**Mục lục**

[**1. Quá trình, trình tự phân mảnh** 2](#_Toc85384123)

[**2. Minterm** 3](#_Toc85384124)

[**3. Phân mảnh ngang** 4](#_Toc85384125)

[**a.** **Cơ sở phân mảnh ngang** 5](#_Toc85384126)

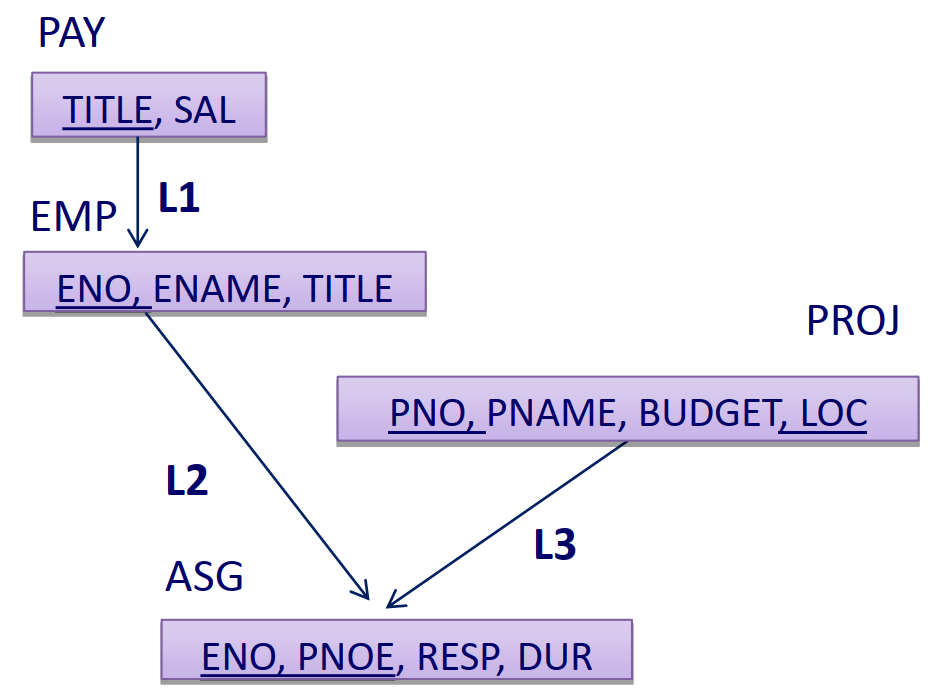
[**b.** **Tính đầy đủ (COM) của Pr** 5](#_Toc85384127)

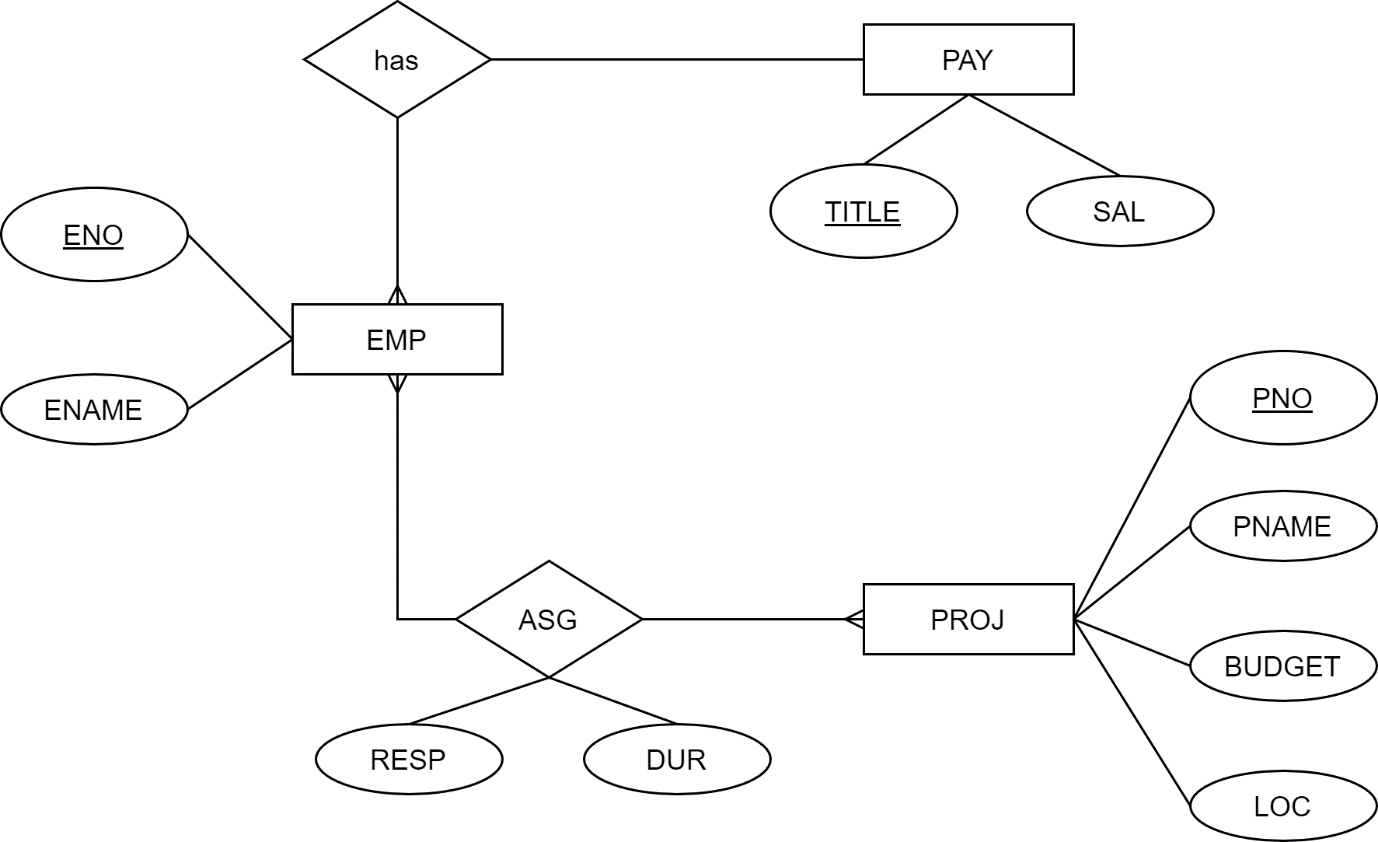
[**c.** **Tính tối thiểu/ cực tiểu (MIN) của Pr** 8](#_Toc85384128)

[**d.** **Thuật toán COM-MIN** 9](#_Toc85384129)

***- by Team 8 CSDL Phân tán PTIT***

# **1. Quá trình, trình tự phân mảnh**





PAY(TITLE, SAL) *\* Khi phân mảnh thì ta phân mảnh từ gốc trước*

EMP(ENO, ENAME, TITLE) *\* Sau khi phân mảnh gốc xong mới phân mảnh các ngọn*

*\* Tức là ta phân mảnh bản PAY trước rồi đến EMP 🡪 PROJ 🡪 ASG*

*Càng về ngọn thì càng phân mảnh sau.*

*PAY là gốc của EMP.*

*(EMP và PROJ) là gốc của ASG.*

*Phân mảnh lần lượt (ngược lại với chiều mũi tên ở thiết kế Logic)*

Thiết kế CSDL như nào thì quá trình phân mảnh đi theo chiều ngược lại với quá trình thiết kế ấy

# **2. Minterm**

Câu truy vấn *Select … from … where*

Bản chất của Where là lựa chọn các bản ghi thỏa mãn điều kiện nào đó

Mỗi điều kiện ở trong phần Where tương ứng với 1 **Predicate (thuộc tính | Vị từ đơn giản)**

Tập hợp tất cả các Predicate trong phần Where đó thì gọi là 1 **Minterm – Vị từ hội**

Minterm là tập hợp các điều kiện kết nối và chọn lọc thông tin. Mỗi Minterm phục vụ cho 1 câu truy vấn, mỗi câu truy vấn bao gồm 1 Minterm

Tóm lại, Minterm là sự kết hợp AND của các Predicates (Không có OR), OR là sự kết hợp các Minterm với nhau

|  |  |
| --- | --- |
| **SELECT**  **FROM**  **WHERE (minterm)** | Ký hiệu của toán bộ câu truy vấn này là:  **Sel(m)** |

VD: 1 câu truy vấn Sel(m) 🡪 ký hiệu là ***q*** (query)

Thực hiện truy vấn lên bảng PAY:

|  |  |  |
| --- | --- | --- |
| ***q*** | **TITLE** | **SAL** |
| ***1*** | ***0*** |

* Câu truy vấn ***q*** thực hiện lên thuộc tính TITLE trên bảng PAY

Bảng trên thể hiện **tần số truy cập** của câu truy vấn lên bảng PAY như thế nào, gọi là Access, ký hiệu là **Acc(q)**

Về bản chất acc(q) là một vector = (1, 0) thể hiện là câu truy vấn này truy cập vào thuộc tính TITLE 1 lần mà không truy cập vào thuộc tính SAL

Có thể truy cập nhiều lần, thì tăng số lên. Còn không truy cập thì là **0**

# **3. Phân mảnh ngang**

|  |  |  |
| --- | --- | --- |
|  | **TITLE** | **SAL** |
| **F1** | 1 | 0 |
| **F2** | … | … |
| **…** | … | … |

Phân mảnh ngang cơ sở được định nghĩa bằng phép chọn trên quan hệ toàn R:

=

; trong đó (minterm) là vị từ hội sơ cấp

**(Minterm Fragment) được gọi là Mảnh** hội sơ cấp

* 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 sơ cấp: phân mảnh từ lược đồ gốc – trên PAY (quan trọng nhất vì nó liên quan đến các lược đồ tiếp theo)
* Phân mảnh thứ cấp: phân mảnh từ các lược đồ tiếp theo – các bảng tiếp theo

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

: { BUDGET ≤ 200000 }  
: { 200000 < BUDGET ≤ 400000 }  
: { 400000 < BUDGET ≤ 600000 }  
: { 600000 < BUDGET }

Khi đó quan hệ PROJ (bảng to nhất) được phân rã thành các mảnh ngang như sau:

PROJ1 =   
PROJ2 =   
PROJ3 = PROJ4 =

## **Cơ sở phân mảnh ngang**

**▪ Cho trước:** Quan hệ *R*, có tập các vị từ đơn giản *Pr***▪ Đầu ra:** Tập các mảnh của *R = {R1, R2,…}* tuân theo các luật phân mảnh  
**▪ Điều kiện:**

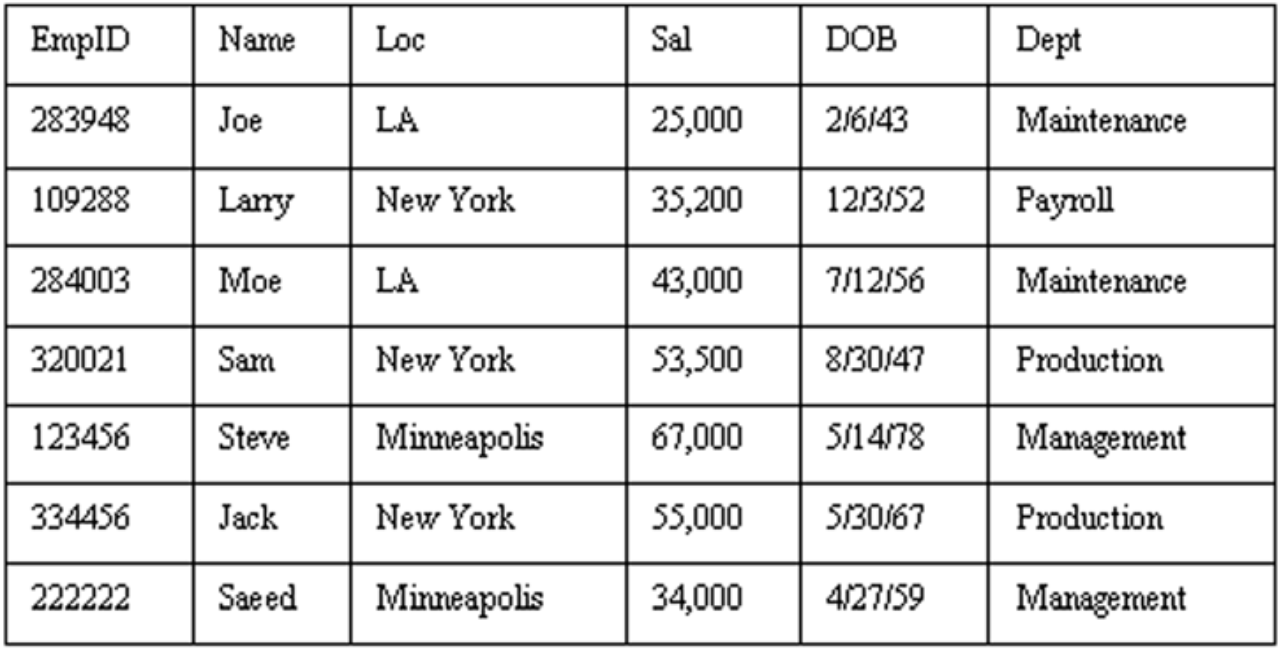
*\* Pr* phải đầy đủ  
\* *Pr* phải tối thiểu

Pr : tập vị từ đơn giản = {}

M : Tập vị từ hội = {}

## **Tính đầy đủ (COM) của Pr**

Vị từ Pr là đầy đủ (Nói về sự không bị thiếu của tập Pr) khi và chỉ khi xác suất truy nhập của mỗi ứng dụng đến một bộ bất kỳ của một mảnh hội sơ cấp bất kỳ được định nghĩa theo Pr là như nhau

**Ví dụ 1:** 

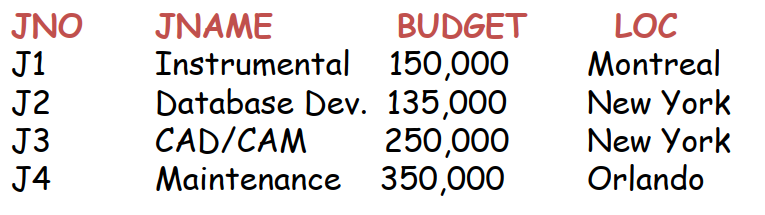
Giả sử ứng dụng APP1 truy vấn vào quan hệ EMP (theo bảng trên) để tìm kiếm những nhân viên làm việc ở Los Angeles (LA), Có:

* Tập vị từ đơn giản là Pr = {: Loc= “LA”}
* Tập vị từ hội là {: Loc = “LA”, : Loc<>“LA”}”,

Vậy tập vị từ Pr là cực tiểu và đầy đủ , các mảnh được phân ra:

* **Mảnh :** Create table LA\_EMPS as Select \* from EMP Where Loc = "LA";
* **Mảnh :** Create table NON\_LA\_EMPS as Select \* from EMP Where Loc <>  
  "LA";

**Ví dụ 2:**

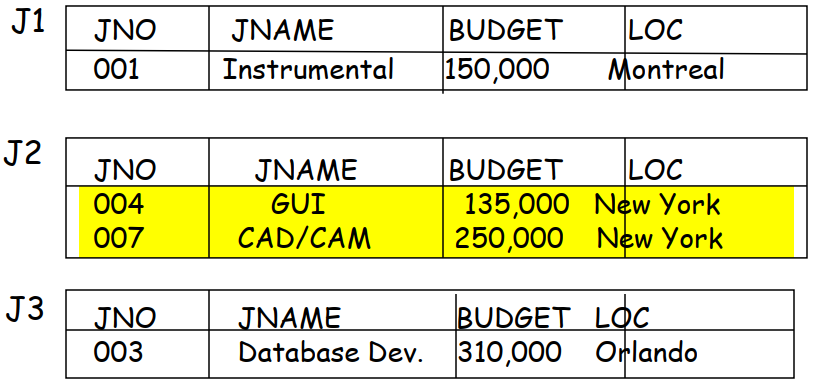


App truy vấn vào bảng để tìm những thành phần đến từ các vị trí trong bản J:



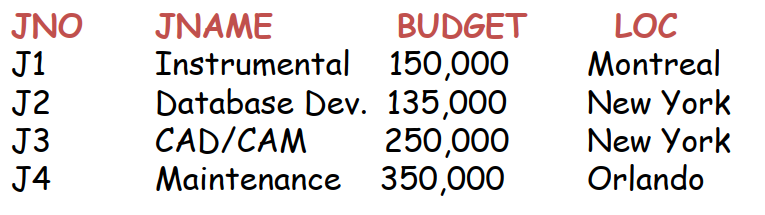
Có tập vị từ đơn giản:

Là một tập vị từ đầy đủ vì mỗi bộ trong 1 mảnh được phân ra có xác suất truy cập như nhau. Ta có các mảnh:



Bảng J ban đầu đã được phân mảnh ngang thành 3 mảnh, mỗi mảnh là các bộ mà trong đó các bộ đều có chung đặc điểm là Location. Vì app chỉ quan tâm đến Location 🡺 Đầy đủ

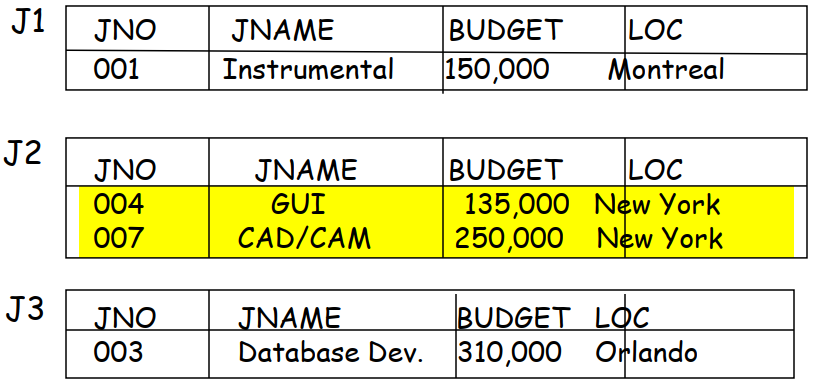
**Ví dụ 3:**



App chỉ truy vấn vào bảng để tìm những thành phần có BUDGET $200.000

Hiện tại là có 2 đk: ở NY và kinh phí <=200$

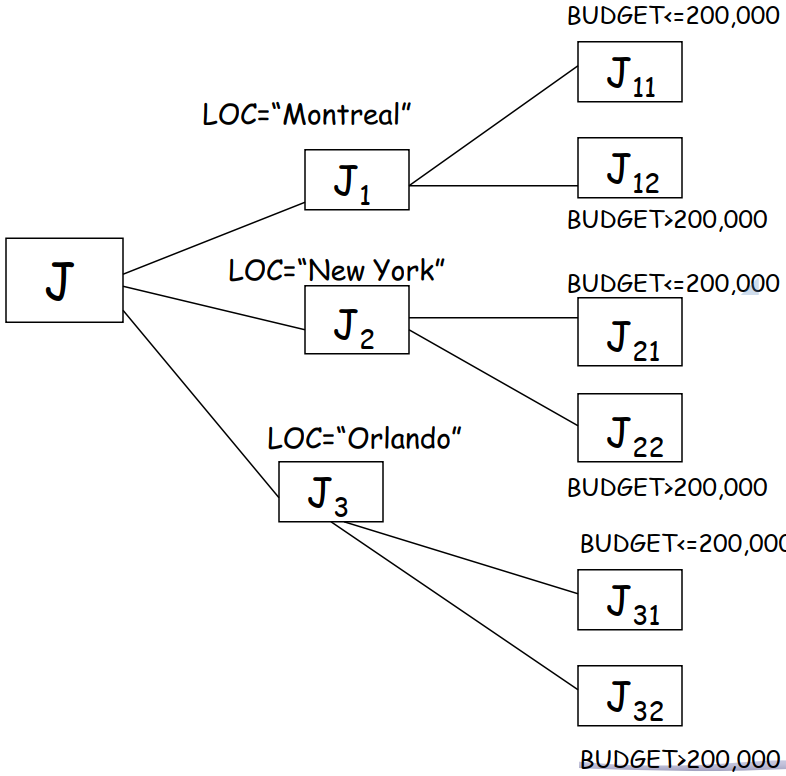
Theo như bảng ở ví dụ 1:



Trong mảnh (bảng) J2, thì bộ JNO = 004 được truy cập nhiều hơn bộ JNO = 007

* Vị từ:

Không còn đầy đủ. Vì vậy chúng ta cần thêm thành phần, chia tiếp ra các bảng:



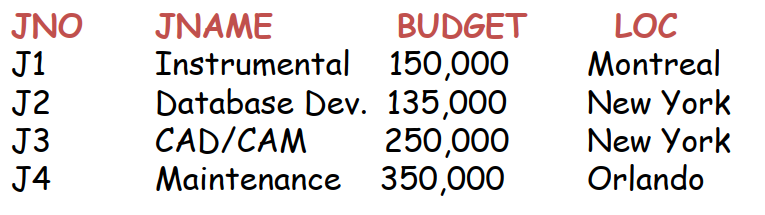
Để đảm bảo tính đầy đủ, bảng J ban đầu đã tiếp tục được phân mảnh ngang thành 6 mảnh, mỗi mảnh là bảng mà trong đó các bộ đều có xác suất truy cập là như nhau (Nếu bảng nào có các bộ mà BUDGET &200.000 thì xác suất truy cập mỗi bộ trong bảng đó có xác suất truy cập như nhau = 0, và ngược lại) 🡺 Đã đầy đủ

## **Tính tối thiểu/ cực tiểu (MIN) của Pr**

**Pr là cực tiểu** (nói đến sự không bị thiếu của Pr) khi tất cả các liên đới

**liên đới** khi và chỉ khi: Chia một mảnh (bảng) thành 2 mảnh thì tồn tại một ứng dụng truy cập vào cả và theo 2 cách khác nhau

**Ví dụ 4:**



Có 2 App (2 loại câu lệnh ~ 2 giao diện cùng tỏng 1 phần mềm):

**APP1:** truy cập theo LOC

**APP2:** truy cập theo BUDGET

Xét vị từ : LOC = “Montreal”

Ta có 2 mảnh:

Mảnh 1: = { : LOC = “Montreal”}

Mảnh 2: = { - }

APP1 truy cập vào mảnh 1 theo cách và truy cập mảnh 2 theo cách . Mà . 🡺 vị từ liên đới

## **Thuật toán COM-MIN**

|  |
| --- |
| **Tóm tắt:**   * Đầu vào: Một quan hệ R và một tập các vị từ đơn giản Pr * Đầu ra: Một tập vị từ *Pr'* đầy đủ và tối thiểu   *Một quan hệ hoặc một mảnh được phân chia thành ít nhất hai phần được truy cập khác nhau bởi ít nhất một ứng dụng* |

|  |
| --- |
| **Thuật toán COM-MIN:**   * Khởi tạo : * Chọn một ∈ sao cho phân chia R theo luật 1 * Thiết lập ban đầu:  1. Thêm vào tập ; 2. Bỏ ra khỏi tập ; 3. Thêm vào tập  * Lặp lại việc thêm các vị từ vào Pr’ cho đến khi hoàn thành: ▪ Tiếp tục chọn một trong sao cho phân chia R theo luật 1 ▪ Tiếp tục theo các bước: Thêm vào tập 🡪 Bỏ ra khỏi tập 🡪 Thêm vào tập  ▪ Nếu ∃ pk ∈ Pr' không liên đới thì: bỏ ra khỏi và Thêm ra khỏi |

Tìm tập các vị từ đơn giản đầy đủ và cực tiểu là bước đầu tiên trong thiết kế phân mảnh ngang nguyên thủy