

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



Semi-join

- Một semi-join giữa hai bảng **trả về các dòng của bảng đầu tiên** mà một hoặc nhiều dòng thích hợp được tìm thấy trong bảng thứ hai.
- Sự khác biệt giữa một join và semi-join:
 - Các dòng trong bảng đầu tiên sẽ được trả về **nhieu nhất một lần**.
 - Ngay cả khi **bảng thứ hai chứa nhiều dòng trùng khớp** với một dòng trong bảng đầu tiên, **chỉ một dòng sẽ được trả về**.
- **Ký hiệu:** \bowtie

Semi-join

$$R1 \ltimes R2 = \pi_{\text{Các thuộc tính của } R1}(R1 \bowtie R2)$$

- Trong **SQL**, phép semi-join được viết bằng cách dùng **EXISTS** hoặc **IN**.

Semi-join

- **Ví dụ:** Tìm tên và thành phố của khách hàng mà có ít nhất 1 tài khoản

JOIN:

```
Select  cname, ccity  
From Customer c, Account a  
Where c.cid = a.cid  
Order by cname;
```

- một KH có nhiều tài khoản sẽ có nhiều dòng trong kết quả
- Dùng DISTINCT để loại bỏ các dòng trùng nhau

Semi-join

- **Ví dụ:** Tìm tên và thành phố của khách hàng mà có ít nhất 1 tài khoản

SEMI-JOIN:

Select cname, ccity

From Customer

Where cid IN (Select cid FROM Account)

Order by cname;

- Không có KH nào có nhiều hơn hai dòng trong kết quả
- Hệ quản trị CSDL dừng việc xử lý 1 KH ngay khi có một KH được đưa vào kết quả

Semi-join

- **Ví dụ:** Tìm tên và thành phố của khách hàng mà có ít nhất 1 tài khoản

ĐSQH:

$\pi_{\text{name, ccity}} (\text{Customer} \bowtie \text{Account})$

Semi-join

- **Bài tập** Cho các quan hệ:

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
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

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

Tính:

EMP ✕ ASG ?
PROJ ✕ ASG ?

Phân đoạn ngang dẫn xuất (Derived fragmentation)

- *Một quan hệ được phân đoạn dựa trên các ràng buộc được định nghĩa trên một quan hệ khác.*
- *Cả hai quan hệ được liên kết với nhau với khoá chính và khoá ngoại*
- Hai quan hệ phải thiết lập mỗi quan hệ owner (sở hữu) và member (thành viên)
 - Quan hệ owner là quan hệ cha
 - Quan hệ member là quan hệ con

Phân đoạn ngang dẫn xuất

- **Ba điều kiện** để thực hiện phân đoạn ngang dẫn xuất
 - Tập các đoạn ngang của quan hệ 'owner'
 - Ví dụ ($F_1, F_2 \dots$).
 - Quan hệ 'member', là quan hệ cần được phân đoạn ngang theo quan hệ 'owner'.
 - Tập các điều kiện cho phép semi-join giữa owner và member
 - Ví dụ: $SINHVIEN.MASV = HOC.MASV$

Phân đoạn ngang dẫn xuất

- Quan hệ owner R phân thành các đoạn ngang $F = \{F_1, F_2, \dots\}$.
 - F có thể là phân đoạn ngang chính hoặc dẫn xuất.
 - Quan hệ S được phân đoạn ngang dẫn xuất theo F thành các đoạn như sau :

$$R, \mathbf{F} = \{ F_1, F_2, \dots \}$$

\Downarrow

$$S, \mathbf{G} = \{ G_1, G_2, \dots \}, G_i = S \ltimes F_i$$

Phân đoạn ngang dẫn xuất

- Ví dụ: Cho các quan hệ:

$\mathbf{F} = \{ F_1, F_2 \}$ theo vị trí (location)

$\mathbf{E}(\text{id}, \text{name}, \text{salary}, \text{location})$

$\mathbf{T}(\text{id}, \text{task})$

Và câu truy vấn:

Tên nhân viên và danh sách các công việc mà nhân viên làm ?

Phân đoạn ngang dẫn xuất

- Ví dụ:

E_1

id	name	location	salary
1	Tom	A	15
3	Ben	A	21

$$E_1 = E \bowtie F_1$$

E_2

id	name	location	salary
2	Ann	B	23
4	Max	B	17

$$E_2 = E \bowtie F_2$$

T

id	task
1	design
1	build
2	advertise
4	sell

Phân đoạn ngang dẫn xuất

- Ví dụ:

id	name	location	salary
1	Tom	A	15
3	Ben	A	21

E_1

id	name	location	salary
2	Ann	B	23
4	Max	B	17

E_2

id	task
1	design
1	build

T_1

$$T_1 = T \bowtie E_1$$

id	task
2	advertise
4	sell

T_2

$$T_2 = T \bowtie E_2$$

Phân đoạn ngang dẫn xuất

- **Bài tập 1** Cho các quan hệ bên dưới, tính
 - Phân đoạn ChiNhanh theo tên chi nhánh, Phân đoạn dẫn xuất TaiKhoan theo ChiNhanh
 - Phân đoạn Khachhang theo độ tuổi (>30 và ≤ 30), Phân đoạn TaiKhoan theo Khachhang

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

Phân đoạn ngang dẫn xuất

- Bài tập 2:** Cho các quan hệ bên dưới, tính:

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

- Phân đoạn PAY theo SAL(≤ 30.000 và > 30.000) ?
- Phân đoạn dẫn xuất EMP theo PAY ?
- Phân đoạn dẫn xuất ASG theo phân đoạn EMP ?

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

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

Phân đoạn ngang dẫn xuất

- **Giải Bài tập 2:** Cho các quan hệ bên dưới, tính

–

$$EMP_1 = EMP \bowtie PAY_1$$

$$EMP_2 = EMP \bowtie PAY_2$$

với:

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

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

EMP₁

ENO	ENAME	TITLE
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E7	R. Davis	Mech. Eng.

EMP₂

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E8	J. Jones	Syst. Anal.

Phân đoạn ngang dẫn xuất

- **Giải bài tập 2:** Cho các quan hệ bên dưới, tính

–

$$ASG_1 = ASG \times EMP_1$$

$$ASG_2 = ASG \times EMP_2$$

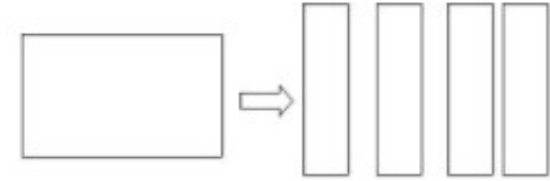
ASG₁

ENO	PNO	RESP	DUR
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E7	P3	Engineer	36

ASG₂

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E5	P2	Manager	24
E6	P4	Manager	48
E8	P3	Manager	40

Phương pháp phân đoạn dọc



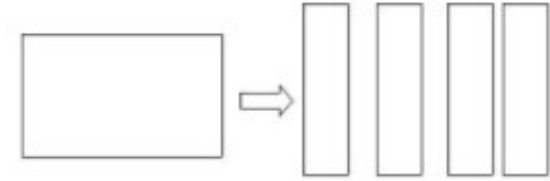
- Phân đoạn theo chiều dọc vốn **phức tạp** hơn phân đoạn theo chiều ngang
- ***Phân đoạn ngang***: Nếu Pr có N predicate đơn giản thì M sẽ có 2^N Minterm (một số minterm có thể loại bỏ).
- ***Phân đoạn dọc***: Nếu có m thuộc tính non-primary key, số đoạn có thể bằng $B(m)$ (số Bell)
 - Ví dụ $B(3)=5$, $B(4)=15$
 - Nếu m lớn $B(m) \approx m^m$, ví dụ : $B(15) = 10^9$

=> Các giải pháp tối ưu là không khả thi, cần phải được áp dụng heuristic

Số Bell

- Trong lý thuyết tổ hợp của toán học, số Bell thứ n là số các phân hoạch của tập gồm n phần tử.
- $B_0 = B_1 = 1$
- Các số Bell đầu tiên là 1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975
- Ví dụ: tập $\{1,2,3\}$ có thể được phân chia theo 5 cách:
 - $\{\{1\}, \{2\}, \{3\}\},$
 - $\{\{1,2\}, \{3\}\},$
 - $\{\{1,3\}, \{2\}\},$
 - $\{\{1\}, \{2,3\}\},$
 - $\{\{1,2,3\}\},$ \Rightarrow vì vậy $B_3 = 5$

Phương pháp phân đoạn dọc



- Hai loại heuristic cho phân đoạn theo chiều dọc:

Nhóm (grouping)



Tách (splitting)



Phương pháp Nhóm

- **Nhóm** là một cách tiếp cận bắt đầu bằng cách
 - Tạo nhiều đoạn dọc có thể có, và
 - Sau đó từng bước giảm số đoạn bằng việc trộn (merge) các đoạn lại với nhau.
 - Tiếp cận dưới lên
- **Bước 1:** Tạo mỗi đoạn cho mỗi cột nonkey bằng cách đặt cột nonkey và khóa chính của bảng vào mỗi đoạn dọc.
 - Bước này tạo số đoạn dọc bằng số cột nonkey có trong bảng

=> Mức độ phân đoạn là quá mỏng và không thực tế.

Phương pháp Nhóm

- **Bước 2:** Tiếp cận này sau đó
 - Sử dụng kết nối (JOIN) trên khóa chính để nhóm các đoạn với nhau, và
 - Tiếp tục quá trình này cho đến khi đạt được thiết kế mong muốn.
- Tuy nhiên, ***nhóm*** thường không được coi là một cách tiếp cận hợp lệ cho việc thiết kế phân đoạn dọc vì ***cùng cột nonkey có thể tham gia vào nhiều hơn một nhóm***

Phương pháp tách - splitting

- Bắt đầu với một quan hệ và quyết định việc phân chia dựa trên hành vi truy cập của các ứng dụng trên các thuộc tính.
- Tiếp cận từ trên xuống
- Kết quả trong các đoạn không trùng nhau
- Chỉ phân đoạn theo chiều dọc được xem xét

Phương pháp tách - splitting

- Việc tách chỉ được áp dụng cho các thuộc tính không tham gia vào khóa chính.
- Tiếp cận tách bao gồm ba bước:
 - 1.** Xây dựng ma trận affinity (mối quan hệ) thuộc tính (AA - attribute affinity matrix); Ma trận này cho biết các thuộc tính liên quan chặt chẽ nhau như thế nào
 - 2.** Sử dụng một giải thuật gom cụm (clustering algorithm) để nhóm một số thuộc tính cùng nhau dựa trên ma trận affinity thuộc tính. Giải thuật này sinh ra ma trận affinity gom cụm (CA- clustered affinity matrix)
 - 3.** Sử dụng giải thuật phân rã để phân rã các thuộc tính.

Các định nghĩa

- **Tần suất truy cập:** tần suất mà ứng dụng truy cập dữ liệu.
 - Nếu $Q = \{q_1; q_2; \dots; q_q\}$ là tập các câu truy vấn của người dùng,
 - Thì $\text{acc}(q_i)$ cho biết tần suất truy cập của câu truy vấn q_i trong một khoảng thời gian nhất định.
- **Độ đo affinity các thuộc tính (AA - attribute affinity):** Cho biết mức độ liên quan chặt chẽ của các thuộc tính. Thường độ đo này không dễ dàng nhận biết được => sử dụng giải thuật

Bước 1 - Tạo ma trận AA

- Xây dựng ma trận AA từ ma trận sử dụng thuộc tính (AU - Attribute Usage Matrix) với
 - Cho $Q = \{q_1 ; q_2; ...; q_q\}$ là tập các câu truy vấn (ứng dụng) truy cập quan hệ R ($A_1, A_2, ..., A_n$).
 - Đối với mỗi truy vấn q_i và mỗi thuộc tính A_j , chúng ta kết hợp thành một giá trị sử dụng thuộc tính, ký hiệu $use(q_i, A_j)$, và được định nghĩa như sau:

$$use(q_i, A_j) = \begin{cases} 1 & \text{nếu } q_i \text{ tham chiếu } A_j \\ 0, & \text{ngược lại} \end{cases}$$

Bước 1 - Tạo ma trận AA

- Ví dụ

PROJ	<u>PNO</u>	PNAME	BUDGET	LOC
	A_1	A_2	A_3	A_4

q1: SELECT BUDGET

FROM PROJ

WHERE PNO=Value;

q2: SELECT PNAME, BUDGET

FROM PROJ;

q3: SELECT PNAME

FROM PROJ

WHERE LOC=Value;

q4: SELECT SUM(BUDGET)

FROM PROJ

WHERE LOC=Value

=>

	A_1	A_2	A_3	A_4
q_1	1	0	1	0
q_2	0	1	1	0
q_3	0	1	0	1
q_4	0	0	1	1

Ma trận AU

Bước 1 - Tạo ma trận AA

- Độ đo AA giữa hai thuộc tính A_i và A_j của của quan hệ $R(A_1, A_2, \dots, A_n)$ đối với tập ứng dụng $Q = \{q_1 ; q_2; \dots; q_q\}$ được định nghĩa như sau:

$$aff(A_i, A_j) = \sum_{k | use(q_k, A_i)=1 \wedge use(q_k, A_j)=1} \sum_l ref_l(q_k) acc_l(q_k)$$

Aff (A_i , A_j): số lần hai thuộc tính được truy cập cùng nhau, được xem xét trên tất cả các site

- Trong đó:
 - $ref_l(q_k)$: Số lần truy cập vào thuộc tính (A_i, A_j) cho mỗi lần thực hiện q_k tại vị trí l
 - $acc_l(q_k)$: tần suất truy cập ứng dụng của q_k tại vị trí l .

Bước 1 - Tạo ma trận AA

- **Ví dụ:** Xây dựng ma trận AA biết:
 - $\text{ref}_l(q_k) = 1$ với tất cả k tại site l .
 - tần xuất ứng dụng (truy cập) $\text{acc}_l(q_k)$ được cho như sau:

	S_1	S_2	S_3
q_1	15	20	10
q_2	5	0	0
q_3	25	25	25
q_4	3	0	0

Bước 1 - Tạo ma trận AA

	A_1	A_2	A_3	A_4
q_1	1	0	1	0
q_2	0	1	1	0
q_3	0	1	0	1
q_4	0	0	1	1

Ma trận AU

	S_1	S_2	S_3
q_1	15	20	10
q_2	5	0	0
q_3	25	25	25
q_4	3	0	0

Ma trận AA

	A_1	A_2	A_3	A_4
A_1	45	0	45	0
A_2	0	80	5	75
A_3	45	5	53	3
A_4	0	75	3	78

$$aff(A_i, A_j) = \sum_{k | use(q_k, A_i)=1 \wedge use(q_k, A_j)=1} \sum_{l} ref_l(q_k) acc_l(q_k)$$

- $aff(A_1, A_2) = aff(A_1, A_4) = 0$ vì không có ứng dụng truy cập cả 2 thuộc tính.

$$aff(A_1, A_3) = \sum_{k=1}^1 \sum_{l=1}^3 acc_l(q_k) = acc_1(q_1) + acc_2(q_1) + acc_3(q_1)$$

$$= 15*1 + 20*1 + 10*1 = 45 \text{ (Vì chỉ có } q_1 \text{ truy xuất cả 2 thuộc tính)}$$

Bước 2 - Giải thuật gom cụm

- Sử dụng ma trận affinity thuộc tính AA và tổ chức lại thứ tự các thuộc tính để tạo thành các cụm, trong đó các thuộc tính trong mỗi cụm thể hiện mối quan hệ cao với nhau.
- Thuật toán năng lượng liên kết (BEA - Bond Energy Algorithm) được sử dụng để nhóm các thuộc tính. BEA tìm thứ tự các thuộc tính để độ đo affinity toàn cục (AM - global affinity measure) là lớn nhất.
 - BEA nhận đầu vào là ma trận AA, hoán vị các dòng và cột của nó và sinh ra ma trận CA

Bước 2 - Giải thuật gom cụm

- Giải thuật BEA - Tạo ma trận CA gồm 3 bước
 - B1. Khởi tạo.** Đặt và cố định một trong các cột của AA vào CA. Cột 1 được lựa chọn trong thuật toán.
 - B2. Lặp.** Lấy mỗi cột trong số $n-i$ cột còn lại (i là số cột đã đặt trong CA) và cố gắng đặt chúng trong $i+1$ vị trí còn lại trong ma trận CA. Đối với mỗi cột, chọn đặt sao cho **sự đóng góp** (cont - contribution) cho độ đo affinity láng giềng toàn cục lớn nhất.
 - B3. Thứ tự dòng.** Khi thứ tự cột được xác định, vị trí của các dòng cũng phải được thay đổi để vị trí tương đối của dòng phù hợp với vị trí tương đối của các cột
- => độ phức tạp giải thuật $O(n*n)$ với n là số thuộc tính

Bước 2 - Giải thuật gom cụm

Algorithm BEA Algorithm (McCormick et al.,1972)

Input: AA: attribute affinity matrix

Output: CA: clustered affinity matrix

begin

 {initialize; remember that AA is an $n \times n$ matrix }

$CA(\bullet, 1) \leftarrow AA(\bullet, 1)$;

$CA(\bullet, 2) \leftarrow AA(\bullet, 2)$;

$index \leftarrow 3$;

while $index \leq n$ **do** {choose the “best” location for attribute AA_{index} }

for i from 1 to $index - 1$ by 1 **do** calculate $cont(A_{i-1}, A_{index}, A_i)$;

 calculate $cont(A_{index-1}, A_{index}, A_{index+1})$; {boundary condition}

$loc \leftarrow$ placement given by maximum $cont$ value ;

for j from $index$ to loc by -1 **do**

$CA(\bullet, j) \leftarrow CA(\bullet, j - 1)$ {shuffle the two matrices}

$CA(\bullet, loc) \leftarrow AA(\bullet, index)$;

$index \leftarrow index + 1$

 order the rows according to the relative ordering of columns

end

Bước 2 - Giải thuật gom cụm

- Sự đóng góp (cont) của cột **khi đặt A_k** giữa A_i và A_j :

$$\text{cont}(A_i, A_k, A_j) = 2\text{bond}(A_i, A_k) + 2\text{bond}(A_k, A_j) - 2\text{bond}(A_i, A_j)$$

$$\text{bond}(A_x, A_y) = \sum_{z=1..n} \text{aff}(A_z, A_x) \text{aff}(A_z, A_y)$$

- **Ví dụ :**

Bước 2 - Giải thuật gom cụm - Ví dụ

- Sử dụng lại AA, tính sự đóng góp của A4 giữa A1 và A2.

	A_1	A_2	A_3	A_4
A_1	45	0	45	0
A_2	0	80	5	75
A_3	45	5	53	3
A_4	0	75	3	78

$$\begin{aligned} cont(A1, A4, A2) = & 2bond(A1, A4) \\ & + 2bond(A4, A2) \\ & - 2bond(A1, A2) \end{aligned}$$

$$bond(Ax, Ay) = \sum_{z=1..n} aff(Az, Ax) aff(Az, Ay)$$

$$\begin{aligned} bond(A1, A4) = & aff(A1, A1) aff(A1, A4) \\ & + aff(A2, A1) aff(A2, A4) \\ & + aff(A3, A1) aff(A3, A4) \\ & + aff(A4, A1) aff(A4, A4) \end{aligned}$$

$$bond(A1, A4) = 45*0 + 0*75 + 45*3 + 0*78 = 135$$

$$bond(A4, A2) = 11865$$

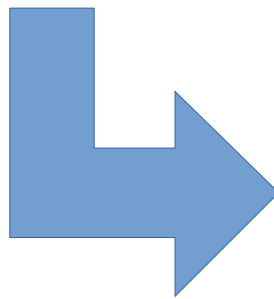
$$bond(A1, A2) = 225$$

$$cont(A1, A4, A2) = 2 * 135 + 2 * 11865 - 2 * 225 = 23550$$

Cách tính Bond

Ví dụ:

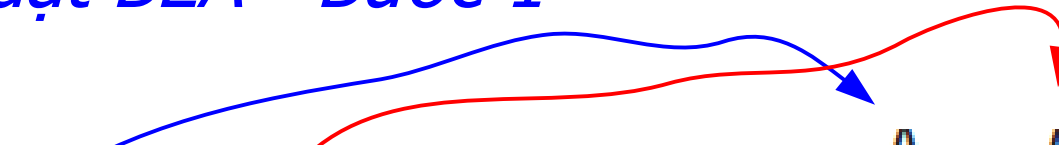
$$\begin{aligned} \text{bond}(A1, A4) &= \text{aff}(A1, A1) \text{aff}(A1, A4) \\ &+ \text{aff}(A2, A1) \text{aff}(A2, A4) \\ &+ \text{aff}(A3, A1) \text{aff}(A3, A4) \\ &+ \text{aff}(A4, A1) \text{aff}(A4, A4) = 135 \end{aligned}$$



	A1	A4	Tích
A1	45	0	0
A2	0	75	0
A3	45	3	135
A4	0	78	0
Tổng			135

Bước 2 - Giải thuật gom cụm - Ví dụ

- Giải thuật BEA - Bước 1



	A_1	A_2	A_3	A_4
A_1	45	0	45	0
A_2	0	80	5	75
A_3	45	5	53	3
A_4	0	75	3	78

AA

	A_1	A_2
A_1	45	0
A_2	0	80
A_3	45	5
A_4	0	75

CA

- BEA - Bước 2: Đặt A3** - Có 3 cách đặt A3

Bước 2 - Giải thuật gom cụm - Ví dụ

- **Có 3 cách đặt A3:**
 - Đặt A3 bên trái A1 => $\text{cont}(A0, A3, A1)$?
 - Đặt A3 giữa A1 và A2 => $\text{cont}(A1, A3, A2)$?
 - Đặt A3 bên phải A2 => $\text{cont}(A2, A3, A_n)$?
- Vì *A3 là cột đang được xét nên nó luôn phải ở giữa trong công thức $\text{cont}()$* . Do đó:
 - Khi đặt A3 ở cột trái nhất ma trận sẽ không có cột bên trái nó, trường hợp này cột **A0** được sử dụng để biểu diễn cho cột không tồn tại tại trái
 - Tương tự **A_n** được sử dụng để biểu diễn cho cột không tồn tại bên phải

$$\text{bond}(A0, A_i) = \text{bond}(A_i, A_n) = 0$$

Với 0 và n là cột không tồn tại của CA

Bước 2 - Giải thuật gom cụm - Ví dụ

- Có 3 cách đặt A3:**

- $cont(A0, A3, A1) = 2bond(A0, A3) + 2bond(A3, A1) - 2bond(A0, A1)$
 $= 2bond(A3, A1) = 8820$

- $cont(A1, A3, A2) = 2bond(A1, A3) + 2bond(A3, A2) - 2bond(A1, A2)$
 $= 4410 + 890 + 225 = 10.150$

- $cont(A2, A3, An) = 2bond(A2, A3) + 2bond(A3, An) - 2bond(A2, An)$
 $= 2bond(A2, A3) = 1780$

Từ các kết quả trên => chọn Đặt A3 giữa A1 và A2

Bước 2 - Giải thuật gom cụm - Ví dụ

	A_1	A_2		A_1	A_3	A_2
A_1	45	0		A_1	45	0
A_2	0	80		A_2	0	5
A_3	45	5		A_3	45	53
A_4	0	75		A_4	0	3

- **BEA - Bước 2: Đặt A4** - Có 4 cách đặt A4

Bước 2 - Giải thuật gom cụm - Ví dụ

- **Có 4 cách đặt A4:**

- Đặt A4 bên trái A1 $\Rightarrow cont(A0, A4, A1) ? 270$
- Đặt A4 giữa A1 và A3 $\Rightarrow cont(A1, A4, A3) ? -7014$
- Đặt A4 giữa A3 và A2 $\Rightarrow cont(A3, A4, A2) ? 23486$
- Đặt A4 bên phải A2 $\Rightarrow cont(A2, A4, A_n) ? 23730$

\Rightarrow Đặt A4 bên phải A2

	A ₁	A ₃	A ₂	A ₄
A ₁	45	45	0	0
A ₂	0	5	80	75
A ₃	45	53	5	3
A ₄	0	3	75	78

Bước 2 - Giải thuật gom cụm - Ví dụ

- BEA - Bước 3:** Các dòng được tổ chức lại như các cột. Kết quả ma trận CA như hình.

	A ₁	A ₃	A ₂	A ₄
A ₁	45	45	0	0
A ₂	0	5	80	75
A ₃	45	53	5	3
A ₄	0	3	75	78

Sắp xếp lại hàng

	A ₁	A ₃	A ₂	A ₄
A ₁	45	45	0	0
A ₃	45	53	5	3
A ₂	0	5	80	75
A ₄	0	3	75	78

Hai cụm:

- Cụm góc trên bên trái với giá trị nhỏ hơn
- Cụm góc dưới bên phải với giá trị lớn hơn

Bước 2 - Giải thuật gom cụm - Ví dụ

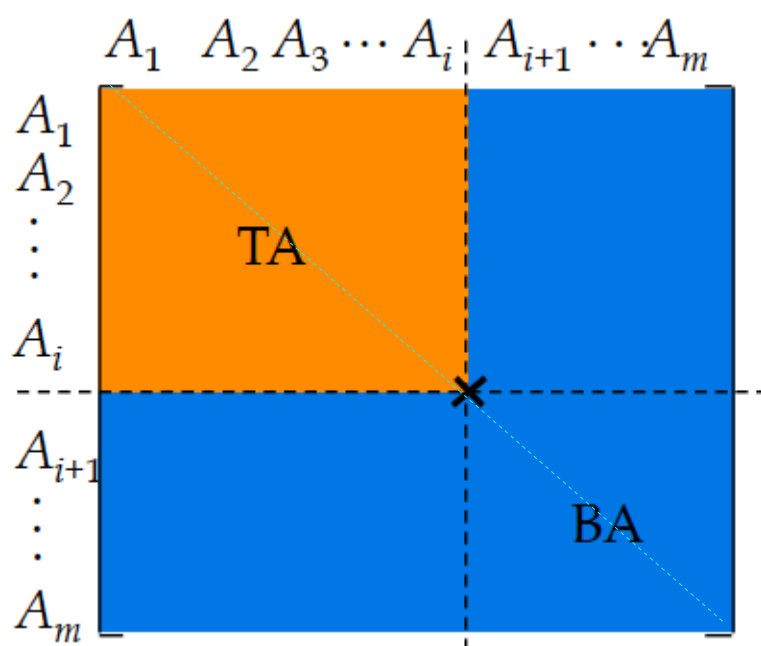
- **Nhận xét:**

- *Sự phân cụm này chỉ ra cách các thuộc tính của quan hệ PROJ nên được tách ra.*
- Tuy nhiên, nói chung biên giới cho sự phân chia này có thể không được rõ ràng. Khi ma trận CA lớn, thường có nhiều hơn hai cụm được hình thành và có nhiều hơn một cách phân đoạn.

=> Cần phải có một tiếp cận để giải quyết vấn đề này một cách có hệ thống hơn.

Bước 3 - Giải thuật phân rã

- Làm thế nào để chia một tập các thuộc tính được gom cụm $\{A_1, A_2, \dots, A_n\}$ thành hai tập (hoặc nhiều hơn hai) $\{A_1, A_2, \dots, A_i\}$ và $\{A_i, \dots, A_n\}$ sao cho **không có** (hoặc tối thiểu) các ứng dụng truy cập **vào cả hai tập** (hoặc nhiều hơn hai) ?



- Lấy **điểm bất kỳ** thuộc đường chéo, điểm này chia đường chéo thành 2 phần:
 - tập các điểm góc trên bên trái gọi là tập *top* các thuộc tính, ký hiệu **TA**
 - tập các điểm góc dưới bên phải gọi là tập *bottom* các thuộc tính, ký hiệu **BA**
- Nếu quan hệ có n thuộc tính thì có $n-1$ vị trí trên đường chéo ma trận CA mà điểm này có thể được đặt.
=> Vấn đề tối ưu hoá, đặt điểm ở đâu ?

Bước 3 - Giải thuật phân rã

- Định nghĩa
 - TQ = tập các ứng dụng chỉ truy cập TA
 - BQ = tập các ứng dụng chỉ truy cập BA
 - OQ = tập các ứng dụng truy cập cả TA và BA
 - CTQ = tổng số lần truy cập vào các thuộc tính bởi các ứng dụng mà chỉ truy cập TA
 - CBQ = tổng số lần truy cập vào các thuộc tính bởi các ứng dụng mà chỉ truy cập BA
 - COQ = tổng số lần truy cập vào các thuộc tính bởi các ứng dụng truy cập cả TA và BA
- Tìm điểm thuộc đường chéo sao cho
 $CTQ * CBQ - COQ^2$ là lớn nhất

Bước 3 - Giải thuật phân rã

- Hàm mục tiêu: $CTQ * CBQ - COQ^2$ với

$$CTQ = \sum_{q_i \in TQ} \sum_{\forall S_j} ref_j(q_i) acc_j(q_i)$$

$$CBQ = \sum_{q_i \in BQ} \sum_{\forall S_j} ref_j(q_i) acc_j(q_i)$$

$$COQ = \sum_{q_i \in OQ} \sum_{\forall S_j} ref_j(q_i) acc_j(q_i)$$

Bước 3 - Giải thuật phân rã

Có hai vấn đề cần được giải quyết:

(1) Cụm hình thành giữa ma trận CA

- Dịch chuyển lên một hàng và một cột bên trái và áp dụng thuật toán để tìm điểm phân vùng "tốt nhất"
- Làm điều này cho tất cả các phép dịch chuyển có thể
- Độ phức tạp $O(m^2)$

Bước 3 - Giải thuật phân rã

Có hai vấn đề cần được giải quyết:

(2) Có hơn 2 cụm

- m cách phân vùng: tìm m điểm tách đồng thời
- Thử 1, 2, ..., m-1 điểm phân chia thuộc đường chéo và cố gắng tìm điểm tốt nhất
- Độ phức tạp $O(2^m)$

Bước 3 - Giải thuật phân rã

Ví dụ:

Khi áp dụng giải thuật phân rã lên ma trận CA thu được cho quan hệ PROJ, kết quả là PROJ được phân thành 2 đoạn:

$$F_{\text{PROJ}} = \{\text{PROJ}_1, \text{PROJ}_2\}$$

trong đó: $\text{PROJ}_1 = \{A_1, A_3\}$

$$\text{PROJ}_2 = \{A_1, A_2, A_4\}$$

Nghĩa là:

$$\text{PROJ}_1 = \{\text{PNO}, \text{BUDGET}\}$$

$$\text{PROJ}_2 = \{\text{PNO}, \text{PNAME}, \text{LOC}\}$$

Bài tập

- Cho quan hệ TEMP(C, C1, C2, C3, C4) và các ứng dụng:

AP1: Select C1 from TEMP where C4 = 100;

AP2: Select C4 from TEMP;

AP3: Update TEMP set C3 = 15 where C2 = 50;

AP4: Update TEMP set C1 = 5 where C3 = 10;

- Tần số truy xuất được cho trong bảng.
Hãy đề nghị 1 cách phân đoạn dọc
quan hệ đã cho.

Tần số truy xuất acc				
	S1	S2	S3	S4
AP1	1	0	2	0
AP2	0	4	3	0
AP3	0	0	4	0
AP4	3	0	0	0