

Chương 2

Thiết kế CSDL phân tán



Phạm Thị Ngọc Diễm

Bộ môn HTTT - Khoa CNTT&TT

ptndiem@cit.ctu.edu.vn

Nội dung

- **Giới thiệu**
- Phân đoạn
- Cấp phát dữ liệu



Giới thiệu

- Trong một D-DBMS, dữ liệu được phân tán **nhằm tận dụng lợi thế của tất cả các tài nguyên máy tính có sẵn của tổ chức.**
 - Đối với các hệ thống này, **thiết kế lược đồ được thực hiện từ trên xuống.**
- Tiếp cận **thiết kế từ trên xuống** xem xét các yêu cầu dữ liệu của toàn bộ tổ chức và tạo ra một mô hình quan niệm toàn cục (GCM) của tất cả các thông tin được yêu cầu.
 - GCM sau đó được phân tán trên tất cả các DBMS cục bộ (LDBMS) phù hợp để tạo ra các mô hình quan niệm cục bộ (Local conceptual model - LCM)
 - D-DBMS có một và chỉ một GCM và một hoặc nhiều LCM

Giới thiệu

- Ngược lại, việc thiết kế một hệ thống **cơ sở dữ liệu liên hợp** (F-DB) được thực hiện từ dưới lên.
 - Tiếp cận **thiết kế từ dưới lên** xem xét **các dữ liệu tồn tại đã phân tán** trong một tổ chức và sử dụng một quá trình gọi là **tích hợp lược đồ** (schema integration) để tạo ít nhất một **lược đồ thống nhất** (unified schema)
 - Unified schema tương tự GCM nhưng điểm khác là có thể có nhiều hơn 1 unified schema.
- => Nội dung học phần tập trung vào CSDL phân tán

Giới thiệu

- Người thiết kế D-DB sẽ quyết định cách phân tán nào là tốt nhất cho một tình huống.
 - *Giữ nguyên vẹn mỗi bảng* (tất cả các dòng và tất cả các cột của mỗi bảng được lưu trữ trong cùng DB tại cùng site)
 - *Phân rã một vài bảng thành những phần dữ liệu nhỏ hơn được gọi là đoạn (fragment).*
- Người thiết kế D-DB cũng có thể quyết định
 - *Lưu trữ cục bộ các đoạn này*
 - *Hoặc lưu trữ chúng thông qua một số LDBMS trên mạng*

Giới thiệu

- **Mục tiêu** của việc phân tán dữ liệu là để
 - tăng tính sẵn sàng,
 - tăng độ tin cậy, và
 - cải thiện thời gian truy cập các câu truy vấn.
- Tuy nhiên dữ liệu được phân tán thông thường *phải mất nhiều thời gian* hơn để cập nhật (sửa, xóa, và thêm).

Nội dung



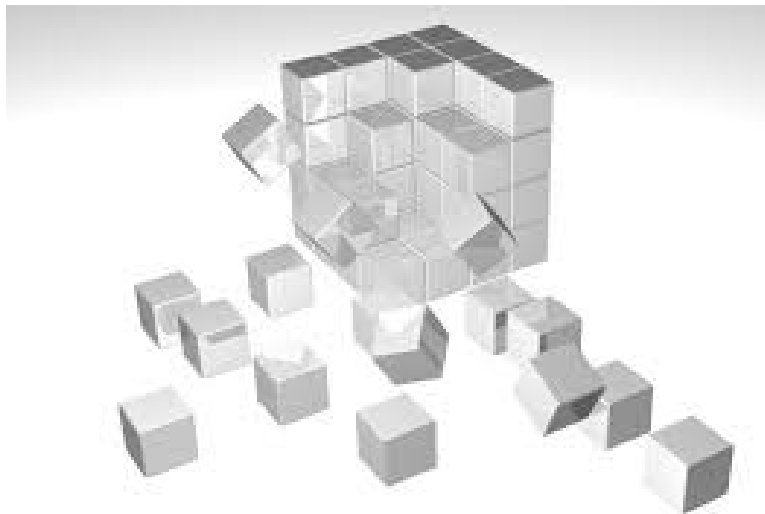
- Giới thiệu
- **Phân đoạn**
 - Tại sao phân đoạn?
 - Làm thế nào để phân đoạn?
 - Bao nhiêu đoạn?
 - Làm thế nào để kiểm tra tính đúng đắn?
- Cấp phát dữ liệu

Nhắc lại

- **Đại số Bool: Luật De Morgan**
 - $\neg (P \wedge Q) \Leftrightarrow (\neg P) \vee (\neg Q)$
 - $\neg (P \vee Q) \Leftrightarrow (\neg P) \wedge (\neg Q)$
- **Đại số quan hệ:**
 - Phép chọn
 - Phép chiếu
 - Nối kết
 - Hợp, giao, trừ

Định nghĩa

- **Phân đoạn** là quá trình **phân rã** của một cơ sở dữ liệu thành một tập hợp các CSDL con được gọi là các đoạn (fragment)
- Mỗi đoạn được lưu trữ tại bất kỳ site nào trong mạng
- Thông tin về phân đoạn dữ liệu như vị trí phân đoạn, nội dung phân đoạn,... được chứa trong GDD.



Mục tiêu

- Phân đoạn nhằm cải thiện:
 - Độ tin cậy
 - Hiệu suất
 - Khả năng lưu trữ và chi phí
 - Chi phí truyền thông
 - Bảo mật
- Các thông tin sau được sử dụng để quyết định phân đoạn:
 - Thông tin định lượng: tần suất truy vấn, site, câu truy vấn được chạy ở đâu, sự chọn lọc của các truy vấn...
 - Thông tin định tính: các loại truy cập dữ liệu, đọc/ghi, ...

Tính đúng đắn của phân đoạn

- **Sự phân rã phải không mất thông tin.**
 - Cho bất kỳ dữ liệu nào của quan hệ R , có một đoạn R_i của quan hệ R có chứa dữ liệu này → **Luật đầy đủ.**
 - Không mất thông tin được kiểm tra bằng cách xây dựng lại quan hệ từ các đoạn khác nhau sử dụng ngôn ngữ thao tác dữ liệu (ví dụ ngôn ngữ SQL)
 - Hợp của các quan hệ phải trả về lược đồ toàn cục ban đầu → **Luật xây dựng**
- Các đoạn khác nhau phải loại trừ nhau nghĩa là giao của chúng là rỗng → **Luật loại trừ**
 - Trong trường hợp có sự trùng lặp → cần tinh chỉnh việc phân đoạn

Kỹ thuật phân đoạn

- Có nhiều cách khác nhau để phân đoạn CSDL
- Đơn vị phân đoạn xác định tính tự nhiên của các thành phần nhỏ nhất được dùng để phân đoạn.
- Bốn đơn vị phân đoạn cơ bản:
 - quan hệ / lớp (class) đối tượng,
 - bộ,
 - thuộc tính và
 - giá trị.

Phân đoạn bởi quan hệ

- Các lớp - class (bảng/quan hệ trong mô hình quan hệ, đối tượng trong mô hình đối tượng...) có thể được phân tán trong các đoạn khác nhau
 - Các đoạn được định nghĩa từ các class của CSDL.
 - Tất cả các thể hiện của cùng lớp thuộc cùng đoạn.
- Thao tác phân rã là định nghĩa lược đồ con.
- Thao tác xây dựng là nhóm lại các sơ đồ con.
- Ví dụ:
 - Xét CSDL ngân hàng gồm 3 bảng: TaiKhoan, KhachHang và ChiNhanh

Phân đoạn bởi quan hệ

TaiKhoan

MaKH	TenCN	Loai	sodu
174 723	Lausanne	The	123 345.89
177 498	Genève	The	34 564.00
201 639	Lausanne	The	45 102.50
178 123	Lausanne	Tietkiem	325 100.00
203 446	Genève	The	274 882.95

KhachHang

MaKH	Ho	Ten	Tuoi
174 723	Villard	Jean	29
177 498	Cattell	Blaise	38
201 639	Tesllis	Alan	51
178 123	Bellot	Patrick	39
203 446	Kovalsky	Validmir	36

ChiNhanh

TenCN	Diachi
Lausanne	Rue du Lac, 3, 1002 Lausanne
Genève	Avenue du Mont Blanc, 21, 1200 Genève

=> CSDL này có thể được phân thành 2 đoạn
{TaiKhoan, KhachHang} và {ChiNhanh}

Phân đoạn ngang

- Phân đoạn ngang (Horizontal Fragmentation) còn được gọi là phân đoạn bởi thể hiện (occurrence) hoặc bởi bộ (tuple)
- Thể hiện/các bộ của các quan hệ/lớp có thể được chia thành các đoạn khác nhau (với tất cả các thuộc tính).
- Thao tác phân rã dùng phép chọn (σ)
- Thao tác xây dựng lại dùng phép hợp (\cup)
- **Ví dụ:** trong CSDL ngân hàng, quan hệ TaiKhoan được phân thành 2 đoạn theo 2 loại tài khoản
 - $Taikhoan1 = \sigma_{Loai='The'}(TaiKhoan)$
 - $Taikhoan2 = \sigma_{Loai='Tietkiem'}(TaiKhoan)$

Phân đoạn ngang

TaiKhoan1

MaKH	TenCN	Loai	sodu
174 723	Lausanne	The	123 345.89
177 498	Genève	The	34 564.00
201 639	Lausanne	The	45 102.50
203 446	Genève	The	274 882.95

TaiKhoan2

MaKH	TenCN	Loai	sodu
178 123	Lausanne	Tietkiem	325 100.00

Thao tác xây dựng lại được thực hiện bởi phép hợp:

$$\text{Taikhoan} = \text{TaiKhoan1} \cup \text{TaiKhoan2}$$

Phân đoạn dọc

- Phân đoạn dọc (Vertical Fragmentation) còn gọi là **phân đoạn theo thuộc tính (attribute)**.
 - Các thuộc tính của cùng quan hệ có thể được phân tán trong nhiều đoạn khác nhau
 - Tất cả các giá trị của một thể hiện/các bộ cho cùng thuộc tính được phân trong cùng đoạn.
 - Phân bố các thuộc tính trong các đoạn khác nhau chỉ đúng khi **định danh/khoá chính (primary key) bị trùng lặp trong các đoạn**. Định danh này được dùng để xây dựng lại quan hệ
- Phân đoạn dọc cần thiết cho việc phân tán các phần dữ liệu trên các site nơi mà mỗi phần này được sử dụng.

Phân đoạn đọc

- Phép toán phân đoạn là phép chiếu (π)
- Phép xây dựng lại là phép kết nối (join)
- Ví dụ: CSDL ngân hàng
 - Ở các quầy giao dịch khách hàng người ta chỉ cần biết số dư, trong khi dịch vụ tranh chấp, người ta cần biết các thông tin khác.

=> phân thành 2 đoạn

- $TaiKhoan1 = \pi_{MaKH, sodu}(TaiKhoan)$
- $Taikhoan2 = \pi_{MaKH, tenCN, Loai}(Taikhoan)$

Phân đoạn đọc

TaiKhoan1

MaKH	sodu
174 723	123 345.89
177 498	34 564.00
201 639	45 102.50
178 123	325 100.00
203 446	274 882.95

TaiKhoan2

MaKH	TenCN	Loai
174 723	Lausanne	The
177 498	Genève	The
201 639	Lausanne	The
178 123	Lausanne	Tietkiem
203 446	Genève	The

Xây dựng lại quan hệ bằng cách dùng phép kết nối

Taikhoan= TaiKhoan 1 * Taikhoan2

Phân đoạn lai

- Phân đoạn lai (hybrid fragmentation) là sự kết hợp của phân đoạn ngang và dọc, còn được gọi là phân đoạn theo giá trị (value)
 - Phân đoạn được thực hiện bằng cách kết hợp dùng phép chọn và phép chiếu
 - Thể hiện/các bộ và các thuộc tính có thể được phân bố trong nhiều đoạn khác nhau
- Xây dựng lại được thực hiện bằng cách kết hợp phép kết nối và phép hợp các đoạn khác nhau.

Phân đoạn lại

- Ví dụ: CSDL ngân hàng
 - Taikhoan1 = $\Pi_{\text{MaKH}, \text{sodu}}(\sigma_{\text{loai} = \text{'The'}}(\text{TaiKhoan}))$
 - Taikhoan2 = $\Pi_{\text{MaKH}, \text{sodu}}(\sigma_{\text{loai} = \text{'Tietkiem'}}(\text{TaiKhoan}))$
 - Taikhoan3 = $\Pi_{\text{MaKH}, \text{TenCN}}(\text{TaiKhoan})$
 - Taikhoan4 = $\Pi_{\text{MaKH}, \text{Loai}}(\text{TaiKhoan})$

Phân đoạn lại

Taikhoan1		Taikhoan2		Taikhoan3		Taikhoan4	
MaKH	sodu	MaKH	sodu	MaKH	TenCN	MaKH	Loai
174 723	123 345.89	178 123	325 100.00	174 723	Lausanne	174 723	The
177 498	34 564.00			177 498	Genève	177 498	The
201 639	45 102.50			201 639	Lausanne	201 639	The
203 446	274 882.95			178 123	Lausanne	178 123	Tietkiem
				203 446	Genève	203 446	The

Xây dựng lại quan hệ bằng cách dùng phép kết nối

Taikhoan= (TaiKhoan1 U Taikhoan2)*Taikhoan3*Taikhoan4

Phương pháp phân đoạn ngang

- Việc áp dụng phân đoạn theo chiều ngang vào một bảng sẽ tạo ra **một tập các đoạn có các dòng/bộ tách rời.**
- Phân đoạn ngang là cần thiết vì nó có thể nhóm lại các dòng của một bảng đáp ứng các vị từ (**predicate**) của các **câu truy vấn được chạy thường xuyên.**
 - **Chú ý:** Vị từ = điều kiện
- Người thiết kế một hệ thống D-DB nên lưu trữ từng đoạn ngang như vậy tại địa điểm nơi mà các câu truy vấn này được chạy.,
- **Phép chọn là cơ sở để phân đoạn ngang**

Phương pháp phân đoạn ngang

- Cho quan hệ R và một tập các vị từ/điều kiện được rút trích từ các câu truy vấn thường xuyên.
- *Chú ý: Người ta chỉ quan tâm đến các câu truy vấn được thực thi thường xuyên*

Mục tiêu: Làm thế nào để chúng ta quyết định điều kiện nào được sử dụng khi chúng ta phân đoạn theo chiều ngang một bảng?



Phương pháp phân đoạn ngang

- **Định nghĩa:**

- **Predicate đơn giản, ký hiệu p :** là điều kiện được biểu diễn dưới dạng:

$\langle \text{Tên_cột} \rangle \langle \text{Phép_so_sánh} \rangle \langle \text{giá_trị} \rangle$

- *Phép_so_sánh* là $\{=, <, >, >=, <=, <>\}$
- Ví dụ: $\text{Sodu} > 500.000$
- p chính là một trong các **điều kiện chọn** được rút trích từ các câu truy vấn

Phương pháp phân đoạn ngang

- **Định nghĩa:**
 - **Minterm predicate**, *ký hiệu m* , được định nghĩa như là sự kết hợp của các predicate đơn giản.
 - Ví dụ: $(tenCN='Lausanne' \wedge sodu > 100.000)$
 - \wedge là phép AND.
 - **Chỉ sử dụng AND** trong Minterm predicate và **không dùng OR**

Phương pháp phân đoạn ngang

- Gọi **Pr** là tập tất cả các vị từ đơn giản được sử dụng bởi tất cả các ứng dụng truy vấn một bảng đã cho
 $\mathbf{Pr} = \{p_1, p_2, \dots, p_n\}$
- Gọi **M** là tập tất cả các vị từ Minterm được sử dụng bởi tất cả các ứng dụng truy vấn bảng đã cho $\mathbf{M} = \{m_1, m_2, \dots, m_k\}$
- Áp dụng **M** vào bảng đã cho sẽ **tạo ra các đoạn ngang** biểu thị bởi tập hợp $F = \{F_1, F_2, \dots, F_k\}$
 - Đối với thiết kế phân mảnh này, tất cả các dòng trong F_i , ($i = 1..k$) thỏa mãn m_i .

Phương pháp phân đoạn ngang

- **Cách tính M từ Pr**

- Nếu có N predicate đơn giản trong Pr thì M sẽ có 2^N Minterm predicate. **Ví dụ:** Có 2 vị từ đơn giản trong Pr \Rightarrow có 4 vị từ Minterm trong M
- Mỗi vị từ minterm m_j có dạng $m_j = \bigwedge_{i=1..N} p_i^*$
 - với $(j = 1.. 2^N)$
 - p_i^* là p_i hoặc $\neg p_i$
- Tuy nhiên có một số các vị từ trong M không phù hợp \Rightarrow Cần xoá chúng khỏi M

\Rightarrow Số lượng các đoạn ngang thực tế trong F thường ít hơn 2^N

Phương pháp phân đoạn ngang

- **Tìm tập tối thiểu và đầy đủ của minterm**
 - Tập hợp các vị từ minterm là **đầy đủ** nếu và chỉ nếu bất kỳ hai bộ trong cùng đoạn được tham chiếu với cùng xác suất bởi bất kỳ ứng dụng
 - Tập hợp các vị từ minterm là **tối thiểu** nếu và chỉ nếu có ít nhất một câu truy vấn truy cập vào đoạn

Ví dụ về tính đầy đủ

- Cho quan hệ J như sau:

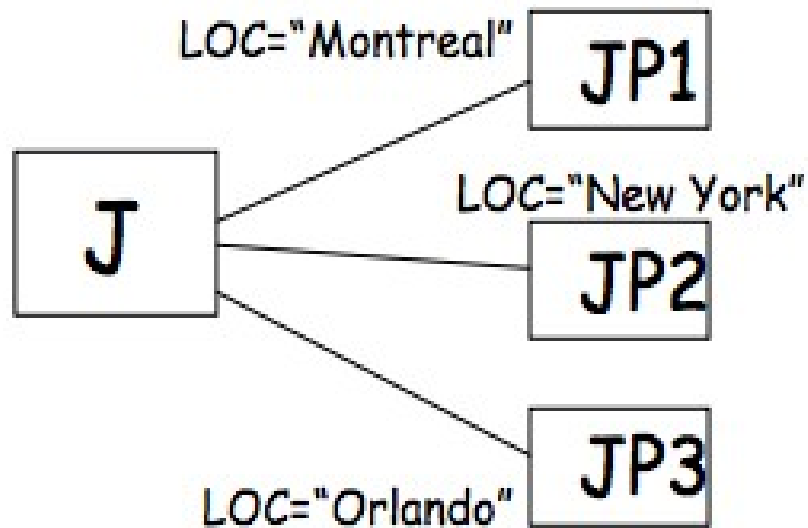
J	JNO	JNAME	BUDGET	LOC
	J1	Instrumental	150,000	Montreal
	J2	Database Dev.	135,000	New York
	J3	CAD/CAM	250,000	New York
	J4	Maintenance	350,000	Orlando

Ví dụ về tính **đầy đủ**

$$JP_1 = \sigma_{LOC = "MONTREAL"}(J)$$

$$JP_2 = \sigma_{LOC = "NewYork"}(J)$$

$$JP_3 = \sigma_{LOC = "Orlando"}(J)$$



- Ứng dụng duy nhất truy cập J muốn truy xuất các bộ theo bất kỳ vị trí nào.
 - 3 vị trí có thể là Orlando, Montreal, và NY.
 - Tập các vị từ đơn giản
 - $Pr = \{LOC = "Montreal", LOC = "New York", LOC = "Orlando"\}$
- => là **đầy đủ** bởi vì mỗi bộ của mỗi đoạn có cùng xác suất được truy cập

Ví dụ về tính đầy đủ

=> Ba đoạn theo 3 vị từ đơn giản.

JP1

JNO	JNAME	BUDGET	LOC
J1	Instrumental	150,000	Montreal

JP2

JNO	JNAME	BUDGET	LOC
J2	GUI	135,000	New York
J3	CAD/CAM	250,000	New York

JP3

JNO	JNAME	BUDGET	LOC
J4	Database Dev.	310,000	Orlando

=> Cách phân đoạn này có còn đầy đủ nếu có ứng dụng thứ 2 truy xuất chỉ các bộ có Budget nhỏ hơn 200.000 ?

Ví dụ về tính đầy đủ

- Các Pr trước đó không còn đầy đủ vì một số bộ trong JP_i đã có xác suất truy cập cao hơn
- Ví dụ, bộ "J2" có xác suất truy cập cao hơn bộ "J3" trong JP_2
 - 2 ứng dụng truy cập J2,
 - nhưng một ứng dụng truy cập J3
- Để làm cho Pr đầy đủ, chúng ta cần thêm vào Pr
(BUDGET \leq 200.000, BUDGET $>$ 200.000)

Ví dụ về tính đầy đủ

- Chỉ một ứng dụng truy cập J muốn truy xuất các bộ ở New York.
- Tập các vị từ đơn giản:
 - $Pr = \{LOC = \text{"Montreal"},$
 $LOC = \text{"New York"},$
 $LOC \neq \text{"New York"},$
 $LOC = \text{"Orlando"}\}$
- Pr đầy đủ không ?

Đầy đủ

Ví dụ về tính tối thiểu

- Chỉ một ứng dụng truy cập J muốn truy xuất các bộ ở New York.

JP1

JNO	JNAME	BUDGET	LOC
J1	Instrumental	150,000	Montreal

JP2

JNO	JNAME	BUDGET	LOC
J2	GUI	135,000	New York
J3	CAD/CAM	250,000	New York

JP3

JNO	JNAME	BUDGET	LOC
J4	Database Dev.	310,000	Orlando

- LOC="Montreal" phù hợp không ?*
- LOC="New York" phù hợp không ?*
- LOC="Orlando" phù hợp không ?*

Ví dụ về tính tối thiểu

=> Nếu tất cả các vị từ của một tập Pr là phù hợp, Pr là tối thiểu.

- Pi phù hợp nếu có ít nhất một ứng dụng
 - sử dụng chính xác pi hoặc $\neg pi$
 - và tần số truy cập của pi (dựa trên ứng dụng này) khác với tần số truy cập $\neg pi$

Ví dụ về tính tối thiểu

- Chỉ một ứng dụng truy cập J muốn truy xuất các bộ ở New York.

JP1

JNO	JNAME	BUDGET	LOC
J1	Instrumental	150,000	Montreal

JP2

JNO	JNAME	BUDGET	LOC
J2	GUI	135,000	New York
J3	CAD/CAM	250,000	New York

JP3

JNO	JNAME	BUDGET	LOC
J4	Database Dev.	310,000	Orlando

- LOC="Montreal" phù hợp không ?* → Không
- LOC="New York" phù hợp không ?* → Có
- LOC="Orlando" phù hợp không ?* → Không

Phương pháp phân đoạn ngang

- Ví dụ 1:** Cho bảng EMP (employee)

EmpID	Name	Loc	Sal	DOB	Dept
283948	Joe	LA	25,000	2/6/43	Maintenance
109288	Larry	New York	35,200	12/3/52	Payroll
284003	Moe	LA	43,000	7/12/56	Maintenance
320021	Sam	New York	53,500	8/30/47	Production
123456	Steve	Minneapolis	67,000	5/14/78	Management
334456	Jack	New York	55,000	5/30/67	Production
222222	Saeed	Minneapolis	34,000	4/27/59	Management

Giả sử ứng dụng A truy vấn bảng EMP để tìm những nhân viên làm việc ở Los Angeles (LA) => Phân đoạn như thế nào ?

Phương pháp phân đoạn ngang

- **Ví dụ 1 (tt):**

- Ta có tập $Pr = \{p1: Loc = 'LA'\}$ là tất cả các vị từ đơn giản được sử dụng bởi A. (Có 1 vị từ)
- Vậy M có 2^1 vị từ Minterm
 $\Rightarrow M = \{m1: Loc = 'LA', m2: Loc \neq 'LA'\}$ là các vị từ minterm tối giản và đầy đủ cho ứng dụng A
- M phân đoạn EMP vào hai đoạn sau:

F1:

```
Create table LA_EMPS as  
Select * from EMP  
Where Loc = 'LA';
```

F2:

```
Create table DIF_EMPS as  
Select * from EMP  
Where Loc <> 'LA';
```

Phương pháp phân đoạn ngang

- **Ví dụ 2 :**

Giả sử A truy vấn bảng EMP tìm các nhân viên cũng ở LA và có mức lương lớn hơn 30000

=> 2 đoạn F1 và F2 không còn phù hợp nữa

=> EMP được phân thành mấy đoạn ?

Phương pháp phân đoạn ngang

- Ví dụ 2 (tt):

$$Pr = \{ p1: Loc = 'LA', \\ p2: salary > 30000 \}$$

=> M có $2^2=4$ minterm

$$M = \{ m1: Loc = 'LA' \wedge Sal > 30000, \\ m2: Loc = 'LA' \wedge Sal \leq 30000, \\ m3: Loc \neq 'LA' \wedge Sal > 30000, \\ m4: Loc \neq 'LA' \wedge Sal \leq 30000 \}$$

Nhận xét:

Các vị từ minterm tối giản và đầy đủ cho ứng dụng A
=> M phân đoạn EMP vào 4 đoạn.

Phương pháp phân đoạn ngang

- Ví dụ 3: Cho bảng PROJ (**PNO**, Pname, Funds, Dno, Loc)

Pno	Pname	Funds	Dno	Loc
P1	Requirements	135,000	D2	NY
P2	Design	310,000	D3	NY
P3	Code	300,000	D1	MPLS
P4	Documentation	450,000	D1	MPLS
P5	Testing	250,000	D4	LA

Giả sử ứng dụng A1 và A2 truy vấn bảng PROJ dựa trên các vị từ đơn giản sau:

Phương pháp phân đoạn ngang

- Ví dụ 3 (tt)
- Các vị từ đơn giản của A1:
 - p1: Loc = 'MPLS'
 - p2: Loc = 'NY'
 - p3: Loc = 'LA'
- Các vị từ đơn giản của A2:
 - p4: Funds <= 300000

=> PROJ được phân thành các đoạn nào ?

Phương pháp phân đoạn ngang

- Ví dụ 3 (tt)

- Kết hợp các vị từ đơn giản của A1 và A2 ta có

Pr={ p1: Loc = 'MPLS'

p2: Loc = 'NY'

p3: Loc = 'LA'

p4: Funds \leq 300000 }

=> M có bao nhiêu vị từ minterm ?

- Pr có 4 vị từ đơn giản => M có $2^4 = 16$ vị từ minterm

Phương pháp phân đoạn ngang

- Ví dụ 3 (tt)

- Kết hợp các vị từ đơn giản của A1 và A2 ta có

Pr={ p1: Loc = 'MPLS'

p2: Loc = 'NY'

p3: Loc = 'LA'

p4: Funds \leq 300000 }

=> M có bao nhiêu vị từ minterm ?

- Pr có 4 vị từ đơn giản => M có $2^4 = 16$ vị từ minterm

Phương pháp phân đoạn ngang

m1 = {Loc = 'MPLS' ^ Loc = 'NY' ^ Loc = 'LA' ^ Funds <= 300000} **s**

m2 = {Loc = 'MPLS' ^ Loc = 'NY' ^ Loc = 'LA' ^ Funds > 300000} **s**

m3 = {Loc = 'MPLS' ^ Loc = 'NY' ^ Loc <> 'LA' ^ Funds <= 300000} **s**

m4 = {Loc = 'MPLS' ^ Loc = 'NY' ^ Loc <> 'LA' ^ Funds > 300000} **s**

m5 = {Loc = 'MPLS' ^ Loc <> 'NY' ^ Loc = 'LA' ^ Funds <= 300000} **s**

m6 = {Loc = 'MPLS' ^ Loc <> 'NY' ^ Loc = 'LA' ^ Funds > 300000} **s**

m7 = {Loc = 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds <= 300000}

m8 = {Loc = 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds > 300000}

m9 = {Loc <> 'MPLS' ^ Loc = 'NY' ^ Loc = 'LA' ^ Funds <= 300000} **s**

m10 = {Loc <> 'MPLS' ^ Loc = 'NY' ^ Loc = 'LA' ^ Funds > 300000} **s**

m11 = {Loc <> 'MPLS' ^ Loc = 'NY' ^ Loc <> 'LA' ^ Funds <= 300000}

m12 = {Loc <> 'MPLS' ^ Loc = 'NY' ^ Loc <> 'LA' ^ Funds > 300000}

m13 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc = 'LA' ^ Funds <= 300000}

m14 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc = 'LA' ^ Funds > 300000}

m15 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds <= 300000} **s**

m16 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds > 300000} **s**

Phương pháp phân đoạn ngang

- **Nhận xét 1:**

- Các vị từ đơn giản p_1 , p_2 , và p_3 là loại trừ lẫn nhau. Nghĩa là chỉ p_1 hoặc p_2 hoặc p_3 có thể đúng và bất kỳ sự kết hợp của chúng bởi phép \wedge là sai
- Ví dụ: $Loc = 'LA'$ thì Loc không thể là $'NY'$ hay $'MPLS'$
=> Cần loại các m_i không hợp lệ
=> loại m_1 , m_2 , m_3 , m_4 , m_5 , m_6 , m_9 , và m_{10}

Phương pháp phân đoạn ngang

Các vị từ còn lại trong M

m7 = {Loc = 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds <= 300000}

m8 = {Loc = 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds > 300000}

m11 = {Loc <> 'MPLS' ^ Loc = 'NY' ^ Loc <> 'LA' ^ Funds <= 300000}

m12 = {Loc <> 'MPLS' ^ Loc = 'NY' ^ Loc <> 'LA' ^ Funds > 300000}

m13 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc = 'LA' ^ Funds <= 300000}

m14 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc = 'LA' ^ Funds > 300000}

m15 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds <= 300000}

m16 = {Loc <> 'MPLS' ^ Loc <> 'NY' ^ Loc <> 'LA' ^ Funds > 300000}

Nhận xét 2:

- m15 và m16 là các vị từ không hợp lệ, vì **Loc** phải là một trong ba giá trị được đề cập.

=> Loại bỏ m15 và m16

Phương pháp phân đoạn ngang

Các vị từ còn lại trong M

$m7 = \{Loc = 'MPLS' \wedge Loc \neq 'NY' \wedge Loc \neq 'LA' \wedge Funds \leq 300000\}$

$m8 = \{Loc = 'MPLS' \wedge Loc \neq 'NY' \wedge Loc \neq 'LA' \wedge Funds > 300000\}$

$m11 = \{Loc \neq 'MPLS' \wedge Loc = 'NY' \wedge Loc \neq 'LA' \wedge Funds \leq 300000\}$

$m12 = \{Loc \neq 'MPLS' \wedge Loc = 'NY' \wedge Loc \neq 'LA' \wedge Funds > 300000\}$

$m13 = \{Loc \neq 'MPLS' \wedge Loc \neq 'NY' \wedge Loc = 'LA' \wedge Funds \leq 300000\}$

$m14 = \{Loc \neq 'MPLS' \wedge Loc \neq 'NY' \wedge Loc = 'LA' \wedge Funds > 300000\}$

Phương pháp phân đoạn ngang

Nhận xét 3: Vì Loc chỉ có thể nhận một giá trị tại một thời điểm nên chúng ta có thể đơn giản hóa các vị từ như sau:

$$m7 = \{Loc = 'MPLS' \wedge Funds \leq 300000\}$$

$$m8 = \{Loc = 'MPLS' \wedge Funds > 300000\}$$

$$m11 = \{Loc = 'NY' \wedge Funds \leq 300000\}$$

$$m12 = \{Loc = 'NY' \wedge Funds > 300000\}$$

$$m13 = \{Loc = 'LA' \wedge Funds \leq 300000\}$$

$$m14 = \{Loc = 'LA' \wedge Funds > 300000\}$$

- Áp dụng các vị từ minterm vào PROJ, **5 đoạn được tạo ra**
- Mặc dù có sáu vị từ, nhưng chỉ có năm trong số chúng tạo ra đoạn có nghĩa từ PROJ.
- Đối với tình trạng đã cho của bảng PROJ, chỉ có năm đoạn trên thực sự chứa các dòng, đoạn thứ sáu không tồn tại (m14).

Phương pháp phân đoạn ngang

Pno	Pname	Funds	Dno	Loc
P3	Code	300,000	D1	MPLS

PROJ1: được sinh ra do áp dụng m7
 $PROJ1 = \sigma_{Loc = 'MPLS' \wedge Funds \leq 300000} (PROJ)$

Pno	Pname	Funds	Dno	Loc
P4	Documentation	450,000	D1	MPLS

PROJ2: được sinh ra do áp dụng m8

Pno	Pname	Funds	Dno	Loc
P1	Requirements	135,000	D2	NY

PROJ3: được sinh ra do áp dụng m11

Pno	Pname	Funds	Dno	Loc
P2	Design	310,000	D3	NY

PROJ4: được sinh ra do áp dụng m12

Pno	Pname	Funds	Dno	Loc
P5	Testing	250,000	D4	LA

PROJ5: được sinh ra do áp dụng m13

Bài tập phân đoạn ngang

- **Bài 1:** Cho bảng đầu bếp DAUBEP như bên dưới, hãy đề nghị một sơ đồ phân đoạn ngang bằng cách sử dụng các câu truy vấn sau:

R1: SELECT ID,NamKN FROM DAUBEP WHERE ten='Jean' AND ho LIKE '%R%';

R2: SELECT * FROM DAUBEP WHERE NamKN = 1;

R3: SELECT ID, ho FROM DAUBEP WHERE NamKN =2 AND ten = 'Jean'

ID	Họ	Tên	namKN
12	DUPONT	Jean	1
34	DUPONT	Jeanne	2
17	DUBOIS	Robert	1
22	DUBALAI	Aline	1
26	DUGENOU	Jean	2
11	DURAND	Aline	2
38	DURACUIRE	ROBERT	2
9	DURALUMIN	Roberte	1
13	DURDUR	Jean	2
20	DURALEX	Jean	1

Bài tập phân đoạn ngang

- Bài 2:** Xem lại bảng TaiKhoan, hãy đề nghị một sơ đồ phân đoạn ngang bằng cách sử dụng các câu truy vấn sau:

$$R1 = \pi_{MaKH, TenCN} (\sigma_{(loai = 'The') \wedge (sodu > 100\ 000)} (TaiKhoan))$$

$$R3 = \pi_{MaKH, sodu} (\sigma_{(TenCN = 'Genève') \wedge (Loai = 'The')} (TaiKhoan))$$

$$R2 = \sigma_{TenCN = 'Lausanne'} (TaiKhoan)$$

TaiKhoan

MaKH	TenCN	Loai	sodu
174 723	Lausanne	The	123 345.89
177 498	Genève	The	34 564.00
201 639	Lausanne	The	45 102.50
178 123	Lausanne	Tietkiem	325 100.00
203 446	Genève	The	274 882.95

- **Bài 3** Cho bảng sinh viên như hình. Có 2 ứng dụng chạy trên 2 site truy cập đến bảng SINHVIEN thực hiện thường xuyên 2 câu truy vấn. Hãy đề nghị một sơ đồ phân đoạn ngang bằng cách sử dụng 2 câu truy vấn sau:

Site1: Select hoten FROM SINHVIEN WHERE tinh/TP='Nantes' and ĐTB >=12

Site2: Select hoten, ĐTB FROM SINHVIEN WHERE ĐTB>=16

<u>hoten</u>	<u>Tinh/TP</u>	ĐTB
Patrick	Nantes	14.5
<u>Bellot</u>	Nice	16.1
Phillip	Nantes	18.5
Christophe	Nice	9.6
Nathalie	Nantes	11.2

- **Bài 4** Cho bảng PHICONG như hình. Có 3 ứng dụng chạy trên 3 site truy cập đến bảng PHICONG thực hiện thường xuyên 3 câu truy vấn. Hãy đề nghị một sơ đồ phân đoạn ngang bằng cách sử dụng 3 câu truy vấn sau:

Site1: Select * FROM PHICONG WHERE nuoc = 'VN'

Site2: Select * FROM PHICONG WHERE nuoc='P' AND km>=20000

Site2: Select * FROM PHICONG WHERE km <20000

MPC	Hoten	Nuoc	km
1	A	VN	20000
2	B	P	25000
3	C	VN	15000
4	D	M	30000