuẩn 2 (2NF)
  – Dạng chuẩn 3 (3NF)
  – Dạng chuẩn Boyce-Code (BCNF)

Dạng chuẩn 1 (1NF)

- **Định nghĩa**: Một sơ đồ quan hệ R được gọi là ở dạng chuẩn 1 nếu tất cả các miền giá trị của các thuộc tính trong R đều chỉ chứa giá trị nguyên tố
  – Giá trị nguyên tố là giá trị mà không thể chia nhỏ ra được nữa
- Một quan hệ r xác định trên sơ đồ quan hệ ở dạng chuẩn 1 thì quan hệ đấy là ở dạng chuẩn 1
- Ví dụ: Quan hệ không ở dạng chuẩn 1 và quan hệ sau khi chuẩn hóa về dạng chuẩn 1

<table>
<tr><th rowspan="2">sname</th><th rowspan="2">city</th><th colspan="2">product</th></tr>
<tr><th>name</th><th>price</th></tr>
<tr><td>Blake</td><td>London</td><td>Nut</td><td>100</td></tr>
<tr><td></td><td></td><td>Bolt</td><td>120</td></tr>
<tr><td>Smith</td><td>Paris</td><td>Screw</td><td>75</td></tr>
</table>

sname	city	item	price
Blake	London	Nut	100
Blake	London	Bolt	120
Smith	Paris	Screw	75

Dạng chuẩn 2 (2NF)

- **Định nghĩa**: Một sơ đồ quan hệ R được coi là ở dạng chuẩn 2 nếu
  - Sơ đồ quan hệ này ở 1NF
  - Tất cả các thuộc tính không khoá đều phụ thuộc hàm đầy đủ vào khoá chính

  (Lưu ý, A là một thuộc tính khoá nếu A thuộc một khoá tối thiểu nào đó của R. Ngược lại A là thuộc tính không khoá)

Phụ thuộc hàm đầy đủ

- Định nghĩa: Cho sơ đồ quan hệ R(U), F là tập phụ thuộc hàm trên R. X, Y $\subseteq$ U. Y được gọi là phụ thuộc đầy đủ vào X nếu:

  - $X \rightarrow Y$ thuộc F^+
  - không tồn tại $X' \subset X$ $(X' \neq X)$: $X' \rightarrow Y \in F^+$

- Các phụ thuộc hàm không đầy đủ còn gọi là phụ thuộc bộ phận

Ví dụ

- Sales(sid, sname, city, item, price)
- F = {sid→(sname,city), (sid,item)→price}
- Khoá chính (sid,item), ta có sname, city không phụ thuộc hàm đầy đủ vào khoá chính => Quan hệ Sales không thuộc 2NF
- S(sid, sname, city) và Sales (sid, item, price) là quan hệ thuộc 2NF

Dạng chuẩn 3 (3NF)

- Định nghĩa: Một sơ đồ quan hệ R được coi là ở dạng chuẩn 3 nếu
  - Sơ đồ quan hệ này ở 2NF
  - Mọi thuộc tính không khoá đều không phụ thuộc bắc cầu vào khoá chính
- Định nghĩa: Cho sơ đồ quan hệ R(U). F là tập phụ thuộc hàm trên R(U). X,Y, Z $\subseteq$ U. Nói Z là phụ thuộc bắc cầu vào X nếu ta có X$\rightarrow$Y, Y$\rightarrow$Z thuộc F^+. Ngược lại, ta nói Z không phụ thuộc bắc cầu vào X

Ví dụ

- Ví dụ 1: Trong ví dụ tách về dạng chuẩn 2, có: S (sid, sname, city) và Sales(sid, item, price).

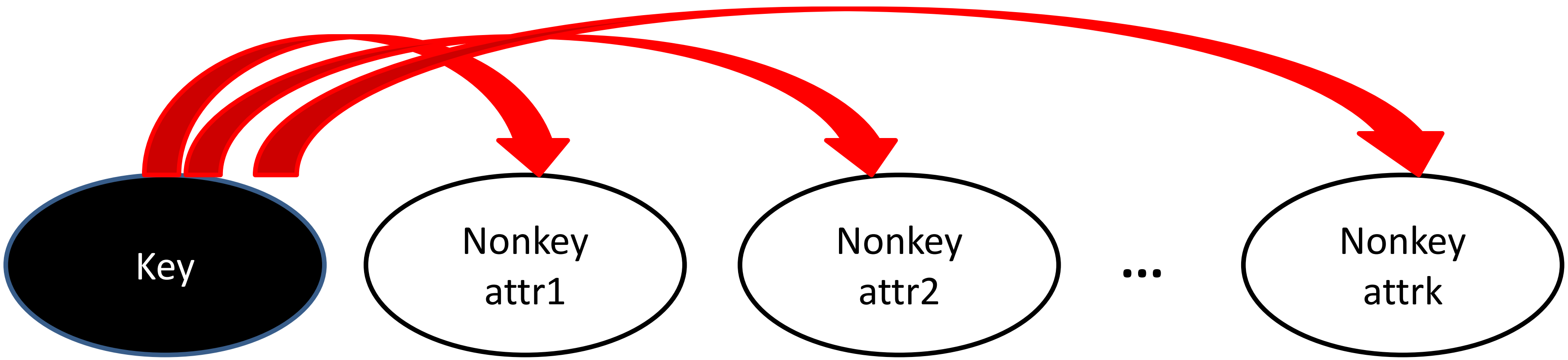
  Xét quan hệ S, pth sid → (sname, city) tồn tại trên S, sid là khoá chính, các thuộc tính không khoá sname, city đều phụ thuộc trực tiếp vào sid. S thuộc 3NF. Tương tự, có Sales cũng thuộc 3NF
- Ví dụ 2:
  - ItemInfo(item, price, discount). F = {item→price, price→discount}. Khoá chính là item, thuộc tính không khoá discount phụ thuộc bắc cầu vào khoá chính item. Vậy quan hệ này không ở 3NF.
  - ItemInfo(item, price) và Discount(price, discount) thuộc 3NF.

Dạng chuẩn Boyce-Codd

- Định nghĩa: Một sơ đồ quan hệ R(U) với một tập phụ thuộc hàm F được gọi là ở dạng chuẩn Boyce-Codd (BCNF) nếu với $\forall$ X→A $\in$ F^+ thì
  - A là thuộc tính xuất hiện trong X <u>*hoặc*</u>
  - X chứa một khoá của quan hệ R.
- Trong một quan hệ ở BCNF, các phụ thuộc hàm không tầm thường duy nhất là một khóa xác định một số thuộc tính. (*không có thuộc tính khóa phụ thuộc vào thuộc tính không khóa*). Do đó, mỗi bộ được xem là một thực thể hoặc liên kết, được nhận diện bởi một khóa và được mô tả bởi các thuộc tính còn lại.

Dạng chuẩn Boyce-Codd (tiếp)

- Hình oval thể hiện thuộc tính/tập thuộc tính
- Các cung thể hiện các phụ thuộc hàm



- Nếu một quan hệ ở BCNF, thì mỗi trường của mỗi bộ chứa thông tin không thể được suy diễn ra từ các giá trị trong các trường khác (mà chỉ sử dụng các phụ thuộc hàm)

Dạng chuẩn Boyce-Codd (tiếp)

- Ví dụ
  - R = {A,B,C} ; F = {AB→C , C→B}.
  - R không phải ở BCNF vì ∃ C→B, C không phải là khoá
- Chú ý:
  - Một quan hệ thuộc 3NF thì chưa chắc đã thuộc BCNF. Nhưng một quan hệ thuộc BCNF thì thuộc 3NF

Tách bảo toàn tập phụ thuộc hàm về 3NF

- **Vào**: R(U), F (giả thiết F là phủ tối thiểu)
- **Ra**: Phép tách bảo toàn tập phụ thuộc hàm về 3NF
- **Thuật toán**

Bước 1. Với các $A_i \in U$, $A_i \notin F$ thì loại A_i khỏi R và lập một quan hệ mới cho các A_i

Bước 2. Nếu $\exists$ $f \in F$, f chứa tất cả các thuộc tính của R (đã bỏ các A_i ở bước trên) thì kết quả là R

Bước 3. Ngược lại, với mỗi $X \rightarrow A \in F$, xác định một quan hệ $R_i(XA)$.

Nếu $\exists$ $X \rightarrow A_i$, $X \rightarrow A_j$ thì tạo một quan hệ chung $R'(XA_iA_j)$

Ví dụ

Cho $R = \{A,B,C,D,E,F,G\}$

$F = \{A \rightarrow B,\ ACD \rightarrow E,\ EF \rightarrow G\}$ (đã tối thiểu)

- Xác định phép tách bảo toàn tập phụ thuộc hàm về 3NF

Bước 1. Không lập được quan hệ nào mới.

Bước 2. $!\exists f \in F$: f chứa tất cả các thuộc tính của R

Bước 3.

$A \rightarrow B$ ⇨ R1(AB)

$ACD \rightarrow E$ ⇨ R2(ACDE)

$EF \rightarrow G$ ⇨ R3(EFG)

Tách không mất mát thông tin và bảo toàn tập phụ thuộc hàm về 3NF

- Yêu cầu:
  - Bảo toàn tập phụ thuộc hàm (như thuật toán trên)
  - Đảm bảo là có một lược đồ con chứa khoá của lược đồ được tách
- Các bước tiến hành

Bước 1. Tìm một khoá tối thiểu của lược đồ quan hệ R đã cho

Bước 2. Tách lược đồ quan hệ R theo phép tách bảo toàn tập phụ thuộc hàm.

Bước 3. Nếu một trong các sơ đồ con có chứa khoá tối thiểu thì kết quả của Bước 2 là kết quả cuối cùng

Ngược lại, thêm vào kết quả đó một sơ đồ quan hệ được tạo bởi khoá tối thiểu tìm được ở Bước 1

Ví dụ

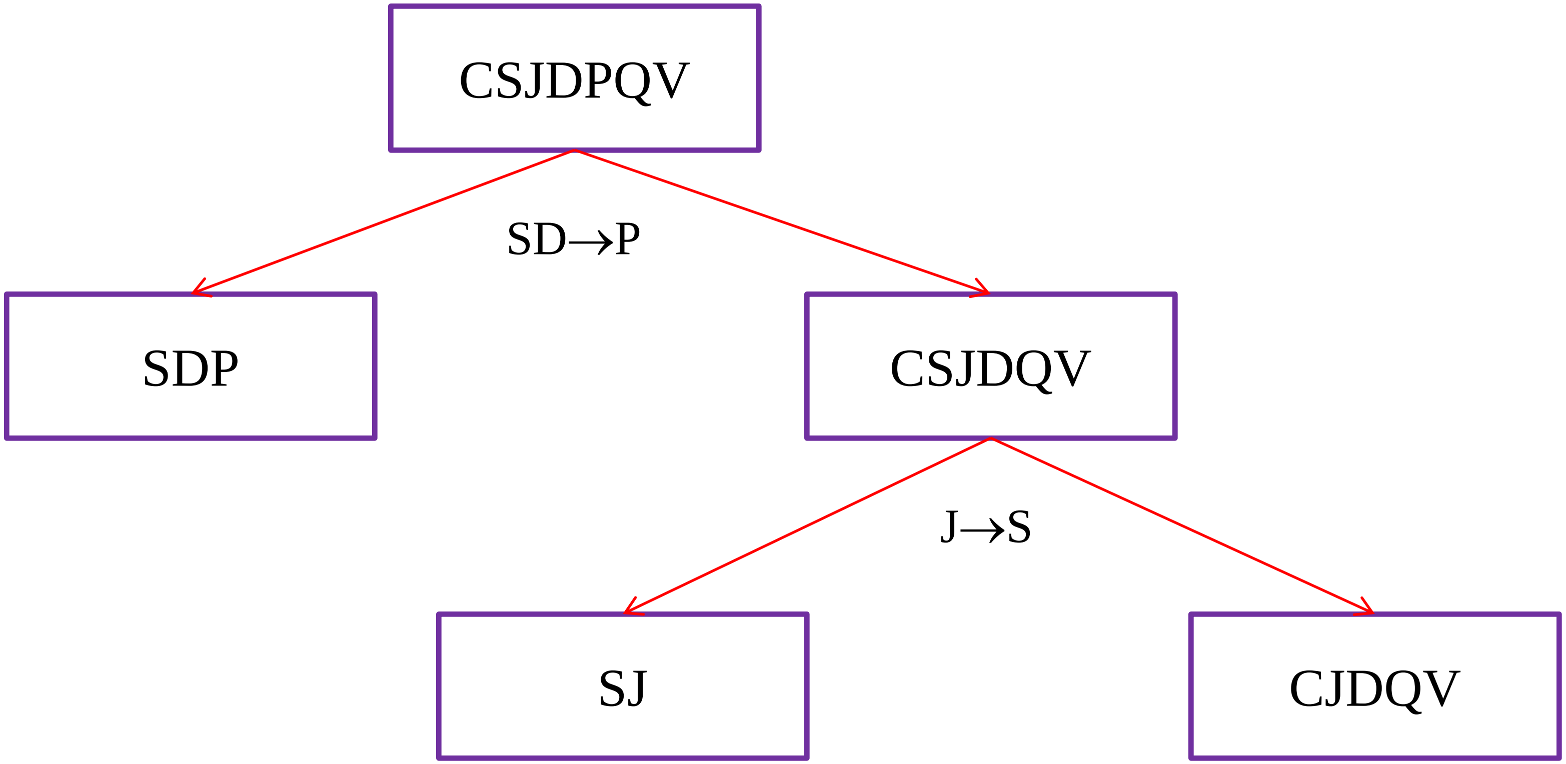
- Cho R(U) trong đó U = {A,B,C,D,E,F,G}.
  F = {A→B, ACD→E, EF→G}
- Tìm một khoá tối thiểu của R:
  K^0 = ABCDEFG
  $K^1 = K^0$ do nếu loại A thì BCDEFG → U không thuộc F^+
  $K^2 = K^1 \setminus \{B\}$ = ACDEFG do ACDEFG → U thuộc F^+
  $K^3 = K^2$ do nếu loại C thì ADEFG → U không thuộc F^+
  $K^4 = K^3$ do nếu loại D thì ACEFG → U không thuộc F^+
  $K^5 = K^4 \setminus \{E\}$ = ACDFG do ACDFG → U thuộc F^+
  $K^6 = K^5$ do nếu loại F thì ACDG → U không thuộc F^+
  $K^7 = K^6 \setminus \{G\}$ = ACDF do ACDF → U thuộc F^+
  Vậy khoá tối thiểu cần tìm là ACDF
- Tách bảo toàn tập phụ thuộc hàm thành 3 sơ đồ con R_1 = AB, R_2= ACDE, R_3 = EFG
- Do khoá ACDF không nằm trong bất kỳ một sơ đồ con nào trong 3 sơ đồ con trên, ta lập một sơ đồ con mới R_4 = ACDF. Kết quả cuối cùng ta có phép tách R thành 4 sơ đồ con {R_1, R_2, R_3, R_4} là một phép tách không mất mát thông tin và bảo toàn tập phụ thuộc hàm

Tách không mất mát thông tin về BCNF

- **Vào**: Sơ đồ quan hệ R, tập phụ thuộc hàm F.
- **Ra**: phép tách không mất mát thông tin bao gồm một tập các sơ đồ con ở BCNF với các phụ thuộc hàm là hình chiếu của F lên sơ đồ đó.
- Cách tiến hành
  – Giả sử R không ở BCNF, $X \subset R$, A là một thuộc tính trong R, $X \rightarrow A$ là một phụ thuộc hàm gây ra vi phạm BCNF. Tách R thành R–A và XA
  – Nếu cả R–A và XA chưa ở BCNF, tiếp tục thực hiện việc tách như trên.

- **Ví dụ**: Contracts(contractid, supplierid, projectid, deptid, partid, qty, value)
  - Viết gọn là CSJDPQV
- Các ràng buộc toàn vẹn:

  C→CSJDPQV, JP→C, SD→P, J→S



Bài tập

Cho sơ đồ S(U), U = {A, B, C, D, E, F, G, H}, tập phụ thuộc hàm F={AB $\rightarrow$CDE, CD$\rightarrow$E, ABC$\rightarrow$FG}

Hãy chuẩn hóa S về dạng chuẩn 3 với phép tách bảo toàn thông tin và phụ thuộc hàm.

Kết luận

- Tầm quan trọng của thiết kế CSDL
  – ảnh hưởng đến chất lượng dữ liệu lưu trữ
  – Hiệu quả của việc khai thác dữ liệu
- Mục đích của thiết kế CSDL:
  – Tránh dư thừa dữ liệu
  – Tránh dị thường dữ liệu khi thêm/xoá/sửa đổi
  – Hiệu quả trong tìm kiếm

➢ Đưa về các dạng chuẩn
  – 2NF: giản ước sự dư thừa để tránh các dị thường khi cập nhật
  – 3NF: tránh các dị thường khi thêm/xoá