

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

---

# CƠ SỞ DỮ LIỆU PHÂN TÁN

THIẾT KẾ  
CƠ SỞ DỮ LIỆU PHÂN TÁN

# *Nội dung*

---

- ☐ Các vấn đề về phân mảnh dữ liệu
- ☐ Phương pháp phân mảnh ngang
- ☐ Phân mảnh ngang dẫn xuất
- ☐ Phân mảnh dọc
- ☐ Phương pháp phân mảnh hỗn hợp
- ☐ Cấp phát và mô hình cấp phát

## *Phân mảnh ngang*

---

- ❑ Thông tin về CSDL bao gồm tập các quan hệ, tập các thuộc tính và tập các phụ thuộc hàm
- ❑ Thông tin về các ứng dụng gồm các câu truy vấn trên các quan hệ, vị trí các truy vấn....
- ❑ Thông tin về mạng máy tính, cấu trúc, băng thông...
- ❑ Thông tin về hệ thống máy tính, bộ nhớ lưu trữ...

Yêu cầu thông tin về mạng và thông tin về hệ thống máy tính chỉ được sử dụng trong các mô hình cấp phát, không sử dụng trong các thuật toán phân mảnh dữ liệu

# *Phân mảnh dữ liệu*

---

- ❑ Tại sao phải phân mảnh dữ liệu
- ❑ Các kiểu phân mảnh dữ liệu
- ❑ Các quy tắc phân mảnh dữ liệu



## *Phân mảnh dữ liệu*

---

- ❑ Thiết kế CSDLPT gồm: Phân mảnh và cấp phát
- ❑ Phân mảnh là bước đầu tiên trong thiết kế CSDLPT
- ❑ Phân mảnh là quá trình chia một quan hệ toàn cục thành nhiều mảnh có mối quan hệ logic
- ❑ Mảnh là đơn vị truy xuất dữ liệu
  - ✓ Sao cho chi phí thực hiện truy vấn là thấp nhất
  - ✓ Thực hiện nhiều giao dịch đồng thời
  - ✓ Thực hiện song song các câu vấn tin con hoạt tác trên các mảnh.

## *Phân mảnh dữ liệu*

---

- ❑ Khung nhìn của người sử dụng được chọn làm đơn vị truy xuất dữ liệu phân tán là hợp lý
- ❑ Mục dữ liệu của mảnh quan hệ là:
  - ✓ n\_bộ: nếu là phân mảnh ngang
  - ✓ Thuộc tính nếu là phân mảnh dọc

# ***Phân mảnh dữ liệu***

---

## ***Như vậy***

- ❑ Việc phân mảnh một quan hệ thành nhiều mảnh, mỗi mảnh được xử lý như một đơn vị dữ liệu, cho phép thực hiện nhiều giao dịch đồng thời, làm tăng lưu lượng hoạt động của hệ thống.
- ❑ Việc phân mảnh các quan hệ sẽ cho phép thực hiện song song một câu vấn tin bằng cách chia nó ra thành một tập các câu vấn tin con hoạt tác trên các mảnh.

***Tối ưu thời gian, thông lượng, chi phí khi thực hiện truy vấn thông tin làm tăng hiệu suất của hệ thống***

***Quá trình phân mảnh phải được gắn liền với vấn đề cấp phát dữ liệu và các ứng dụng trên nó.***

## *Nhược điểm phân mảnh dữ liệu*

---

- ❑ Những ứng dụng có những yêu cầu “xung đột”, sử dụng độc quyền sẽ ngăn cản quá trình phân mảnh
- ❑ Những ứng dụng có các khung nhìn được định nghĩa trên nhiều mảnh khác nhau sẽ làm giảm hiệu suất hoạt động của hệ thống, làm tăng chi phí truy xuất dữ liệu đến các mảnh và tăng chi phí kết nối các mảnh
- ❑ Kiểm soát ngữ nghĩa và đặc biệt là kiểm tra tính toàn vẹn dữ liệu sẽ khó khăn hơn.



## *Các kiểu phân mảnh*

---

- ❑ Một quan hệ thường được biểu diễn dưới dạng bảng. Việc phân mảnh một quan hệ thành nhiều quan hệ con.
- ❑ Phân mảnh dọc: Các quan hệ được chia theo chiều dọc. Nghĩa là thiết lập một quan hệ mới chỉ có một số thuộc tính từ quan hệ gốc. Thực chất đây là phép chiếu trên tập con các thuộc tính của quan hệ.
- ❑ Phân mảnh ngang: Quan hệ được chia theo chiều ngang. Thực chất là phép chọn quan hệ. Chọn những bộ của quan hệ thỏa mãn một biểu thức điều kiện cho trước.
- ❑ Phân mảnh hỗn hợp.

# *Các kiểu phân mảnh*

---

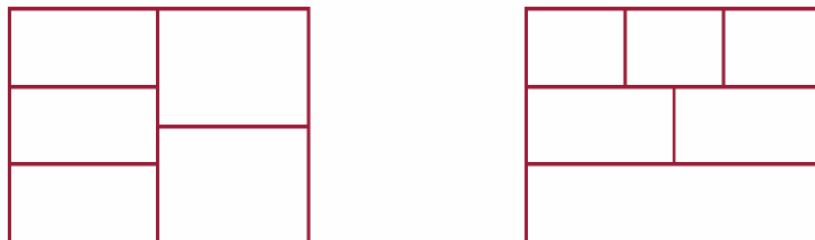
- ❑ Phân mảnh ngang (horizontal fragmentation)



- ❑ Phân mảnh dọc (vertical fragmentation)



- ❑ Phân mảnh hỗn hợp (hibrid fragmentation)



## *Các quy tắc phân mảnh: Yêu cầu*

---

*Các quy tắc đảm bảo cho cơ sở dữ liệu khi phân mảnh sẽ giảm thiểu tổn thất thông tin, mất thông tin hay ít tổn thất thông tin khi thực hiện các truy vấn dữ liệu phân tán*

*Đảm bảo tính không thay đổi về ngữ nghĩa, toàn vẹn dữ liệu, độc lập dữ liệu.*

## *Các quy tắc phân mảnh: Tính đầy đủ*

---

❑ R được phân rã thành các mảnh  $R_1, R_2 \dots R_k$

❑ *Phân mảnh ngang:*

$$\forall r \in R(\Omega) \Rightarrow \exists i \in [1..k] \text{ sao cho } r \in R_i(\Omega)$$

Mục dữ liệu là các n\_bộ

❑ *Phân mảnh dọc:*

$$\forall A \in \Omega \Rightarrow \exists i \in [1..k] \text{ sao cho } A \in \Omega_i$$

Mục dữ liệu là các thuộc tính

❑ Quy tắc này đảm bảo cho các mục dữ liệu trong R được ánh xạ hoàn toàn vào các mảnh và không bị mất.

❑ Phân rã không tổn thất thông tin

## *Các quy tắc phân mảnh: Tính phục hồi*

---

- Nếu  $R$  được phân rã thành các mảnh  $R_1, R_2, \dots, R_k$ . Khi đó:

*Phân mảnh ngang:*

$$R = R_1 \cup R_2 \cup \dots \cup R_k,$$

Nghĩa là quan hệ toàn cục phục hồi lại bằng cách hợp các quan hệ mảnh con.

*Phân mảnh dọc:*

- $R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_k$
- Quan hệ toàn cục bằng kết nối tự nhiên các quan hệ mảnh

- Tính phục hồi đảm bảo quan hệ toàn cục phân rã không tổn thất thông tin.

## *Các quy tắc phân mảnh: Tính tách biệt*

---

- ❑ Nếu R được phân rã thành các mảnh  $R_1, R_2, \dots, R_k$ . Khi đó:

*Phân mảnh ngang:*

$$\forall i \neq j \in [1..k] \Rightarrow R_i \cap R_j = \phi$$

Nghĩa là mỗi một n\_bộ của quan hệ toàn cục được chứa duy nhất trong một quan hệ con.

*Phân mảnh dọc:*

$$\forall i \neq j \in [1..k] \Rightarrow \Omega_i \cap \Omega_j = \{\text{các thuộc tính khóa}\}$$

Các thuộc tính của quan hệ con chỉ chung nhau thuộc tính khóa

- ❑ Quy tắc này đảm bảo các mảnh phân rã rời nhau.

# *Phân mảnh ngang*

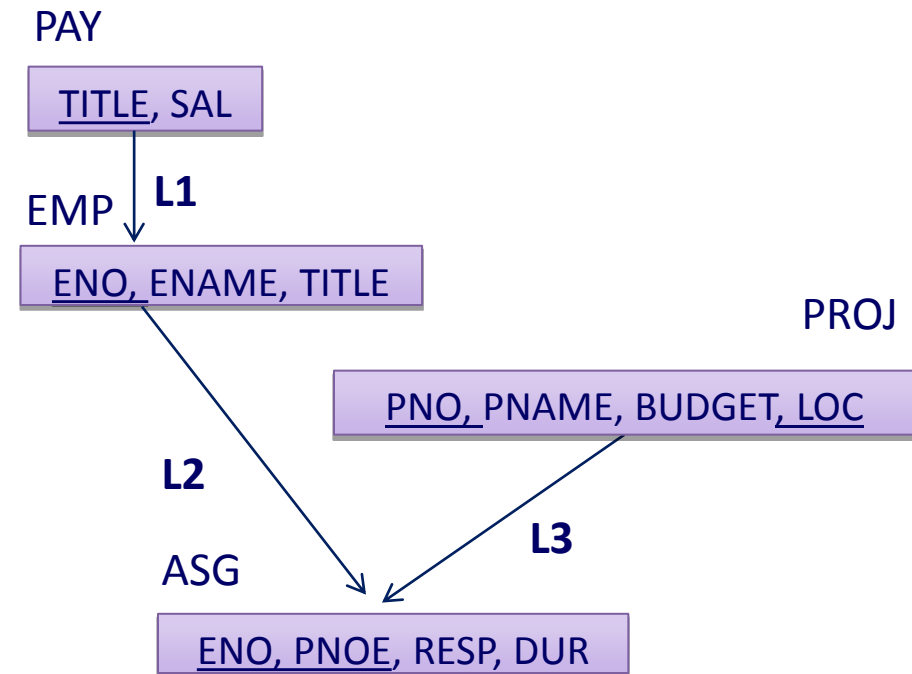
---

- ❑ Phân mảnh ngang là việc tách ngang một quan hệ toàn cục thành nhiều nhiều mảnh. Mỗi một mảnh là một quan hệ khả hợp, chứa một số bộ và các bộ trong các quan hệ con là tách biệt nhau.
- ❑ Phân mảnh ngang thực chất là phép chọn quan hệ thỏa mãn một biểu thức điều kiện cho trước.
- ❑ Có hai phương pháp phân mảnh ngang:
  - *Phân mảnh ngang nguyên thủy*: Là phân mảnh ngang được thực hiện trên các vị từ của chính quan hệ.
  - *Phân mảnh ngang dẫn xuất*: Là phân mảnh một quan hệ dựa trên các vị từ của quan hệ khác.

# ***Phân mảnh ngang: Dựa trên thông tin về CSDL***

- ❑ Thông tin về CSDL: Là thông tin về mối quan hệ một - một, một - nhiều và nhiều - nhiều giữa các bảng, được liên kết bằng các đường nối (Link) có hướng, kết nối bằng
- ❑ Ký hiệu  $|R|$  là  $\text{Card}(R)$ .

- Mỗi quan hệ một - nhiều trở từ các quan hệ PAY đến quan hệ EMP bằng đường nối L1
- Mỗi quan hệ nhiều - nhiều trở từ các quan hệ EMP và PROJ đến quan hệ ASG bằng hai đường nối L2 và L3.





# ***Phân mảnh ngang: Dựa trên thông tin ứng dụng***

---

Thông tin về ứng dụng: Thông tin định tính cơ bản

## ***a) Vị từ đơn giản***

□ Ký hiệu:  $p_j: A_i \theta$  “value”,

Trong đó:

$A_i$  là thuộc tính của  $R(A_1, A_2, \dots, A_n)$ ,

$$\theta \in \{=, <, \neq, \leq, >, \geq\}$$

“Value” là một giá trị  $A_i$

□ Ký hiệu  $Pr$  là tập tất cả các vị từ đơn giản được định nghĩa trên quan hệ  $R$ :  $Pr = \{p_1, p_2, \dots, p_m\}$ .

## ***Phân mảnh ngang: Dựa trên thông tin ứng dụng***

---

- ❑ Ví dụ: Quan hệ dự án PROJ (PNO, PNAME, BUDGET)  
PNO : Mã số dự án PNAME: Tên dự án  
BUDGET Kinh phí dự án.
- ❑ Các vị từ đơn giản trên quan hệ PROJ  
p1: PNAME = “Maintenance”  
p2: BUDGET  $\leq$  200000

PNO	PNAME	BUDGET
P1	Instrumentation (Dụng cụ mỗ)	150000
P2	Database Develop	135000
P3	CAD/CAM	250000
P4	Maintenance (bảo trì)	310000

## ***Phân mảnh ngang: Dựa trên thông tin ứng dụng***

---

### ***b). Vị từ hội sơ cấp***

□  $P_r = \{p_1, p_2, \dots, p_m\}$  là một tập các vị từ đơn giản

$$M = \left\{ m_i \mid m_i = \bigwedge_{p_k \in P_r} p_k^* \right\}$$

□ Trong đó,  $p_k^* = p_k$  hoặc  $p_k^* = \neg p_k$ . Như vậy, vị từ đơn giản xuất hiện trong vị từ hội sơ cấp dưới dạng tự nhiên hoặc dạng phủ định của nó.

## ***Phân mảnh ngang: Dựa trên thông tin ứng dụng***

---

- ❑ Attribute = Value: không có phủ định.
- ❑ Attribute  $\leq$  Value, phủ định là Attribute  $>$  Value
- ❑ Cận\_dưới  $\leq$  Attribute\_1  
phủ định là  $\neg(\text{Cận\_dưới} \leq \text{Attribute\_1})$
- ❑ Attribute\_1  $\leq$  Cận\_trên,  
phủ định là  $\neg(\text{Attribute\_1} \leq \text{Cận\_trên})$
- ❑ Cận\_dưới  $\leq$  Attribute\_1  $\leq$  Cận\_trên,  
phủ định là:  $\neg(\text{Cận\_dưới} \leq \text{Attribute\_1} \leq \text{Cận\_trên})$

# ***Phân mảnh ngang: Dựa trên thông tin ứng dụng***

---

*Ví dụ:*

- $p_1$ : TITLE = “Elect.Eng”
- $p_2$ : TITLE = “Syst. Anal”
- $p_3$ : TITLE = “Mech. Eng”
- $p_4$ : TITLE = “Programmer”
- $p_5$ : SAL  $\leq$  30000
- $p_6$ : SAL  $>$  30000

PAY

TITLE	SAL
Elect.Eng	40000
Mech.Eng	27000
Programmer	24000
Syst.Anal	34000

## ***Phân mảnh ngang: Dựa trên thông tin ứng dụng***

---

$m_1:$      $\text{TITLE} = \text{"Elect.Eng"} \wedge \text{SAL} \leq 30000$

$m_2:$      $\text{TITLE} = \text{"Elect.Eng"} \wedge \text{SAL} > 30000$

$m_3:$      $\neg(\text{TITLE} = \text{"Elect.Eng"}) \wedge \text{SAL} \leq 30000$

$m_4:$      $\neg(\text{TITLE} = \text{"Elect.Eng"}) \wedge \text{SAL} > 30000$

$m_5:$      $\text{TITLE} = \text{"Programmer"} \wedge \text{SAL} \leq 30000$

$m_6:$      $\text{TITLE} = \text{"Programmer"} \wedge \text{SAL} > 30000$

## ***Phân mảnh ngang: Dựa trên thông tin ứng dụng***

---

### ***Ký hiệu***

- ❑ Độ tuyển hội sơ cấp (Minterm Selectivity): số bộ của quan hệ kết quả được chọn theo vị từ hội sơ cấp cho trước. Ký hiệu là  $sel(m)$ . Ví dụ,  $sel(m1)=0$ .  $sel(m2)=1$ .
- ❑ Tần số ứng dụng người dùng truy nhập dữ liệu.  
Nếu  $Q = \{q1, q2, \dots, qk\}$  là tập truy vấn, tần số truy nhập của truy vấn  $qi$  trong một khoảng thời gian đã cho, ký hiệu là  $acc(qi)$
- ❑ Tần số truy nhập hội sơ cấp là tần số truy nhập của hội sơ cấp  $m$ , ký hiệu là  $acc(m)$ .

## *Phân mảnh ngang cơ sở*

---

- ❑ Phân mảnh ngang cơ sở được định nghĩa bằng phép chọn trên quan hệ toàn R:  $f_i = \sigma_{m_i}(R)$   
 $i=1 \dots n$  ; trong đó  $m_i$  là vị từ hội sơ cấp.
- ❑  $f_i$  được gọi là mảnh hội sơ cấp (Minterm Fragment).
- ❑ Một tập M các vị từ hội sơ cấp, số lượng phân mảnh ngang của quan hệ R bằng số lượng các vị từ hội sơ cấp.



## *Phân mảnh ngang cơ sở*

---

Ví dụ: Giả sử tập các vị từ hội sơ cấp:

$$m_1: \{ \text{BUDGET} \leq 200000 \}$$

$$m_2: \{ 200000 < \text{BUDGET} \leq 400000 \}$$

$$m_3: \{ 400000 < \text{BUDGET} \leq 600000 \}$$

$$m_4: \{ 600000 < \text{BUDGET} \}$$

Khi đó quan hệ PROJ được phân rã thành các mảnh ngang như sau:

$$\text{PROJ}_1 = \sigma_{\text{BUDGET} \leq 200000}(\text{PROJ})$$

$$\text{PROJ}_2 = \sigma_{200000 < \text{BUDGET} \leq 400000}(\text{PROJ})$$

$$\text{PROJ}_3 = \sigma_{400000 < \text{BUDGET} \leq 600000}(\text{PROJ})$$

$$\text{PROJ}_4 = \sigma_{600000 < \text{BUDGET}}(\text{PROJ})$$

## *Tính đầy đủ và tính cực tiểu của vị từ đơn giản*

---

Tập vị từ đơn giản có tính đầy đủ (Completeness) và tính cực tiểu (Minimality).

- ❑ Pr là đầy đủ khi và chỉ khi xác suất truy nhập của mỗi ứng dụng đến bộ bất kỳ của mảnh hội sơ cấp bất kỳ được định nghĩa theo Pr là như nhau.
- ❑ Vị từ đầy đủ sẽ đảm bảo cho các mảnh sơ cấp nhất quán về mặt logic. Đồng nhất về mặt thống kê theo cách ứng dụng truy nhập. Vì vậy, một tập vị từ đầy đủ là cơ sở cho việc phân mảnh ngang cơ sở.

## Tính đầy đủ và tính cực tiểu của vị từ đơn giản

---

- Pr là cực tiểu, nếu tất cả các vị từ của nó có tính liên đới.
- Tính liên đới của vị từ đơn giản được định như sau: Gọi  $m_i$  và  $m_j$  là hai vị từ hội sơ cấp:  $p_i \in m_i$ ,  $p_j \in m_j$ ,  $p_j = \neg p_i$   
Gọi  $f_i$  và  $f_j$  là hai mảnh tương ứng được định nghĩa theo  $m_i$  và  $m_j$ . Khi đó  $p_i$  là có liên đới khi và chỉ khi:

$$\frac{\text{acc}(m_i)}{\text{card}(f_i)} \neq \frac{\text{acc}(m_j)}{\text{card}(f_j)} \quad \begin{array}{l} \text{card}(f) \text{ là lực lượng } |f|. \\ \text{acc}(m) \text{ là tần số truy nhập hội sơ cấp } m \end{array}$$

- Nói rằng vị từ đơn giản  $f$  có tính liên đới (Relevant) đến việc xác định một phân mảnh, nếu  $f$  phân mảnh thành các mảnh nhỏ hơn  $f_1$  và  $f_2$ , thì phải có ít nhất một ứng dụng truy nhập đến  $f_1$  và  $f_2$  theo các cách khác nhau.

# *Tính đầy đủ và tính cực tiểu của vị từ đơn giản*

---

*Ví dụ:*

Xét quan hệ PROJ, Tập vị từ đơn giản

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

Các mảnh ngang được định nghĩa dựa vào vị trí các dự án.

Các mảnh như sau:

$$PROJ_1 = \sigma_{LOC = \text{"Montreal"}}(PROJ);$$

$$PROJ_2 = \sigma_{LOC = \text{"New York"}}(PROJ);$$

$$PROJ_3 = \sigma_{LOC = \text{"Paris"}}(PROJ)$$

## *Tính đầy đủ và tính cực tiểu của vị từ đơn giản*

---

*Tính đầy đủ:*

- ❑  $Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"}\}$  là đầy đủ, vì các ứng dụng truy nhập vào PROJ theo vị trí, mỗi bộ của các mảnh  $PROJ_i$ ,  $i=1,2,3$ , đều có xác suất được truy nhập bằng nhau.
- ❑ Nếu  $Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"}, BUDGET > 200000\}$ , thì Pr sẽ là không đầy đủ, vì một số bộ trong các mảnh  $PROJ_i$  có xác suất được truy nhập lớn hơn.
- ❑  $Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"}, BUDGET \leq 200000, BUDGET > 200000\}$  là tập vị từ đầy đủ.

# *Tính đầy đủ và tính cực tiểu của vị từ đơn giản*

---

## *Tính cực tiểu*

- ❑  $Pr = \{LOC = \text{"Montreal"}, LOC = \text{"New York"}, LOC = \text{"Paris"}, BUDGET \leq 200000, BUDGET > 200000\}$  là cực tiểu
- ❑ Nếu thêm vị từ  $PNAME = \text{"Instrumentation"}$  và tập  $Pr$ , khi đó sẽ không đảm bảo tính cực tiểu, vì vị từ thêm vào không có tính liên đới ứng với  $Pr$ . Không một ứng dụng nào truy xuất khác nhau đến các mảnh được tạo ra.

## *Thuật toán xác định tập vị từ đầy đủ và cực tiểu*

---

- ❑ Thuật toán COM\_MIN tạo ra một tập vị từ đầy đủ và cực tiểu các vị  $Pr'$  từ một tập các vị từ đơn giản  $Pr$  cho trước theo quy tắc:
- ❑ Một quan hệ hoặc một mảnh “được phân hoạch thành ít nhất hai phần và chúng được truy nhập khác nhau bởi ít nhất một ứng dụng”.
- ❑ Mảnh  $f_i$  được phân hoạch theo vị từ sơ cấp  $p_i \in Pr'$ , qui ước là  $f_i$  của  $Pr'$

# Thuật toán COM\_MIN

Input:  $R$  là quan hệ và  $Pr$  là tập các vị từ đơn giản

Output:  $Pr'$  là tập các vị từ đơn giản.

Khai báo:  $F$  là tập các mảnh hội sơ cấp.

Begin

    Tìm một vị từ  $p_i \in Pr$  sao cho  $p_i$  phân hoạch  $R$  theo qui tắc

$Pr' \leftarrow p_i$

$Pr \leftarrow Pr - p_i$

$F \leftarrow f_i$              $\{f_i \text{ là mảnh hội sơ cấp theo } p_i\}$

    do

        begin

            Tìm một  $p_j \in Pr$  sao cho  $p_j$  phân hoạch một mảnh  $f_k$  của  $Pr'$  theo qui tắc

$Pr' \leftarrow Pr' \cup p_j$

$Pr \leftarrow Pr - p_j$

$F \leftarrow F \cup f_i$

            If  $\exists p_k \in Pr'$ , một vị từ không có liên đới then

                begin

$Pr' \leftarrow Pr' - p_k$

$F \leftarrow F - p_k$

                end-if

        end-begin

    until  $Pr$  đầy đủ

End. { COM\_MIN }



## *Một số nhận xét về phân mảnh ngang nguyên thủy*

---

- ❑ Tìm tập các vị từ đơn giản đầy đủ và cực tiểu là bước đầu tiên trong thiết kế phân mảnh ngang nguyên thủy
- ❑ Thuật toán bắt đầu bằng cách tìm một vị từ có liên đới và phân hoạch quan hệ đã cho. Vòng lặp do-until thêm các vị từ vào  $Pr'$ , đảm bảo  $Pr'$  có tính cực tiểu tại mỗi bước.
- ❑ Bước tiếp theo của thiết kế phân mảnh ngang nguyên thủy là tìm tập các vị từ hội sơ cấp được định nghĩa trên các vị từ trong  $Pr'$ . Các vị từ hội sơ cấp xác định các mảnh cho cấp phát.

## ***Một số nhận xét về phân mảnh ngang nguyên thủy***

---

- ❑ Tuy nhiên, các vị từ hội sơ cấp có thể rất lớn, tỷ lệ hàm mũ theo số lượng các vị từ đơn giản. Vì vậy cần phải loại bỏ những mảnh không có ý nghĩa, bằng cách xác định những vị từ mâu thuẫn với tập các phép kéo theo (Implication).
- ❑ Thuật toán ***PHORIZONTAL*** tìm cách làm giảm số lượng vị từ hội sơ cấp cần được định nghĩa trong phân mảnh, bằng cách loại bỏ một số mảnh vô nghĩa. Điều này được thực hiện bằng cách xác định những vị từ mâu thuẫn với tập các phép kéo theo I.

# *Một số nhận xét về phân mảnh ngang nguyên thủy*

---

- ❑ Nếu  $Pr' = \{p_1, p_2\}$ , trong đó:
  - $p_1$ :  $att = value\_1$
  - $p_2$ :  $att = value\_2$
- ❑ I chứa hai phép kéo theo với khẳng định:
  - $i_1$ :  $(att=value\_1) \Rightarrow \neg(att=value\_2)$
  - $i_2$ :  $\neg(att=value\_1) \Rightarrow (att=value\_2)$

## *Một số nhận xét về phân mảnh ngang nguyên thủy*

---

- ❑ Bốn vị từ hội sơ cấp sau đây được tính theo Pr:

$$m_1: (\text{att}=\text{value\_1}) \wedge (\text{att}=\text{value\_2})$$

$$m_2: (\text{att}=\text{value\_1}) \wedge \neg(\text{att}=\text{value\_2})$$

$$m_3: \neg(\text{att}=\text{value\_1}) \wedge (\text{att}=\text{value\_2})$$

$$m_4: \neg(\text{att}=\text{value\_1}) \wedge \neg(\text{att}=\text{value\_2})$$

- ❑ Các vị từ hội sơ cấp  $m_1$  và  $m_4$  mâu thuẫn với các phép kéo theo I và vì thế bị loại ra khỏi M. Để thực hiện bước này ta sẽ áp dụng thuật toán phân mảnh ngang nguyên thủy.

## *Thuật toán PHORIZONTAL*

---

Input: R là quan hệ cần phân mảnh ngang cơ sở.

Pr là tập các vị từ đơn giản

Output: M là tập các vị từ hội sơ cấp

Begin

Pr'  $\leftarrow$  COM\_MIN(R, Pr)

Xác định tập M các vị từ hội sơ cấp

Xác định tập I các phép kéo theo giữa các  $p_i \in \text{Pr}'$

For mỗi  $m_i \in M$  do

    If  $m_i$  mâu thuẫn với I then

        M  $\leftarrow$  M -  $m_i$

    End\_if

End\_for

End. {PHORIZONTAL}

## *Ví dụ về phân mảnh ngang nguyên thủy - PAY*

---

- ❑ Giả sử có một ứng dụng kiểm tra thông tin lương và xác định số lương sẽ tăng trên quan hệ PAY.
- ❑ Giả sử có hai vị trí,
  - Vị trí thứ nhất:  $SAL \leq 30000$
  - Vị trí thứ hai:  $SAL > 30000$
- ❑ Câu truy vấn sẽ thực hiện trên cả hai vị trí. Tập vị từ đơn giản sử dụng để phân hoạch quan hệ PAY là:
  - $p_1: SAL \leq 30000$
  - $p_2: SAL > 30000$

## *Ví dụ về phân mảnh ngang nguyên thủy – PAY*

---

- ❑ Từ tập vị từ đơn giản khởi đầu là  $Pr = \{p1, p2\}$ .
- ❑ Áp dụng COM\_MIN với  $i=1$  làm giá trị khởi đầu tạo ra  $Pr'=\{p1\}$  là đầy đủ và cực tiểu vì  $p2$  không phân hoạch  $f_1$  (là mảnh hội sơ cấp được tạo ra ứng với  $p1$ ) theo quy tắc.
- ❑ Các vị từ hội sơ cấp sau đây là các phần tử của M:  
     $m_1: (SAL \leq 30000)$   
     $m_2: \neg (SAL \leq 30000) = SAL > 30000$
- ❑ Khi đó, hai mảnh  $F = \{PAY_1, PAY_2\}$  theo M là:

PAY <sub>1</sub>	
TITLE	SAL
Mech. Eng.	27000
Programmer	24000

PAY <sub>2</sub>	
TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000

## *Ví dụ về phân mảnh ngang nguyên thủy – PROJ*

---

Giả sử có hai ứng dụng trên quan hệ PROJ

- Ứng dụng 1: Tên và kinh phí của dự án trên ba vị trí.

```
SELECT      PNAME, BUDGET
FROM        PROJ
WHERE       LOC = Value
```

- Các vị từ đơn giản sử dụng cho ứng dụng này là:

$p_1$ : LOC = "Montreal"  
 $p_2$ : LOC = "New York"  
 $p_3$ : LOC = "Paris"

.



## *Ví dụ về phân mảnh ngang nguyên thủy – PROJ*

---

- ❑ Ứng dụng 2: Liên quan đến các dự án có kinh phí nhỏ hơn hoặc bằng 200000\$ được quản lý tại một vị trí và các dự án có kinh phí lớn hơn 200000 được quản lý tại vị trí thứ hai. Vì vậy, các vị từ đơn giản được sử dụng để phân mảnh ứng dụng thứ hai là:

$p_4:$  BUDGET  $\leq$  200000

$p_5:$  BUDGET  $>$  200000

- ❑ Sử dụng thuật toán COM\_MIN kiểm tra tập  $Pr' = \{p_1, p_2, p_3, p_4, p_5\}$  là đầy đủ và cực tiểu.

## *Ví dụ về phân mảnh ngang nguyên thủy – PROJ*

---

- ❑ Tập M các vị từ hội sơ cấp tạo ra M dựa trên Pr' như sau:

$m_1: (LOC="Montreal") \wedge (BUDGET \leq 20000)$

$m_2: (LOC="Montreal") \wedge (BUDGET > 20000)$

$m_3: (LOC="New York") \wedge (BUDGET \leq 20000)$

$m_4: (LOC="New York") \wedge (BUDGET > 20000)$

$m_5: (LOC="Paris") \wedge (BUDGET \leq 20000)$

$m_6: (LOC="Paris") \wedge (BUDGET > 20000)$

- ❑ Kết quả phân mảnh ngang cơ sở PROJ tạo ra sáu mảnh  $FPROJ = \{PROJ_1, PROJ_2, PROJ_3, PROJ_4, PROJ_5, PROJ_6\}$  theo các vị từ hội sơ cấp M.

## *Ví dụ về phân mảnh ngang nguyên thủy – PROJ*

---

- ❑ Các mảnh PROJ<sub>2</sub>, PROJ<sub>5</sub> rỗng.

PROJ<sub>1</sub>

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal

PROJ<sub>3</sub>

PNO	PNAME	BUDGET	LOC
P2	Database Develop.	135000	New York

PROJ<sub>4</sub>

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	250000	New York

PROJ<sub>6</sub>

PNO	PNAME	BUDGET	LOC
P4	Maintenance	31000	Paris

# *Phân mảnh ngang dẫn xuất*

---

Phân mảnh ngang dẫn xuất là việc phân mảnh một quan hệ theo kết nối bằng nhau (Equijoin) hoặc kết nối nửa bằng nhau (Semijoin) đến các quan hệ khác trong cơ sở dữ liệu. Việc quyết định chọn phân mảnh nào tối ưu hơn cần dựa trên hai tiêu chuẩn sau:

1. Phân mảnh có đặc tính kết nối tốt hơn
2. Phân mảnh sử dụng cho nhiều ứng dụng hơn

## *Phân mảnh ngang dẫn xuất*

---

Ví dụ: Phân mảnh dẫn xuất quan hệ EMP theo quan hệ PAY.  
Nhóm người tham gia dự án thành hai nhóm theo lương

$$SAL \leq 30000 \quad \text{và} \quad SAL > 30000$$

$$PAY_1 = \sigma_{SAL \leq 30000}(PAY)$$

$$PAY_2 = \sigma_{SAL > 30000}(PAY)$$

$$EMP_1 = EMP \triangleright \leq PAY_1$$

$$EMP_2 = EMP \triangleright \leq PAY_2$$

# *Phân mảnh ngang dẫn xuất*

---

Ví dụ: Phân mảnh dẫn xuất quan hệ ASG theo quan hệ PROJ và EMP

Xét quan hệ ASG có hai ứng dụng trên nó:

*Ứng dụng 1:* Danh sách các kỹ sư làm việc trong các dự án tại chỗ. Ứng dụng này thực hiện trên các vị trí (thí dụ ba vị trí)

$$\text{PROJ}_1 = \sigma_{\text{LOC}=\text{"Montreal"}}(\text{PROJ})$$

$$\text{PROJ}_2 = \sigma_{\text{LOC}=\text{"New York"}}(\text{PROJ})$$

$$\text{PROJ}_3 = \sigma_{\text{LOC}=\text{"Paris"}}(\text{PROJ})$$

## *Phân mảnh ngang dẫn xuất*

---

Phân mảnh dẫn xuất ASG theo các mảnh  $PROJ_1$ ,  $PROJ_2$  và  $PROJ_3$  như sau:

$$ASG_1 = ASG \bowtie_{\leq} PROJ_1$$

$$ASG_2 = ASG \bowtie_{\leq} PROJ_2$$

$$ASG_3 = ASG \bowtie_{\leq} PROJ_3$$

*Ứng dụng 2:* Tại 2 vị trí quản lý nhân viên, yêu cầu thông tin về kinh phí và thời gian thực hiện các dự án của các nhân viên.

$$EMP_1 = \sigma_{TITLE = \text{“ Syst. Anal”}}(EMP)$$

$$EMP_2 = \sigma_{TITLE = \text{“ Programmer”}}(EMP)$$

## *Phân mảnh ngang dẫn xuất*

---

Phân mảnh dẫn xuất ASG theo  $EMP_1$  và  $EMP_2$  như sau:

$$ASG_{11} = ASG_1 \triangleright \leq EMP_1$$

$$ASG_{12} = ASG_1 \triangleright \leq EMP_2$$

$$ASG_{21} = ASG_2 \triangleright \leq EMP_1$$

$$ASG_{22} = ASG_2 \triangleright \leq EMP_2$$

$$ASG_{31} = ASG_3 \triangleright \leq EMP_1$$

$$ASG_{32} = ASG_3 \triangleright \leq EMP_2$$



# *Phân mảnh ngang dẫn xuất*

---

## Nhận xét

- ❑ Phân mảnh dẫn xuất có thể xảy ra dây chuyền, trong đó một quan hệ được phân mảnh như là hệ quả của một phân mảnh cho một quan hệ khác, và đến lượt nó lại làm cho các quan hệ khác phải phân mảnh (như dây chuyền PAY-EMP-ASG).
- ❑ Một quan hệ có thể có nhiều cách phân mảnh. Chọn lựa một lược đồ phân mảnh nào cho tối ưu phụ thuộc vào ứng dụng và cấp phát.

# KIỂM TRA

---

Cho quan hệ QLSV(MA, HT, QQ,NS,GT, DT, TB)

Trong đó MA: Mã sinh viên; HT: Họ và tên sinh viên, QQ: Quê quán; NS: Năm sinh; GT: Giới tính; DT: Dân tộc; TB: Điểm trung bình.

- 1) Tập vị từ đơn giản có tính đầy đủ và tính cực tiểu
- 2) Tập các vị từ hội sơ cấp M
- 3) Phân mảnh ngang cơ sở trên tập các vị từ hội sơ cấp