

Lý thuyết thiết kế cơ sở dữ liệu quan hệ

Nguyễn Hồng Phương

phuongnh@soict.hut.edu.vn

<http://is.hut.edu.vn/~phuongnh>

Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung

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

2

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

Các vấn đề đối với CSDL VD

- **Dư thừa dữ liệu:** Hãng nào cung ứng nhiều hơn 1 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)
- **Di thường dữ liệu khi thêm:** Nếu có một hãng chứa 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
- **Di thường dữ liệu khi xóa:** Nếu một hãng chỉ cung cấp 1 mặt hàng, nếu ta muốn xóa thông tin về sự cung cấp này thì ta sẽ mất thông tin về hãng cung cấp
- **Di thường dữ liệu khi sửa đổi:** Do thông tin bị lặp lại nên việc sửa đổi 1 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

4

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 chúng ta phải thực hiện kết nối 2 bảng chứ không chỉ là chọn và chiếu trên 1 bảng như ở cách thiết kế trước

5

Mục đích của chuẩn hoá

- Xác định được 1 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 có thể được sử dụng để nhận biết các lược đồ chưa chuẩn và đề xuất hướng cải tiến

6

Phụ thuộc hàm

- Định nghĩa: Cho $R(U)$ là một sơ đồ quan hệ với U là tập thuộc tính $\{A_1, A_2, \dots, A_n\}$. X, Y là tập con của U . Nói rằng X xác định Y hay Y là phụ thuộc hàm vào X ($X \rightarrow Y$) nếu với 1 quan hệ r xác định trên $R(U)$ và 2 bộ bất kỳ t_1, t_2 thuộc r mà $t_1[X] = t_2[X]$ thì ta có $t_1[Y] = t_2[Y]$

7

Ví dụ

- Ví dụ 1:

A	B	C
a1	b1	c1
a2	b2	c2
a3	b1	c1
a4	b3	c2

- $A \rightarrow B, A \rightarrow C, B \rightarrow C$
- 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 ta có $sid \rightarrow sname$, $sid \rightarrow city$, $sid \rightarrow status$

8

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 $X \rightarrow Z$

9

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

10

Ví dụ

- Ví dụ 1:

Cho tập phụ thuộc hàm $\{AB \rightarrow C, C \rightarrow A\}$

Chứng minh: $BC \rightarrow ABC$

$C \rightarrow A$	$BC \rightarrow AB$
$AB \rightarrow C$	$AB \rightarrow ABC$
$BC \rightarrow AB, AB \rightarrow ABC$	$BC \rightarrow ABC$

- Ví dụ 2:

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

Chứng minh: $AB \rightarrow GH$

11

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
- Bao đóng của một tập phụ thuộc hàm có thể rất lớn, sẽ chi phí rất tốn kém cho việc tìm kiếm bao đóng của 1 tập phụ thuộc hàm. Do đó để thuận tiện cho việc kiểm tra xem một phụ thuộc hàm có được suy diễn từ một tập phụ thuộc hàm có sẵn không, người ta có thể sử dụng Bao đóng của 1 tập thuộc tính

12

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

- Định nghĩa: Cho một lược 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 ký hiệu là X^+ là tập tất cả các thuộc tính được xác định hàm bởi X thông qua tập F
 $X^+ = \{A \in U \mid X \rightarrow A \in F^+\}$
- Ta có thể thấy là định nghĩa về bao đóng của một tập thuộc tính dựa trên bao đóng của tập phụ thuộc hàm. Trên thực tế, người ta đưa ra một thuật toán để giúp xác định bao đóng của một tập thuộc tính dễ dàng hơn

13

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 phụ thuộc hàm F trên U
 $X \subseteq U$
- Ra:** X^+
- Thuật toán**
 - B⁰** $X^0 = X$
 - Bⁱ** Tính X^i từ X^{i-1}
 Nếu $\exists Y \rightarrow Z \in F$ 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 Bⁱ
 ngược lại, chuyển Bⁿ
 - Bⁿ** $X^+ = X^i$

14

Ví dụ

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

15

Bổ đề

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

16

Khoá

- Định nghĩa: Cho lược đồ quan hệ $R(U)$, F là một tập các phụ thuộc hàm xác định trên U . K là một tập con của U , K được gọi là khoá tối thiểu của R nếu như
 - $K \rightarrow U$ là một phụ thuộc hàm trong F^+
 - Với mọi tập con thực sự K' của K thì $K' \rightarrow U$ không thuộc F^+
- Với những gì ta đã đề cập trong phần bao đóng ở trên, ta 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 như vậy

17

Thuật toán 2: Tìm khoá tối thiểu

- Vào:** $U = \{A_1, A_2, \dots, A_n\}$, F
- Ra:** khoá tối thiểu K xác định được trên U và F
- Thuật toán**
 - B⁰** $K^0 = U$
 - Bⁱ** Nếu $(K^{i-1} \setminus \{A_i\}) \rightarrow U$ thì $K^i = K^{i-1} \setminus \{A_i\}$
 ngược lại, $K^i = K^{i-1}$
 - Bⁿ⁺¹** $K = K^n$

18

Ví dụ

- Cho $U = \{A, B, C, D, E\}$
- $F = \{AB \rightarrow C, AC \rightarrow B, BC \rightarrow DE\}$. 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 $(K^0 \setminus \{A\}) \rightarrow U$ ($BCDE \rightarrow U$) hay không. Ta cần phải sử dụng thuật toán 1 để kiểm tra điều kiện tương đương là $(BCDE)^+ \subseteq U$ hay 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 $K^2 = K^1 \setminus \{B\} = ACDE$
- $B3: K^3 = ACDE$
- $B4: K^4 = ACE$
- $B5: K^5 = AC$
- Vậy AC là một khoá tối thiểu mà ta cần tìm

19

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?

20

Tập phụ thuộc hàm tương đương

- Định nghĩa: Tập phụ thuộc hàm F là **phủ** của tập phụ thuộc hàm G hay G là **phủ** của F hay F và G **tương đương** nếu $F^+ = G^+$.
– Ký hiệu là $F \approx G$
- Kiểm tra tính tương đương của 2 tập phụ thuộc hàm
 - Với mỗi phụ thuộc hàm $Y \rightarrow Z \in F, Z \subseteq Y^+$ (trên G) thì $Y \rightarrow Z \in G^+$
Nếu với $\forall$ phụ thuộc hàm $f \in F, f \in G^+$ thì $F \subseteq G^+$
 - Tương tự, nếu $\forall$ phụ thuộc hàm $g \in G, g \in F^+$ thì $G^+ \subseteq F^+$
 - Nếu $F^+ \subseteq G^+$ và $G^+ \subseteq F^+$ thì $F \approx G$

21

Ví dụ

- Cho lược đồ quan hệ $R(U)$ với $U = \{A, B, C, D, E, F\}$
 $F = \{AB \rightarrow C, D \rightarrow EF, C \rightarrow BD\}$
 $G = \{AC \rightarrow B, D \rightarrow EF, B \rightarrow CD\}$
Hỏi F và G có phải là 2 tập pth tương đương hay không?
- Thực hiện:
Đối với các phụ thuộc hàm trong F
 - $f_1 = AB \rightarrow C$. AB^+ (đối với G) = $ABCDEF = U$. Vậy f_1 thuộc G^+
 - $f_2 = D \rightarrow EF$ thuộc G nên chắc chắn thuộc G^+
 - $f_3 = C \rightarrow BD$. C^+ (đối với G) = C không chứa BD . Vậy f_3 không thuộc G^+
 - Kết luận F không tương đương với G

22

Tập phụ thuộc hàm không dư thừa

- Đ/N: Tập phụ thuộc hàm F là **không dư thừa** nếu không $\exists X \rightarrow Y \in F$ sao cho $F \setminus \{X \rightarrow Y\} \approx F$.
- Thuật toán 3: Tìm phủ không dư thừa của 1 tập phụ thuộc hàm
 - Vào: Tập thuộc tính $U, F = \{L_i \rightarrow R_i: i = 1..n\}$
 - Ra: Phủ không dư thừa F' của F
 - Thuật toán
 - $B^0: F^0 = F$
 - B^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, $F^i = F^{i-1}$
 - $B^{n+1}: F' = F^n$

23

Phủ tối thiểu của 1 tập phụ thuộc hàm

- Đ/N: F_c được gọi là **phủ tối thiểu** của 1 tập phụ thuộc hàm F nếu thỏa mãn 3 điều kiện sau:
 - Đk1:** Với $\forall f \in F_c, f$ có dạng $X \rightarrow A$, trong đó A là 1 thuộc tính
 - Đk2:** Với $\forall f = X \rightarrow Y \in F_c, \exists A \in X$ (A là 1 thuộc tính):
 $(F_c \setminus f) \cup \{X \setminus A \rightarrow Y\} \approx F_c$
 - Đk3:** $\exists X \rightarrow A \in F_c : F_c \setminus \{X \rightarrow A\} \approx F_c$

24

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 1 thuộc tính bất kỳ thuộc U (thỏa mãn dk1)
 - B.2. Loại bỏ thuộc tính thừa trong vế trái của các phụ thuộc hàm**
Lần lượt giảm ước từng thuộc tính trong vế trái của từng phụ thuộc hàm trong F_1 thu được F_1' . Nếu $F_1' \approx F_1$ thì loại bỏ thuộc tính đang xét.
Khi không có sự giảm ước nào xảy ra nữa ta thu được F_2 thỏa mãn dk2
 - B.3. Loại bỏ phụ thuộc hàm dư thừa**
Lần lượt kiểm tra từng phụ thuộc hàm f . Nếu $F_2 \setminus f \approx F_2$ thì loại bỏ f .
Khi không còn phụ thuộc hàm nào có thể loại bỏ thì thu được F_3 thỏa mãn dk3
 - B.4. $F_c = F_3$**

25

Ví dụ 1

- $U = \{A, B, C\}$
 $F = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$. Tìm phủ tối thiểu của F ?
- $F_1 = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow C\}$
- Xét các pth trong F_1 mà vế trái có nhiều hơn 1 thuộc tính $AB \rightarrow C$. Giảm ước A thì ta còn $B \rightarrow C$ có trong F_1 , vậy A là thuộc tính thừa. Tương tự ta cũng tìm được B là thừa, vậy loại bỏ luôn $AB \rightarrow C$ khỏi F_1 . $F_2 = \{A \rightarrow B, A \rightarrow C, B \rightarrow C\}$
- Bỏ pth thừa: $A \rightarrow C$ là thừa. Vậy $F_c = \{A \rightarrow B, B \rightarrow C\}$

26

Ví dụ 2

- Tìm phủ tối thiểu của tập phụ thuộc hàm
 $F = \{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow G, ACDF \rightarrow EG\}$
- $F_1 = \{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow G, ACDF \rightarrow E, ACDF \rightarrow G\}$
- Loại bỏ thuộc tính thừa** trong 3 phụ thuộc hàm $ABCD \rightarrow E, ACDF \rightarrow E$ và $ACDF \rightarrow G$
Xét $ABCD \rightarrow E$: Giả sử giảm ước A , ta còn $BCD \rightarrow E$, kiểm tra $BCD \rightarrow E$ có được suy ra từ F_1 không, ta tính $(BCD)^+$ (đối với F_1). $(BCD)^+ = BCD$, không chứa E , vậy $BCD \rightarrow E$ không được suy diễn ra từ F , vậy A không phải là thuộc tính thừa trong pth đang xét. B là thừa vì từ F_1 ta có $A \rightarrow B$ dẫn đến $(ACD)^+ = ABCDE$ có chứa E
Làm tương tự ta thấy không có thuộc tính nào là thừa nữa. $F_2 = \{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G, ACDF \rightarrow E, ACDF \rightarrow G\}$

27

Ví dụ 2 (tiếp)

- Loại bỏ pth thừa trong F_2 :** Lần lượt thử loại bỏ 1 pth ra khỏi F_2 , nếu tập pth thu được sau khi loại bỏ vẫn tương đương với F_2 thì pth vừa loại là thừa
 $A \rightarrow B$ không thừa vì nếu loại pth này khỏi F_2 thì từ tập phụ thuộc hàm còn lại A không chứa B
Tương tự, $ACD \rightarrow E, EF \rightarrow G$ không thừa
 $ACDF \rightarrow E$ là phụ thuộc hàm thừa vì nếu loại bỏ pth này, trong tập pth vẫn còn lại $ACD \rightarrow E$, theo tiên đề tăng trưởng ta sẽ suy ra được $ACDF \rightarrow E$
 $ACDF \rightarrow G$ là thừa vì nếu loại bỏ pth này, trong tập pth còn lại vẫn có $ACD \rightarrow E$ và $EF \rightarrow G$, do đó ta vẫn có $(ACDF)^+ = ACDEFG$ có chứa G
- Vậy $F_c = \{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G\}$

28

Phép tách các Sơ đồ quan hệ

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

29

Phép tách không mất mát thông tin

- Đ/N: Cho lược đồ quan hệ $R(U)$ phép tách R thành các sơ đồ con $\{R_1, R_2, \dots, R_k\}$ được gọi là **phép tách không mất mát thông tin** đ/v một tập phụ thuộc hàm F nếu với mọi quan hệ r xác định trên R thỏa mãn F thì:
$$r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) \bowtie \dots \bowtie \Pi_{R_k}(r)$$
- Ví dụ: Phép tách mất mát thông tin**
Supplier(sid, sname, city, NOE, pid, pname, colour, quantity)
 $\rightarrow S1(sid, sname, city, NOE)$ và
 $SP1(pid, pname, colour, quantity)$
- Ví dụ: Phép tách không mất mát thông tin**
 $\rightarrow S1(sid, sname, city, NOE)$ và
 $SP2(sid, pid, pname, colour, quantity)$

30

Định lý tách đôi

- Cho lược đồ quan hệ $R(U)$, tập pth F , phép tách R thành $R_1(U_1)$, $R_2(U_2)$ là một phép tách không mất mát thông tin nếu 1 trong 2 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 lược đồ quan hệ $R(U)$ và phụ thuộc hàm $X \rightarrow Y$ thỏa mãn trên $R(U)$. Phép tách R thành 2 lược đồ con $R_1(U_1)$, $R_2(U_2)$ là một phép tách không mất mát thông tin với:

$$U_1 = XY$$

$$U_2 = XZ$$

$$Z = U \setminus XY$$

31

Thuật toán 5: Kiểm tra tính không mất mát thông tin của 1 phép tách

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

Thuật toán

B.1. Thiết lập một bảng k hàng, n cột

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

Nếu không thì điền b_{ij} .

B.i. Xét $f = X \rightarrow Y \in F$

Nếu $\exists$ 2 hàng t_1, t_2 thuộc bảng : $t_1[X] = t_2[X]$ thì đồng nhất

$t_1[Y] = t_2[Y]$, ưu tiên về giá trị a

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

B.n. Nếu bảng có 1 hàng gồm các kí hiệu $a_1, a_2, \dots, a_n$ thì phép tách là không mất mát thông tin ngược lại, phép tách không bảo toàn thông tin

32

Ví dụ

- $R = ABCD$ được tách thành $R_1=AB$, $R_2=BD$, $R_3=ABC$, $R_4=BCD$. $F = \{A \rightarrow C, B \rightarrow C, CD \rightarrow B, C \rightarrow D\}$
- B.1: Tạo bảng gồm 4 hàng, 4 cột

	A	B	C	D
R1	a1	a2	b31	b41
R2	b12	a2	b32	a4
R3	a1	a2	a3	b43
R4	b14	a2	a3	a4

33

Ví dụ (tiếp)

- B.2 & 3:
- Từ $A \rightarrow C$, ta có

	A	B	C	D
R1	a1	a2	a3	b41
R2	b12	a2	b32	a4
R3	a1	a2	a3	b43
R4	b14	a2	a3	a4

- Từ $B \rightarrow C$, ta có

	A	B	C	D
R1	a1	a2	a3	b41
R2	b12	a2	a3	a4
R3	a1	a2	a3	b43
R4	b14	a2	a3	a4

34

Ví dụ (tiếp)

- Từ $C \rightarrow D$, ta có

	A	B	C	D
R1	a1	a2	a3	a4
R2	b12	a2	a3	a4
R3	a1	a2	a3	a4
R4	b14	a2	a3	a4

- Vậy ta có 2 hàng có toàn các giá trị a . Chứng tỏ phép tách đã cho là không mất mát thông tin

35

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 $\{R_1, R_2, \dots, R_k\}$ 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 $\{R_1, R_2, \dots, R_k\}$ là một phép tách bảo toàn tập phụ thuộc hàm F nếu

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

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 .

36

Ví dụ

- Ví dụ 1:** $R = \{A, B, C\}$, $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ 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:** $R = \{A, B, C\}$, $F = \{AB \rightarrow C, C \rightarrow B\}$ được tách thành $R_1 = AB$, $R_2 = BC$. Phép tách này có bảo toàn tập pth không, có mất mát thông tin không?
- Ví dụ 3:** $R = \{A, B, C, D\}$, $F = \{A \rightarrow B, C \rightarrow D\}$ được tách thành $R_1 = AB$, $R_2 = CD$. Phép tách này có bảo toàn tập pth 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

37

Các dạng chuẩn đối với SDQH

- Quay lại vấn đề thiết kế cơ sở dữ liệu quan hệ, câu hỏi mà chúng ta đặt ra trong quá trình này là Có cần thiết phải tinh chỉnh thiết kế nữa hay không, thực sự thiết kế mà chúng ta có được đã là tốt hay chưa. Để giúp trả lời câu hỏi này, người ta đưa ra các định nghĩa về các dạng chuẩn. Có một vài dạng chuẩn đã được xem xét, khi một quan hệ thuộc vào một dạng chuẩn nào đó thì ta 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 Boye-Code (BCNF)

38

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

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

39

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)

40

Phụ thuộc hàm đầy đủ

- Định nghĩa: Cho lược 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^+
 - $\nexists X' \subset X : X' \rightarrow Y \in F^+$
- Các phụ thuộc hàm không đầy đủ còn gọi là phụ thuộc bộ phận

41

Ví dụ

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

42

Dạng chuẩn 3 (tiếp)

- 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

43

Phụ thuộc bắc cầu

- Định nghĩa: Cho lược đồ quan hệ $R(U)$. F là tập phụ thuộc hàm trên $R(U)$. $X, Y, Z \subseteq U$. Ta 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

44

Ví dụ

- Ví dụ 1: Trong ví dụ tách về dạng chuẩn 2 ta có: $S(sid, sname, city)$ và $Sales(sid, item, price)$. Xét quan hệ S , pth $sid \rightarrow sname, city$ 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 ta có $Sales$ cũng thuộc 3NF
- Ví dụ 2:
 - $ItemInfo(item, price, discount)$. $F = \{item \rightarrow price, price \rightarrow discount\}$. Khoá chính là $item$, thuộc tính không khoá $price, discount$ phụ thuộc bắc cầu vào khoá chính $item$. Vậy quan hệ này không ở 3NF.
 - $ItemInfo(item, price)$ và $Discount(price, discount)$ thuộc 3NF.

45

Dạng chuẩn Boye-Codd

- Định nghĩa: Một sơ đồ quan hệ $R(U)$ với một tập phụ thuộc hàm F được gọi là ở dạng chuẩn Boye-Codd (BCNF) nếu với $\forall X \rightarrow A \in F^+$ thì
 - A là thuộc tính xuất hiện trong X hoặc
 - X chứa một khoá của quan hệ R .
- Ví dụ
 - $R = \{A, B, C\}$; $F = \{AB \rightarrow C, C \rightarrow B\}$.
 - R không phải ở BCNF vì $\exists C \rightarrow B$, 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

46

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**
 - B1.** Với các $A_i \in U$, $A_i \notin F$ thì loại A_i khỏi R và lập 1 quan hệ mới cho các A_i
 - B2.** 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
 - B3.** 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 $R'(XA, A_j)$

47

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
 - B1.** Không lập được quan hệ nào mới.
 - B2.** $\exists f \in F$: f chứa tất cả các thuộc tính của R
 - B3.**

$A \rightarrow B$	$\Rightarrow R_1(AB)$
$ACD \rightarrow E$	$\Rightarrow R_2(ACDE)$
$EF \rightarrow G$	$\Rightarrow R_3(EFG)$

48

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
 - Tìm một khoá tối thiểu của lược đồ quan hệ R đã cho
 - Tách lược đồ quan hệ R theo phép tách bảo toàn tập phụ thuộc hàm.
 - Nếu 1 trong các sơ đồ con có chứa khoá tối thiểu thì kết quả của B2 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 1

49

Ví dụ

- Cho $R(U)$ trong đó $U = \{A, B, C, D, E, F, G\}$. $F = \{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G\}$
- Tìm một khoá tối thiểu của R:
 - $K^0 = ABCDEFG$
 - $K^1 = K^0$ do nếu loại A thì BCDEFG $\rightarrow U$ không thuộc F^+
 - $K^2 = K^1 \setminus \{B\} = ACDEFG$ do ACDEFG $\rightarrow U$ thuộc F^+
 - $K^3 = K^2$ do nếu loại C thì ADEFG $\rightarrow U$ không thuộc F^+
 - $K^4 = K^3$ do nếu loại D thì ACEFG $\rightarrow U$ không thuộc F^+
 - $K^5 = K^4 \setminus \{E\} = ACDFG$ do ACDFG $\rightarrow U$ thuộc F^+
 - $K^6 = K^5$ do nếu loại F thì ACDG $\rightarrow U$ không thuộc F^+
 - $K^7 = K^6 \setminus \{G\} = ACDF$ do ACDF $\rightarrow U$ thuộc F^+
- Vậy khoá tối thiểu cần tìm là ACDF

50

Ví dụ (tiếp)

- Dùng kết quả của ví dụ ở phần tách bảo toàn tập phụ thuộc hàm ta có một phép tách R 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

51

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
 - $KQ = \{R\}$,
 - Với mỗi $S \in KQ$, S không ở BCNF, xét $X \rightarrow A \in S$, với điều kiện X không chứa khoá của S và $A \notin X$. Thay thế S bởi S_1, S_2 với $S_1 = A \cup \{X\}$, $S_2 = \{S\} \setminus A$.
 - Lặp (B2) cho đến khi $\forall S \in KQ$ đều ở BCNF
KQ gồm các sơ đồ con của phép tách yêu cầu

52

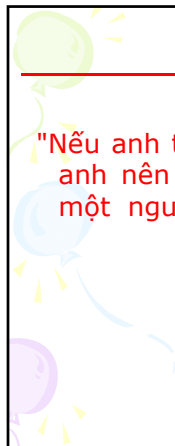
Kết luận

- 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: giảm ước sự dư thừa để tránh các dị thường khi cập nhật
 - 3NF: tránh các dị thường khi thêm/xoá

53



54



Lời hay ý đẹp

"Nếu anh thấy một gia đình hạnh phúc,
anh nên tin rằng ở trong gia đình có
một người đàn bà biết quên mình."

(René Bazin)

55

cuu duong than cong . com

cuu duong than cong . com