

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

Thông tin cho thiết kế

- 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 mảng quan hệ là:
 - ✓ n_bộ: nếu là phân mảng ngang
 - ✓ Thuộc tính nếu là phân mảng 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 cách 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 ngũ 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

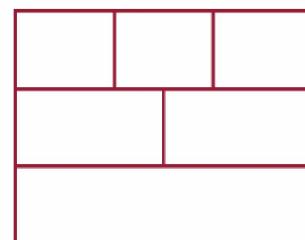
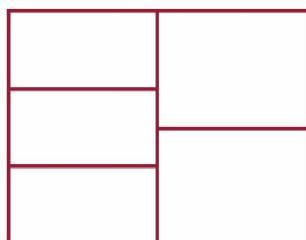
- Phân mảnh ngang (horizontal fragmentation)



- Phân mảnh dọc (vertical fragmentation)



- Phân mảnh hỗn hợp (hybrid 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:

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

Nghĩa là mọi 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 chí chung nhau thuộc tính khóa

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

- Thực hiện phân mảnh quan hệ $R := (\underline{A}_1, A_2, A_3, A_4, A_5, A_6)$ thành tập các mảnh sau: $R_1 := (\underline{A}_1, A_5, A_6); R_2 := (\underline{A}_1, A_3, A_4); R_3 := (\underline{A}_1, A_6)$
- Kiểm tra tính đúng đắn của phép phân mảnh.

Thông tin cho Phân tán

- Thông tin về CSDL bao gồm tập các quan hệ, mối 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

Thông tin cần thiết của phân mảnh ngang

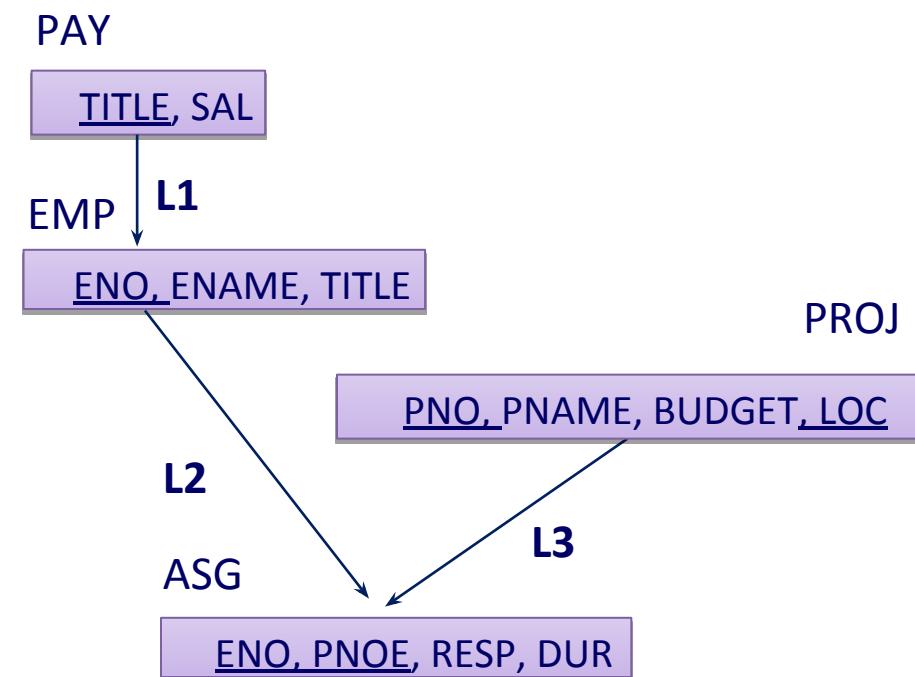
- *Thông tin về CSDL*
- *Thông tin về ứng dụng*

Phân mảng ngang

- Phân phân mảng ngang nguyên thủy: Dựa trên các vị từ của chính qh muốn phân mảng
- Phân mảng dẫn xuất: Dựa trên vị từ của 1 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. Đg liên kết bằng các đường nối (Link) có hướng chỉ 1- nhiều, kết nối bằng
- Ký hiệu $|R|$ là Card(R).
 - Mối quan hệ một - nhiều trả từ các quan hệ PAY đến quan hệ EMP bằng đường nối L1
 - Mối quan hệ nhiều - nhiều trả 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 P_r là tập tất cả các vị từ đơn giản được định nghĩa trên quan hệ R : $P_r = \{p_1, p_2, \dots, p_m\}$.

Phân mảng ngang: Dựa trên thông tin ứng dụng

- Ví dụ: Quan hệ dự án PROJ (PNO, PNAME, BUGET)
PNO : Mã số dự án PNAME: Tên dự án
BUGET Kinh phí dự án.
- Các ví dụ đơn giản trên quan hệ PROJ
 - p1: PNAME = “Maintenance”
 - p2: BUGET ≤ 200000

PNO	PNAME	BUGET
P1	Instrumentation (Dụng cụ mỏ)	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

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

- m_i được gọi là vị từ hội sơ cấp i Trong đó, $p_k^* = p_k$ hoặc . Như $p_k^* = \neg p_k$ 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 mảng ngang: Dựa trên thông tin ứng dụng

Ví dụ:

- p₁: TITLE = “Elect.Eng”
- p₂: TITLE = “Syst. Anal”
- p₃: TITLE = “Mech. Eng”
- p₄: TITLE = “Programmer”
- p₅: SAL ≤ 30000
- p₆: 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

Ký hiệu

- Độ tuyển hội sơ cấp (Minterm Selectivity): số bộ của quan hệ kết quả được chọn theo vị trí hội sơ cấp cho trước. Ký hiệu là $sel(m)$. Ví dụ, $sel(m_1)=0$. $sel(m_2)=1$.
- Tần số ứng dụng người dùng truy nhập dữ liệu.
Nếu $Q = \{q_1, q_2, \dots, q_k\}$ 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 mi , ký hiệu là $acc(mi)$.

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.

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 nhau 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 nghĩa 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:

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

card(f) là lực lượng $|f|$.

acc(mj) là tần số truy nhập hội sơ cấp mj

- 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.

Minimality

Liên đới (Relevant):

Đặt m_i và m_j là các vị từ hội xác định từ các vị từ $P1, p2, p3$

$$m_i = p_1 \wedge p_2 \wedge p_3 \quad \xrightarrow{\text{fragment } f_i}$$

$$m_j = p_1 \wedge \neg p_2 \wedge p_3 \quad \xrightarrow{\text{fragment } f_j}$$

p_2 là liên đới nếu và chỉ nếu

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

Access frequency
Cardinality

Tức là, phải có ít nhất 1 ứng dụng truy nhập khác nhau vào f_i và f_j
i.e., Vị từ đơn giản p_i phải liên đới trong việc xác định 1 mảnh

Cực tiểu: Nếu tất cả các vị từ của tập Pr là liên đới

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris

PAY

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

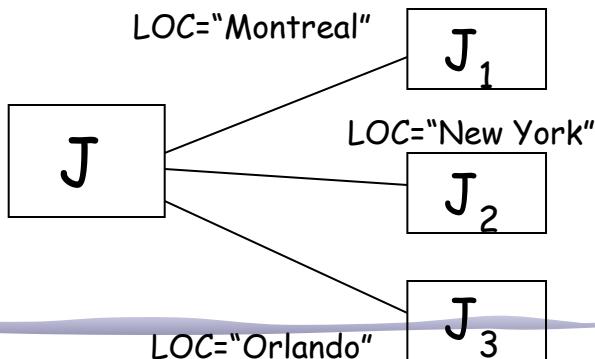
Completeness (4)

JNO	JNAME	BUDGET	LOC
J1	Instrumental	150,000	Montreal
J2	Database Dev.	135,000	New York
J3	CAD/CAM	250,000	New York
J4	Maintenance	350,000	Orlando

$$J_1 = \sigma_{LOC="MONTREAL"}(J)$$

$$J_2 = \sigma_{LOC="New York"}(J)$$

$$J_3 = \sigma_{LOC="Orlando"}(J)$$



Case 1: The only application that accesses J wants to access the tuples according to the location.

The set of simple predicates

$$Pr = \begin{cases} LOC="Montreal", \\ LOC="New York", \\ LOC="Orlando" \end{cases}$$

Là đầy đủ vì mỗi bộ trong 1 mảng dduwwocj phân ra có xác suất truy nhập như sau

Completeness (5)

Example:

$Pr =$ LOC="Montreal",
LOC="New York",
LOC="Orlando"

J1

JNO	JNAME	BUDGET	LOC
001	Instrumental	150,000	Montreal

J2

JNO	JNAME	BUDGET	LOC
004	GUI	135,000	New York
007	CAD/CAM	250,000	New York

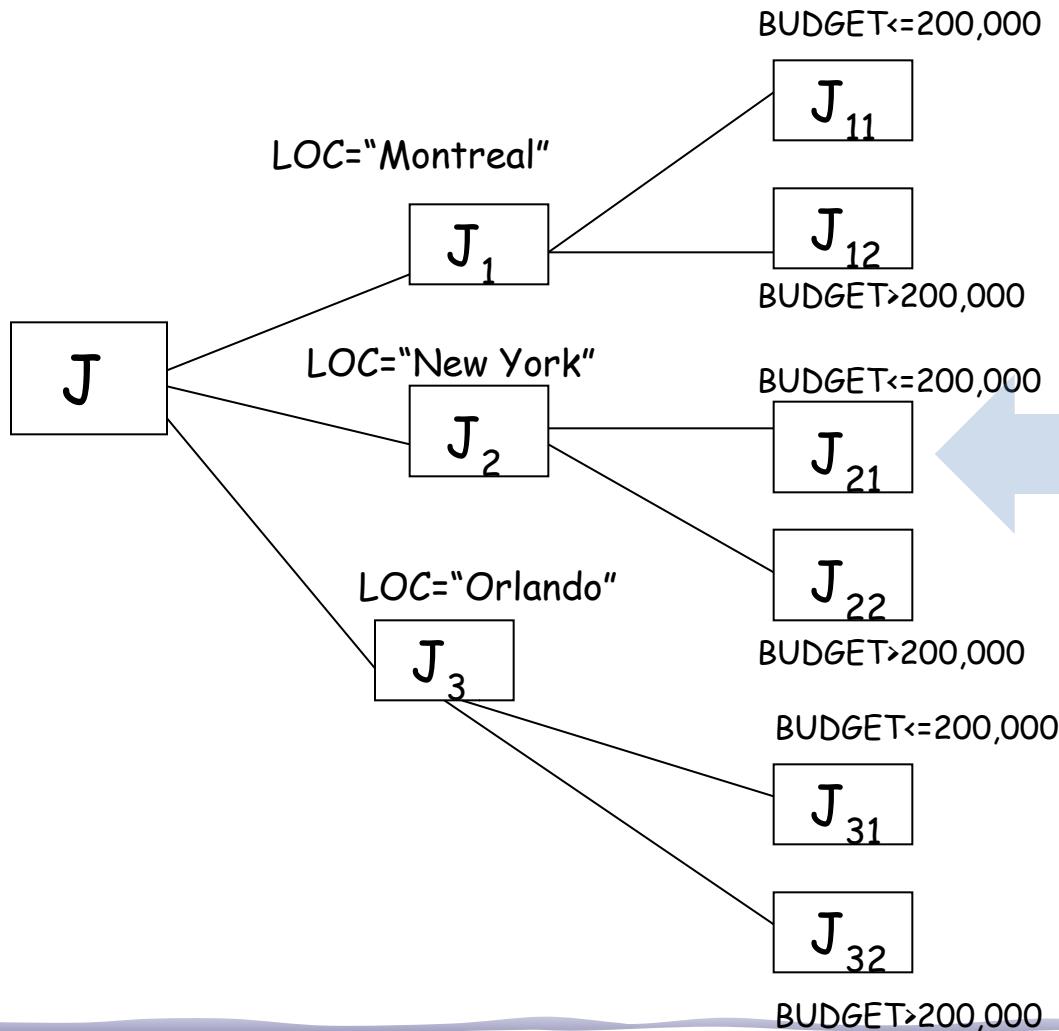
J3

JNO	JNAME	BUDGET	LOC
003	Database Dev.	310,000	Orlando

Case 2: There is a second application which accesses only those project tuples where the budget is less than \$200,000.

- Since tuple "004" is accessed more frequently than tuple "007", Pr is not complete.
- To make the set complete, we need to add (BUDGET < 200,000) to Pr .

Completeness (6)



Note: Completeness is a desirable property because a complete set defines fragments that are not only **logically uniform** in that they all satisfy the minterm predicate, but **statistically homogeneous**.

Vd về Complete and Minimal

2 ứng dụng:

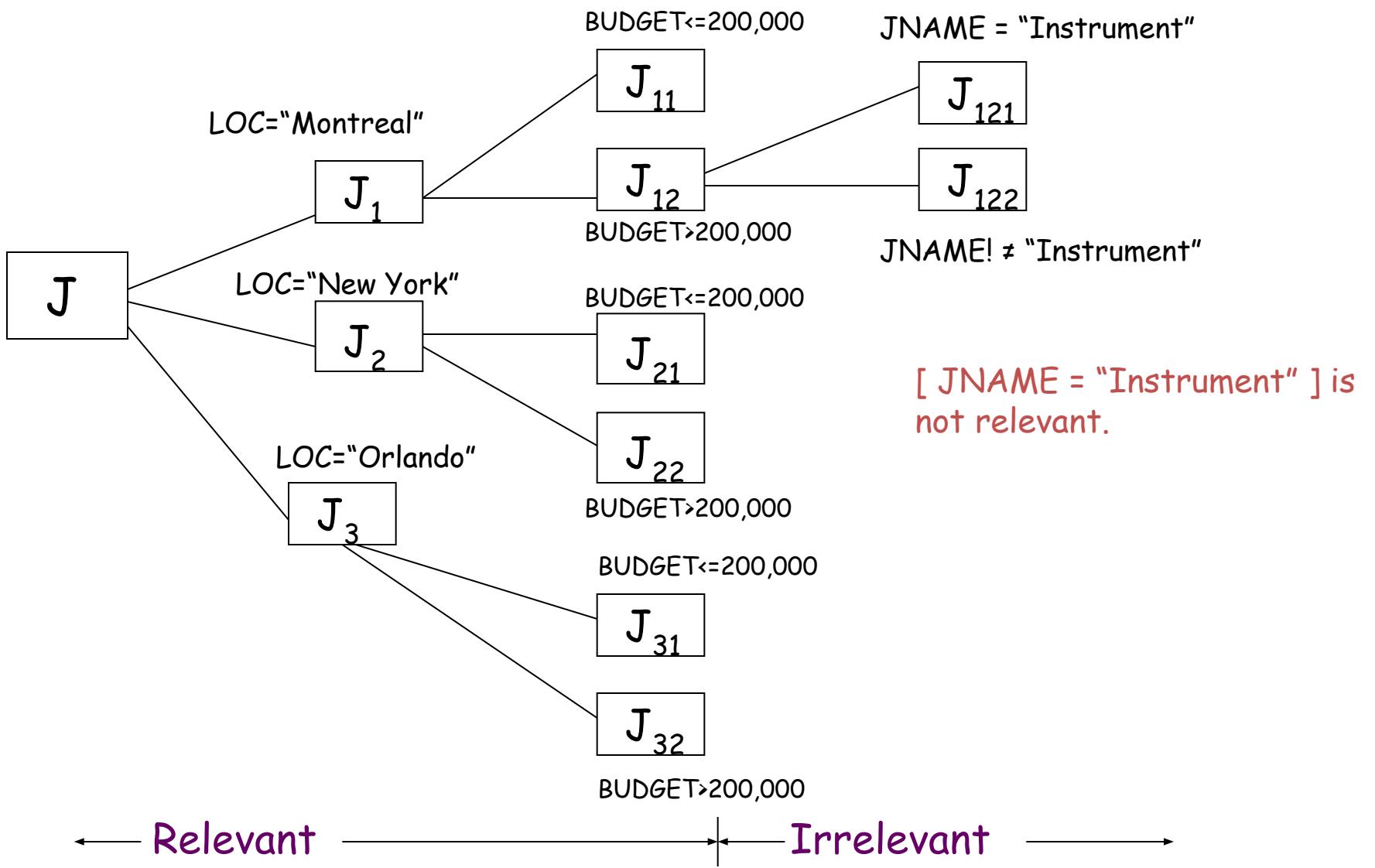
1. UD1 truy cập theo vị trí (Loc)
2. UD2 chỉ truy cập vào những bộ có kinh phí (BUDGET) lâ»»n hơn \$200,000.

TH1: $\text{Pr}=\{\text{Loc}=\text{"Montreal"}, \text{Loc}=\text{"New York"}, \text{Loc}=\text{"Orlando"}$

$\text{BUDGET}>=200\}$ is

Đầy đủ

TH2: Nếu đưa thêm vị từ JNAME= "Instrumentation" vào Pr, thì vị từ mới sẽ không liên đới gì tới các ứng dụng, => Pr không cực tiểu



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”. – qui tắc
- 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 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 mǔ 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 mảng ngang nguyên thủy

- Nếu $\text{Pr}' = \{p_1, p_2\}$, trong đó:
 - $p_1: \text{att} = \text{value_1}$
 - $p_2: \text{att} = \text{value_2}$
- I chứa hai phép kéo theo với khẳng định:
 - $i_1: (\text{att} = \text{value_1}) \Rightarrow \neg(\text{att} = \text{value_2})$
 - $i_2: \neg(\text{att} = \text{value_1}) \Rightarrow (\text{att} = \text{value_2})$

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 $p_i \in Pr'$

 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: $\text{SAL} \leq 30000$
 - Vị trí thứ hai: $\text{SAL} > 30000$
- 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:

$p_1: S / p_1: \text{SAL} \leq 30000$

$p_2: S / p_2: \text{SAL} > 30000$

Ví dụ về phân mảnh ngang nguyên thủy – PAY

- Từ tập vị từ đơn giản khởi đầu là $Pr = \{p_1, p_2\}$.
- Áp dụng COM_MIN với $i=1$ làm giá trị khởi đầu tạo ra $Pr'=\{p_1\}$ là đầy đủ và cực tiểu vì p_2 không phân hoạch f_1 (là mảnh hội sơ cấp được tạo ra ứng với p_1) 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 ₁	
TITLE	SAL
Mech. Eng.	27000
Programmer	24000

PAY2	
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

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:

p₁: LOC = “Montreal”

p₂: LOC = “New York”

p₃: 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:

$p_4: \text{BUDGET} \leq 200000$

$p_5: \text{BUDGET} > 200000$

- Sử dụng thuật toán COM_MIN để tìm Pr' là đầy đủ và cực tiểu từ từ tập Pr
 $Pr = \{p1, p2, p3, p4, p5\}$







Ví dụ về phân mảnh ngang nguyên thủy – PROJ

- Tập M các vị trí hội sở 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 trí hội sở cấp M.



Ví dụ về phân mảng ngang nguyên thủy – PROJ

- ☐ Các mảng PROJ₂, PROJ₅ rỗng.

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	250000	New York

PROJ6

PNO	PNAME	BUDGET	LOC
P4	Maintenance	31000	Paris

Phân mảnh ngang dã̃n xuất

Phân mảnh ngang dã̃n 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 mảng ngang dẫn xuất

Ví dụ: Phân mảng dẫn xuất quan hệ EMP theo quan hệ PAY.

Nhóm người tham gia dự án thành hai nhóm theo lương

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

$\text{PAY1} = \sigma_{\text{SAL} \leq 30000}(\text{PAY})$

$\text{PAY2} = \sigma_{\text{SAL} > 30000}(\text{PAY})$

$\text{EMP}_1 = \text{EMP} \triangleright \leq \text{PAY}_1$

$\text{EMP}_2 = \text{EMP} \triangleright \leq \text{PAY}_2$

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

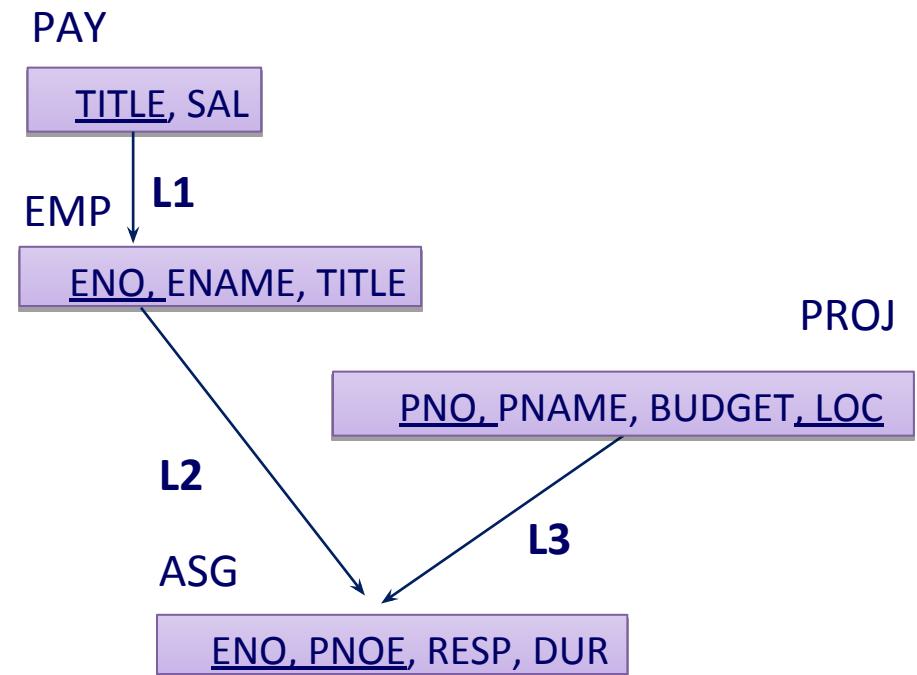
PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris

PAY

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

Phân mảng dẫn xuất quan hệ ASG theo quan hệ PROJ và EMP
Xét quan hệ ASG có hai ứng dụng trên nó:



Phân mảng ngang dân xuất

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)

- Dựa vào UD1 phân mảng ngang nguyên thủy PROJ thahf 3 mảng theo 3 vị trí thực hiện dự án (GS dự án chỉ triển khai tại vị trí M,N,P)

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

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

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

Phân mảng ngang dẫn xuất

-Phân mảng dẫn xuất ASG theo các mảng PROJ₁, PROJ₂ và PROJ₃ như sau:

$$ASG_1 = ASG \bowtie PROJ_1$$

$$ASG_2 = ASG \bowtie PROJ_2$$

$$ASG_3 = ASG \bowtie PROJ_3$$

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.

-Phân mảng gnang nguyên thủy EMP theo 2 vị trí công việc(Gs 2 công việc đó là Syst,Anal và Programmer

$$EMP_1 = \sigma_{TITLE = "Syst. Anal"}(EMP)$$

$$EMP_2 = \sigma_{TITLE = "Programmer"}(EMP)$$

Phân mảng ngang dẫn xuất

-Phân mảng dẫn xuất các ASGi theo EMP₁ và EMP₂ như sau:

$$ASG_{11} = ASG_1 \bowtie EMP_1$$

$$ASG_{12} = ASG_1 \bowtie EMP_2$$

$$ASG_{21} = ASG_2 \bowtie EMP_1$$

$$ASG_{22} = ASG_2 \bowtie EMP_2$$

$$ASG_{31} = ASG_3 \bowtie EMP_1$$

$$ASG_{32} = ASG_3 \bowtie EMP_2 |$$

Phân mảng ngang dẫn xuất

Nhận xét

- Phân mảng dẫn xuất có thể xảy ra dây chuyền, trong đó một quan hệ được phân mảng như là hệ quả của một phân mảng cho một quan hệ khác, và đến lượt nó lại làm cho các quan hệ khác phải phân mảng (như dây chuyền PAY-EMP-ASG).
- Một quan hệ có thể có nhiều cách phân mảng. Chọn lựa một lược đồ phân mảng 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.

Tự xây dựng 2 ứng dụng và thực hiện

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

- Thiết lập một lược đồ thực thể liên kết mô tả CSDL quản lý việc bán hàng. Trong lược đồ này, mỗi nhân viên bắt buộc phải bán một vài mặt hàng nào đó, và mỗi mặt hàng chỉ có thể được bán bởi một nhân viên. Mỗi mặt hàng được xác định bởi một mã số phân biệt, giá bán, chủng loại. Mỗi nhân viên sẽ có một mã số nhân viên riêng biệt, và các thông tin về tên, ngày sinh, tuổi, địa chỉ, giới tính. Một nhân viên có thể có nhiều địa chỉ sinh sống. Thời điểm bán hàng của từng mặt hàng sẽ được lưu lại trong CSDL.

- 1) Thiết kế lược đồ quan hệ cho CSDL trên
- 2) Tự xây dựng 2 ứng dụng và thực hiện
 - Xây dựng lược đồ phục vụ cho phân mảng ngang dẫn xuất, nguyên thủy
 - Thiết kế phân mảng ngang

- Thiết lập một lược đồ ER mô tả CSDL quản lý dự án gồm thông tin về các dự án và các nhân viên. Trong kịch bản này, mỗi nhân viên có thể thực hiện tối đa 10 dự án. Mỗi dự án được thực hiện bởi ít nhất 2 nhân viên. Khi một nhân viên thực hiện một dự án nào đó thì cần phải có một nhân viên khác giám sát quá trình thực hiện đó. Sinh viên tự bổ sung các thuộc tính cho các tập thực thể và các liên kết để đảm bảo mỗi tập thực thể có ít nhất 4 thuộc tính, trong đó có ít nhất 1 thuộc tính khoá và 1 thuộc tính dẫn xuất.

1.Thiết kế lược đồ quan hệ cho CSDL trên

2.Tự xây dựng 2 ứng dụng và thực hiện

a. Tập vị từ đơn giản có thể có

b. Tìm tập vị từ có tính đầy đủ và tính cực tiểu theo thuật toán COM_MIN

c. Tìm Tập các vị từ hội sơ cấp M

d.Phân mảnh ngang cơ sở trên tập các vị từ hội sơ cấp