

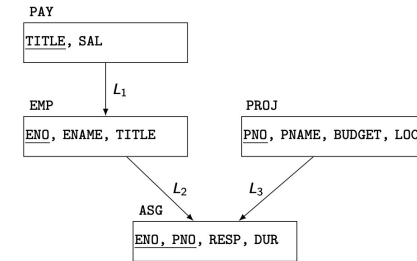
Phân mảnh

- Phân mảnh ngang (HF)
 - Phân mảnh ngang cơ sở (PHF)
 - Phân mảnh ngang dẫn xuất (DHF)
- Phân mảnh dọc (VF)
- Phân mảnh lai (HF)

16

PHF – Thông tin yêu cầu

- Thông tin về CSDL
 - Quan hệ



- Lực lượng của mỗi quan hệ: $card(R)$

17

PHF – Thông tin yêu cầu

- Thông tin về ứng dụng
 - **Vị từ đơn giản**: Cho $R[A_1, A_2, \dots, A_n]$, một vị từ đơn giản p_j là $p_j: A_i \theta Value$ trong đó $\theta \in \{=, <, \leq, >, \geq, \neq\}$, $Value \in D_i$ và D_i là miền xác định của A_i .
Đối với quan hệ R , định nghĩa $Pr = \{p_1, p_2, \dots, p_m\}$
Ví dụ:
 $PNAME = "Maintenance"$
 $BUDGET \leq 200000$
 - **Vị từ hội sơ cấp (minterm predicate)**: Cho R và $Pr = \{p_1, p_2, \dots, p_m\}$ xác định $M = \{m_1, m_2, \dots, m_r\}$ như sau:
 $M = \{ m_i \mid m_i = \bigwedge_{p_j \in Pr} p_j^* \}, 1 \leq i \leq m, 1 \leq j \leq r$
 trong đó $p_j^* = p_j$ hoặc $p_j^* = \neg(p_j)$.

18

PHF – Thông tin yêu cầu

Ví dụ

- $m_1: PNAME = "Maintenance" \wedge BUDGET \leq 200000$
- $m_2: \text{NOT}(PNAME = "Maintenance") \wedge BUDGET \leq 200000$
- $m_3: PNAME = "Maintenance" \wedge \text{NOT}(BUDGET \leq 200000)$
- $m_4: \text{NOT}(PNAME = "Maintenance") \wedge \text{NOT}(BUDGET \leq 200000)$

19

PHF – Thông tin yêu cầu

■ Thông tin về ứng dụng

□ Độ tuyến hội sơ cấp: $sel(m_i)$

- Số lượng các bộ của quan hệ sẽ được truy nhập bởi truy vấn của người dùng được xác định theo một vị từ hội sơ cấp m_i .

□ Tần số truy nhập: $acc(q_i)$

- Tần số mà một ứng dụng người dùng q_i truy nhập dữ liệu.
- Tần số truy nhập cho một vị từ hội sơ cấp, ký hiệu là $acc(m_i)$, cũng có thể được xác định từ tần số truy nhập.

20

Phân mảnh ngang cơ sở

Định nghĩa:

$$R_j = \sigma_{F_j}(R), \quad 1 \leq j \leq w$$

trong đó F_j là một phép toán chọn, là một vị từ hội sơ cấp (tốt nhất).

Vì thế,

Một phân mảnh ngang R_j của quan hệ R bao gồm tất cả các bộ của R thỏa mãn vị từ hội sơ cấp m_i .



Cho một tập các vị từ hội sơ cấp M , số lượng mảnh ngang của quan hệ R bằng số lượng các vị từ hội sơ cấp.

Tập các mảnh ngang còn được gọi là **các mảnh hội sơ cấp**.

21

PHF – Thuật toán

Đầu vào: Một quan hệ R , một tập các vị từ đơn giản Pr

Đầu ra: Tập các mảnh của $R = \{R_1, R_2, \dots, R_w\}$ tuân theo các quy tắc phân mảnh.

Yêu cầu: Đặc tính quan trọng của vị từ đơn giản

- Pr là đầy đủ
- Pr là tối thiểu

22

Tính đầy đủ của các vị từ đơn giản

- Một tập các vị từ đơn giản Pr được gọi là **đầy đủ** khi và chỉ khi xác suất truy nhập bởi mỗi ứng dụng tới một bộ bất kỳ, thuộc về một mảnh hội sơ cấp nào đó được xác định trên Pr , đều như nhau.

■ Ví dụ:

- Giả sử PROJ[PNO,PNAME,BUDGET,LOC] có hai ứng dụng được xác định trên đó.
- Tìm ngân sách của các dự án tại mỗi vị trí. (1)
- Tìm các dự án với ngân sách ít hơn \$200000. (2)

23

Tính đầy đủ của các vị từ đơn giản

Theo (1),

$Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"}\}$

là không đầy đủ đối với (2).

24

Tính đầy đủ của các vị từ đơn giản

Theo (1),

$Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"}\}$

là không đầy đủ đối với (2).

Biến đổi

$Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"},$
 $BUDGET \leq 200000, BUDGET > 200000\}$

là đầy đủ.

25

Tính tối thiểu của vị từ đơn giản

- Nếu một vị từ ảnh hưởng đến cách thực hiện phân mảnh, (nghĩa là, làm cho một mảnh f bị phân mảnh thêm thành hai mảnh, f_i và f_j) thì phải có ít nhất một ứng dụng truy nhập đến f_i và f_j theo các cách khác nhau.
- Nói cách khác, vị từ đơn giản phải có *liên quan (relevant)* trong việc xác định phân mảnh.
- Nếu tất cả các vị từ của tập Pr đều có liên quan thì Pr là *tối thiểu*.

$$\frac{acc(m_i)}{card(f_i)} \neq \frac{acc(m_j)}{card(f_j)}$$

26

Tính tối thiểu của vị từ đơn giản

Ví dụ:

$Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"},$
 $BUDGET \leq 200000, BUDGET > 200000\}$

là tối thiểu (ngoài tính đầy đủ). Tuy nhiên, nếu thêm

$PNAME = \text{"Instrumentation"}$

thì Pr không phải là tối thiểu.

27

Thuật toán COM_MIN

Đầu vào: Một quan hệ R cần phân mảnh ngang cơ sở và một tập vị từ đơn giản Pr

Đầu ra: một tập đầy đủ và tối thiểu vị từ đơn giản Pr' cho Pr

Quy tắc 1: một quan hệ hoặc một mảnh được phân chia thành ít nhất hai phần thì sẽ được truy nhập khác nhau bởi ít nhất một ứng dụng.

28

Thuật toán COM_MIN

1 Khởi tạo:

- Tìm $p_i \in Pr$ sao cho p_i phân chia R theo Quy tắc 1
- Đặt $Pr' = p_i$; $Pr \leftarrow Pr - \{p_i\}$; $F \leftarrow \{f_i\}$

2 Lặp lại thêm các vị từ vào Pr' đến khi đầy đủ

- Tìm $p_j \in Pr$ sao cho p_j phân một mảnh f_k của Pr' theo Quy tắc 1
- Đặt $Pr' = Pr' \cup \{p_j\}$; $Pr \leftarrow Pr - \{p_j\}$; $F \leftarrow F \cup \{f_j\}$
- Nếu $\exists p_k \in Pr'$ là một vị từ không liên quan thì

$$Pr' \leftarrow Pr' - \{p_k\}$$

$$F \leftarrow F - \{f_k\}$$

29

Thuật toán phân mảnh ngang cơ sở - PHORIZONTAL

Sử dụng COM_MIN để thực hiện phân mảnh.

Đầu vào: một quan hệ R và một tập các vị từ đơn giản Pr

Đầu ra: tập các vị từ hội cơ sở M theo đó quan hệ R sẽ được phân mảnh

- 1 $Pr' \leftarrow \text{COM_MIN}(R, Pr)$
- 2 Xác định tập M các vị từ hội cơ sở
- 3 Xác định tập I các phép kéo theo giữa các $p_i \in Pr$
- 4 Loại bỏ các vị từ hội cơ sở mâu thuẫn với I ra khỏi M

30

PHF – Ví dụ

- Hai quan hệ ứng viên: PAY và PROJ.
- Phân mảnh quan hệ PAY
 - Ứng dụng: Kiểm tra thông tin lương và xác định tăng lương.
 - Các bản ghi nhân viên được lưu ở 2 vị trí \Rightarrow ứng dụng chạy ở 2 vị trí: một vị trí xử lý các mẫu tin có lương thấp hơn hoặc bằng 30000\$ và vị trí còn lại xử lý các mẫu tin có lương cao hơn 30000\$.
 - Tập các vị từ đơn giản?
 - Tập các vị từ hội sơ cấp?
 - Các mảnh của quan hệ?

31

PHF – Ví dụ

- Hai quan hệ ứng viên: PAY và PROJ.
- Phân mảnh quan hệ PAY
 - Ứng dụng: Kiểm tra thông tin lương và xác định tăng lương.
 - Các bản ghi nhân viên được lưu ở 2 vị trí \Rightarrow ứng dụng chạy ở 2 vị trí
 - Các vị từ đơn giản
 - p_1 : SAL \leq 30000
 - p_2 : SAL $>$ 30000
 - $Pr = \{p_1, p_2\}$ đầy đủ và tối thiểu $Pr' = Pr$
 - Các vị từ hội cơ sở
 - m_1 : (SAL \leq 30000)
 - m_2 : **NOT**(SAL \leq 30000) = (SAL $>$ 30000)

32

PHF – Ví dụ

PAY₁

TITLE	SAL
Mech. Eng.	27000
Programmer	24000

PAY₂

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000

33

PHF – Ví dụ

- Phân mảnh quan hệ PROJ
 - Các ứng dụng:
 - Tìm tên và ngân sách của các dự án khi biết mã (số) của chúng.
 - Tại 3 vị trí (trạm)
 - Truy nhập thông tin dự án theo ngân sách
 - Một trạm truy nhập \leq 200000, trạm khác truy nhập $>$ 200000
 - Tập các vị từ đơn giản?
 - Tập các vị từ hội sơ cấp?
 - Các mảnh của quan hệ?

34

PHF – Ví dụ

- Phân mảnh quan hệ PROJ
 - Các ứng dụng:
 - Tìm tên và ngân sách của các dự án khi biết mã (số) của chúng.
 - Tại 3 vị trí (trạm)
 - Truy nhập thông tin dự án theo ngân sách
 - Một trạm truy nhập \leq 200000, trạm khác truy nhập $>$ 200000
 - Các vị từ đơn giản
 - Đối với ứng dụng (1)
 - p_1 : LOC = "Montreal"
 - p_2 : LOC = "New York"
 - p_3 : LOC = "Paris"
 - Đối với ứng dụng (2)
 - p_4 : BUDGET \leq 200000
 - p_5 : BUDGET $>$ 200000
 - $Pr = Pr' = \{p_1, p_2, p_3, p_4, p_5\}$

35

PHF – Ví dụ

■ Phân mảnh quan hệ PROJ (tiếp theo)

- Các mảnh hội cơ sở còn lại sau khi loại bỏ

$$m_1 : (\text{LOC} = \text{"Montreal"}) \wedge (\text{BUDGET} \leq 200000)$$

$$m_2 : (\text{LOC} = \text{"Montreal"}) \wedge (\text{BUDGET} > 200000)$$

$$m_3 : (\text{LOC} = \text{"New York"}) \wedge (\text{BUDGET} \leq 200000)$$

$$m_4 : (\text{LOC} = \text{"New York"}) \wedge (\text{BUDGET} > 200000)$$

$$m_5 : (\text{LOC} = \text{"Paris"}) \wedge (\text{BUDGET} \leq 200000)$$

$$m_6 : (\text{LOC} = \text{"Paris"}) \wedge (\text{BUDGET} > 200000)$$

36

PHF – Ví dụ

PROJ₁

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal

PROJ₃

PNO	PNAME	BUDGET	LOC
P2	Database Develop.	135000	New York

PROJ₄

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	255000	New York

PROJ₆

PNO	PNAME	BUDGET	LOC
P4	Maintenance	310000	Paris

37

PHF – Tính đúng đắn

■ Tính đầy đủ

- Do P' là đầy đủ và tối thiểu, nên các vị từ lựa chọn là đầy đủ

■ Tính phục hồi

- Nếu quan hệ R được phân mảnh thành $F_R = \{R_1, R_2, \dots, R_t\}$

$$R = \bigcup_{\forall R_i \in F_R} R_i$$

■ Tính tách biệt

- Các vị từ hội cơ sở tạo ra cơ sở phân mảnh phải loại trừ lẫn nhau.

38