

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 **nhiều 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: \ltimes

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**.

CUSTOMER (**CID**, CNAME, STREET, CCITY);
BRANCH (**BNAME**, ASSETS, BCITY);
ACCOUNT (**A#**, CID, BNAME, BAL);
LOAN (**L#**, CID, BNAME, AMT);
TRANSACTION (**TID**, CID, A#, Date, AMOUNT);

Semi-join

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

CID	CNAME	STREET	CCITY
001	Lê Nguyên	Lê Lai	Can Tho
002	Huỳnh Huy	Lê Thái Tổ	Vinh Long
003	Trần Văn	Nguyễn T M Khai	Cà Mau

A#	CID	BNAME	BAL
000011110001	001	Can Tho	1000000
000011110002	001	Can Tho	3000000
000011110003	002	Vinh Long	1000000
000011110004	003	Cà Mau	5000000
000011110005	003	Cà Mau	6000000
000011110005	003	Cà Mau	6000000

JOIN:

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



CNAME	CCITY
Lê Nguyên	Can Tho
Lê Nguyên	Can Tho
Huỳnh Huy	Vinh Long
Trần Văn	Cà Mau
Trần Văn	Cà Mau
Trần Văn	Cà Mau

07/16/21 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) ;



CNAME	CCITY
Lê Nguyên	Can Tho
Huỳnh Huy	Vĩnh Long
Trần Văn	Cà Mau

- 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:

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

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

Tính:

EMP ✕ ASG ?
PROJ ✕ ASG ?

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

Semi-join

- **Bài tập** Cho các quan hệ:

EMP			ASG			
ENO	ENAME	TITLE	ENO	PNO	RESP	DUR
E1	J. Doe	Elect. Eng	E1	P1	Manager	12
E2	M. Smith	Syst. Anal.	E2	P1	Analyst	24
E3	A. Lee	Mech. Eng.	E2	P2	Analyst	6
E4	J. Miller	Programmer	E3	P3	Consultant	10
E5	B. Casey	Syst. Anal.	E3	P4	Engineer	48
E6	L. Chu	Elect. Eng.	E4	P2	Programmer	18
E7	R. Davis	Mech. Eng.	E5	P2	Manager	24
E8	J. Jones	Syst. Anal.	E6	P4	Manager	48
			E7	P3	Engineer	36
			E8	P3	Manager	40



EMP \bowtie **ASG**

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer

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)

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

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

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

–

$$ASG_1 = ASG \times EMP_1$$

$$ASG_2 = ASG \times EMP_2$$

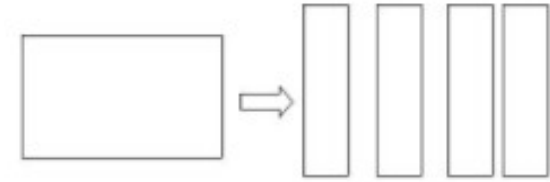
ASG₁

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



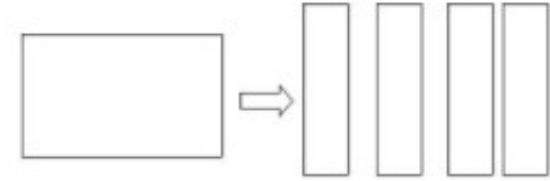
- Phân đoạn theo chiều dọc vốn **phức tạp** hơn phân đoạn theo chiều ngang
- ***Phân đoạn ngang***: Nếu Pr có N predicate đơn giản thì M sẽ có 2^N Minterm (một số minterm có thể loại bỏ).
- ***Phân đoạn dọc***: Nếu có m thuộc tính non-primary key, số đoạn có thể bằng $B(m)$ (số Bell)
 - Ví dụ $B(3)=5$, $B(4)=15$
 - 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 $R (A1, A2, ..., An)$.
 - 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:

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

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 Ak** giữa Ai và Aj:

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

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

$$\text{cont}(A_0, A_2, A_1) = 2\text{bond}(A_0, A_2) + \mathbf{2\text{bond}(A_2, A_1)} - 2\text{bond}(A_0, A_1)$$

$$\text{cont}(A_1, A_2, A_n) = \mathbf{2\text{bond}(A_1, A_2)} + 2\text{bond}(A_2, A_n) - 2\text{bond}(A_1, A_n)$$

- 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

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

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

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

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

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

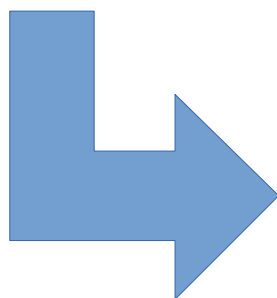
$$bond(A1, A2) = 225$$

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

Cách tính Bond

Ví dụ:

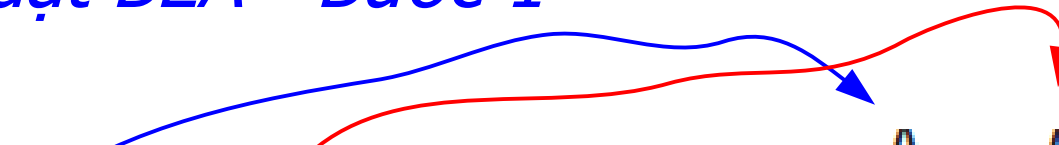
$$\begin{aligned} \text{bond}(A1, A4) &= \text{aff}(A1, A1) \text{aff}(A1, A4) \\ &+ \text{aff}(A2, A1) \text{aff}(A2, A4) \\ &+ \text{aff}(A3, A1) \text{aff}(A3, A4) \\ &+ \text{aff}(A4, A1) \text{aff}(A4, A4) = 135 \end{aligned}$$



	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

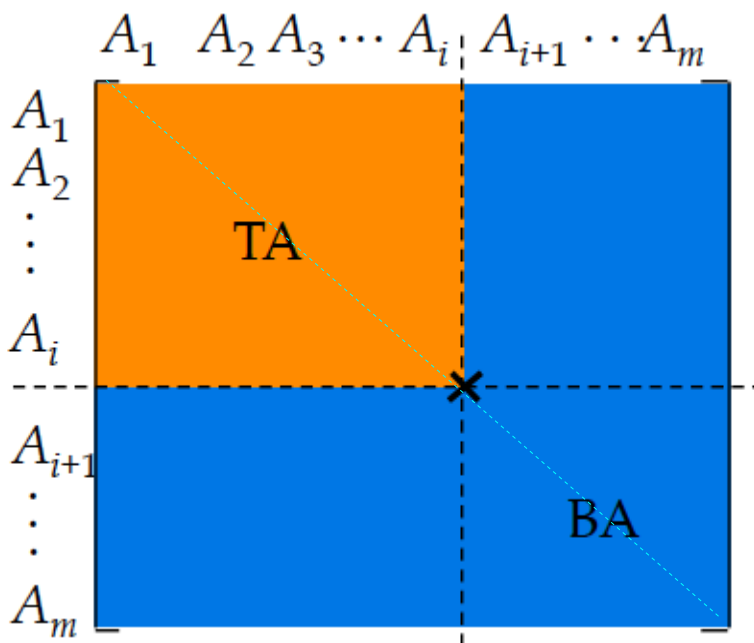
- **Nhận xét:**

- *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 $\{A_1, A_2, \dots, A_m\}$ thành hai tập (hoặc nhiều hơn hai) $\{A_1, A_2, \dots, A_i\}$ và $\{A_i, \dots, A_m\}$ 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/truy vấn** 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
$$Z = CTQ * CBQ - COQ^2$$
 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

B1. Xây dựng ma trận AA từ AU

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;

	C1	C2	C3	C4
AP1	1	0	0	1
AP2	0	0	0	1
AP3	0	1	1	0
AP4	1	0	1	0

B1. Xây dựng ma trận AA

AU

	C1	C2	C3	C4	Tần xuất truy cập				Tổng
					S1	S2	S3	S4	
AP1	1	0	0	1	1	0	2	0	3
AP2	0	0	0	1	0	4	3	0	7
AP3	0	1	1	0	0	0	4	0	4
AP4	1	0	1	0	3	0	0	0	3

AA



	C1	C2	C3	C4
C1	6	0	3	3
C2	0	4	4	0
C3	3	4	7	0
C4	3	0	0	10

B2. Xây dựng ma trận CA

- Cố định C1 và C2 trong CA, tìm vị trí C3

$$\text{Cont}(C0, C3, C1) = 2 * 0 + 2 * 39 - 2 * 0 = 78$$

$$\text{Cont}(C1, C3, C2) = 2 * 39 + 2 * 44 - 2 * 12 = 142$$

$$\text{Cont}(C2, C3, Cn) = 2 * 44 + 2 * 0 - 2 * 0 = 88$$

=> C3 đặt giữa C1 và C2

- Tìm vị trí C4

$$\text{Cont}(C0, C4, C1) = 2 * 0 + 2 * 48 - 2 * 0 = 96$$

$$\text{Cont}(C1, C4, C3) = 2 * 48 + 2 * 9 - 2 * 39 = 36$$

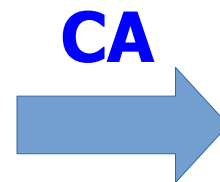
$$\text{Cont}(C3, C4, C2) = 2 * 9 + 2 * 0 - 2 * 44 = -70$$

$$\text{Cont}(C2, C4, Cn) = 2 * 0 + 2 * 0 - 2 * 0 = 0$$

=> C4 trước C1

B2. Xây dựng ma trận CA

	C4	C1	C3	C2
C1	3	6	3	0
C2	0	0	4	4
C3	0	3	7	4
C4	10	3	0	0



	C4	C1	C3	C2
C4	10	3	0	0
C1	3	6	3	0
C3	0	3	7	4
C2	0	0	4	4

B3. Phân rã

- Hàm mục tiêu $Z = CTQ * CBQ - COQ^2$
- Áp dụng giải thuật phân rã, bắt đầu tại điểm x (B1)

- $TA = \{C1, C3, C4\}$
- $BA = \{C2\}$
- $TQ = \{AP1, AP2, AP4\}$
- $BQ = \{\emptyset\}$
- $OQ = \{AP3\}$

$$\Rightarrow Z = 13 * 0 - 4 * 4 = -16$$

	C1	C2	C3	C4	Tổng tần xuất
AP1	1	0	0	1	3
AP2	0	0	0	1	7
AP3	0	1	1	0	4
AP4	1	0	1	0	3

	C4	C1	C3	C2
C4	10	3	0	0
C1	3	6	3	0
C3	0	3	7	4
C2	0	0	4	4

B3. Phân rã

- Hàm mục tiêu $Z = CTQ * CBQ - COQ^2$
- Áp dụng giải thuật phân rã, bắt đầu tại điểm x (B2)

- $TA = \{C1, C4\}$
- $BA = \{C2, C3\}$
- $TQ = \{AP1, AP2\}$
- $BQ = \{AP3\}$
- $OQ = \{AP4\}$

$$\Rightarrow Z = 10 * 4 - 3 * 3 = 31$$

	C1	C2	C3	C4	Tổng tần xuất
AP1	1	0	0	1	3
AP2	0	0	0	1	7
AP3	0	1	1	0	4
AP4	1	0	1	0	3

	C4	C1	C3	C2
C4	10	3	0	0
C1	3	6	3	0
C3	0	3	7	4
C2	0	0	4	4

B3. Phân rã

- Hàm mục tiêu $Z = CTQ * CBQ - COQ^2$
- Áp dụng giải thuật phân rã, bắt đầu tại điểm x (B3)

- $TA = \{C4\}$
- $BA = \{C1, C2, C3\}$
- $TQ = \{AP2\}$
- $BQ = \{AP3, AP4\}$
- $OQ = \{AP1\}$

$$\Rightarrow Z = 7*7 - 3*3 = 40$$

	C1	C2	C3	C4	Tổng tần xuất
AP1	1	0	0	1	3
AP2	0	0	0	1	7
AP3	0	1	1	0	4
AP4	1	0	1	0	3

	C4	C1	C3	C2
C4	10	3	0	0
C1	3	6	3	0
C3	0	3	7	4
C2	0	0	4	4

B3. Phân rã

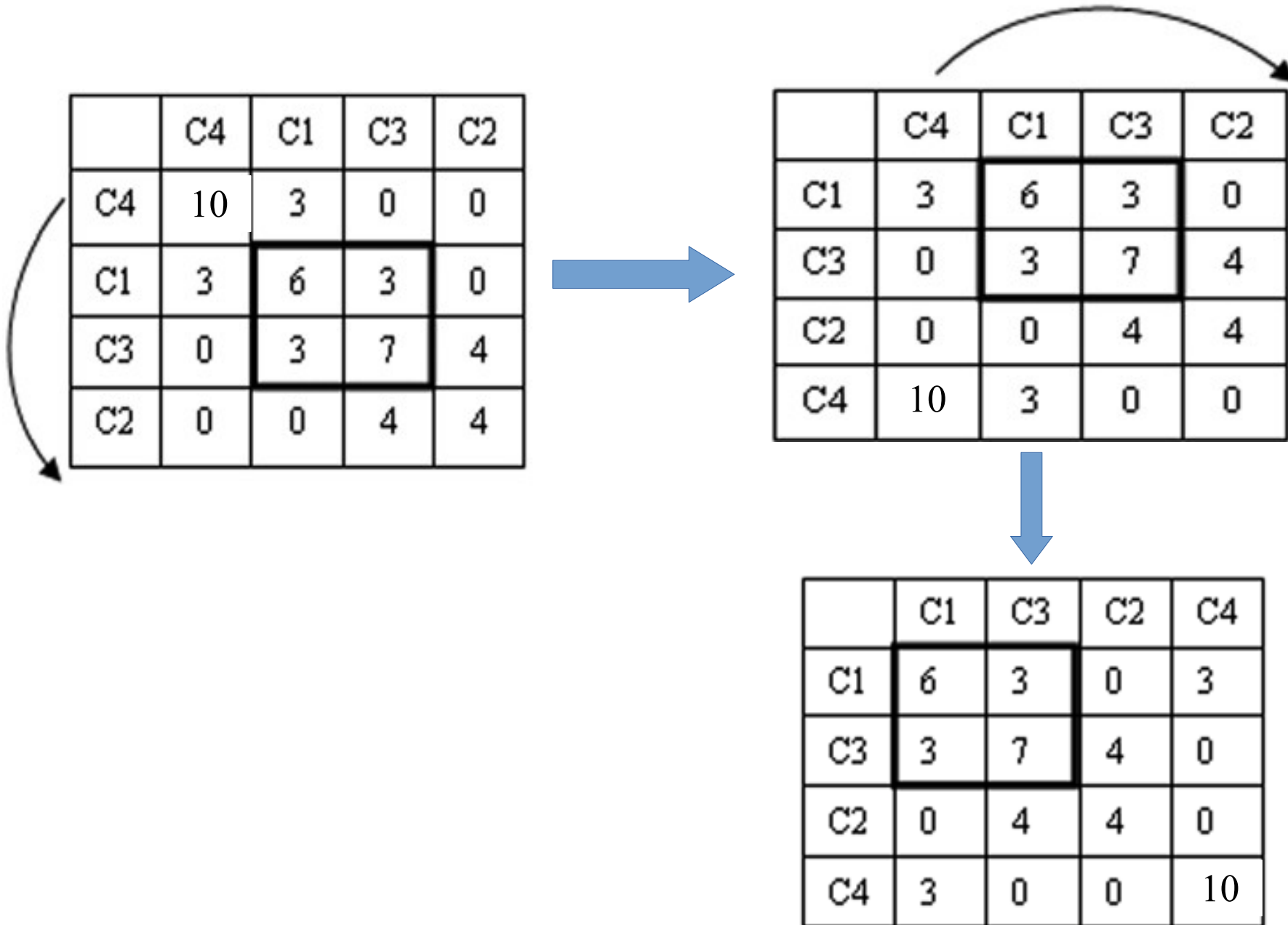
- **Nhận xét:**

- Cách tiếp cận này tạo ra các đoạn không chồng nhau bằng cách di chuyển theo đường chéo ma trận.
- Tuy nhiên giải thuật phân rã đã đề xuất có bất lợi là không thể xem một khối các cột bên trong như một đoạn cần phân rã. Ví dụ:
- Để khắc phục điều này tiếp cận phải đặt $\{C1, C3\}$ trong 1 đoạn và $\{C2, C4\}$ trong đoạn khác.

=> Thêm một phép dịch chuyển

	C4	C1	C3	C2
C4	10	3	0	0
C1	3	6	3	0
C3	0	3	7	4
C2	0	0	4	4

B3. Phân rã



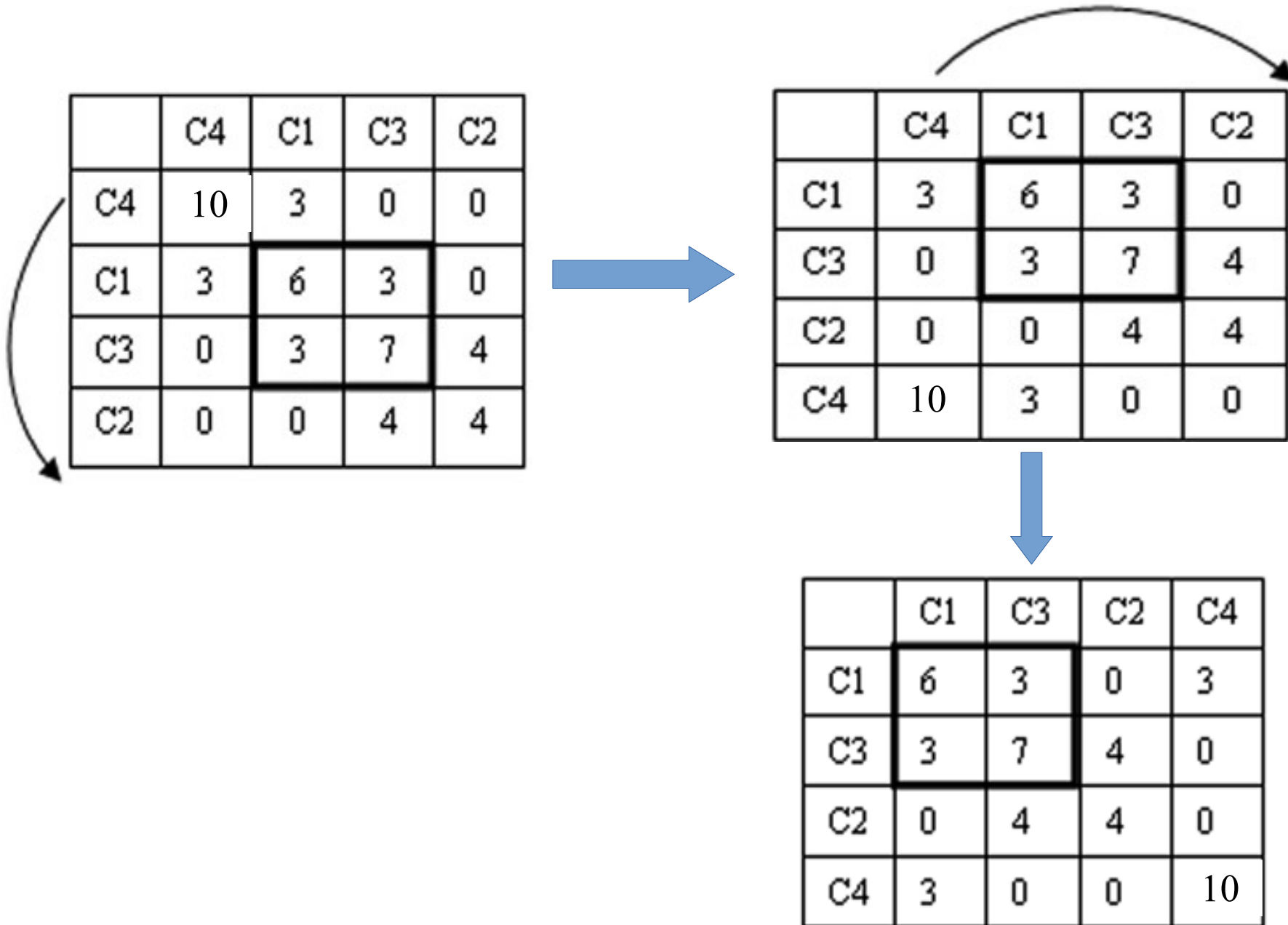
B3. Phân rã

- Hàm mục tiêu $Z = CTQ * CBQ - COQ^2$
- Áp dụng giải thuật phân rã (B4)
 - TA = {C1, C3}
 - BA = {C2, C4}
 - TQ = {AP4}
 - BQ = {AP2}
 - OQ = {AP1, AP3}
$$\Rightarrow Z = 3 * 7 - 7 * 7 = -28$$

	C1	C2	C3	C4	Tổng tần xuất
AP1	1	0	0	1	3
AP2	0	0	0	1	7
AP3	0	1	1	0	4
AP4	1	0	1	0	3

	C1	C3	C2	C4
C1	6	3	0	3
C3	3	7	4	0
C2	0	4	4	0
C4	3	0	0	10

B3. Phân rã



B3. Phân rã

- **Kết luận:**

Từ các kết quả trên, TEMP được phân thành 2 đoạn

$$F_{TEMP} = \{T_1, T_2\}$$

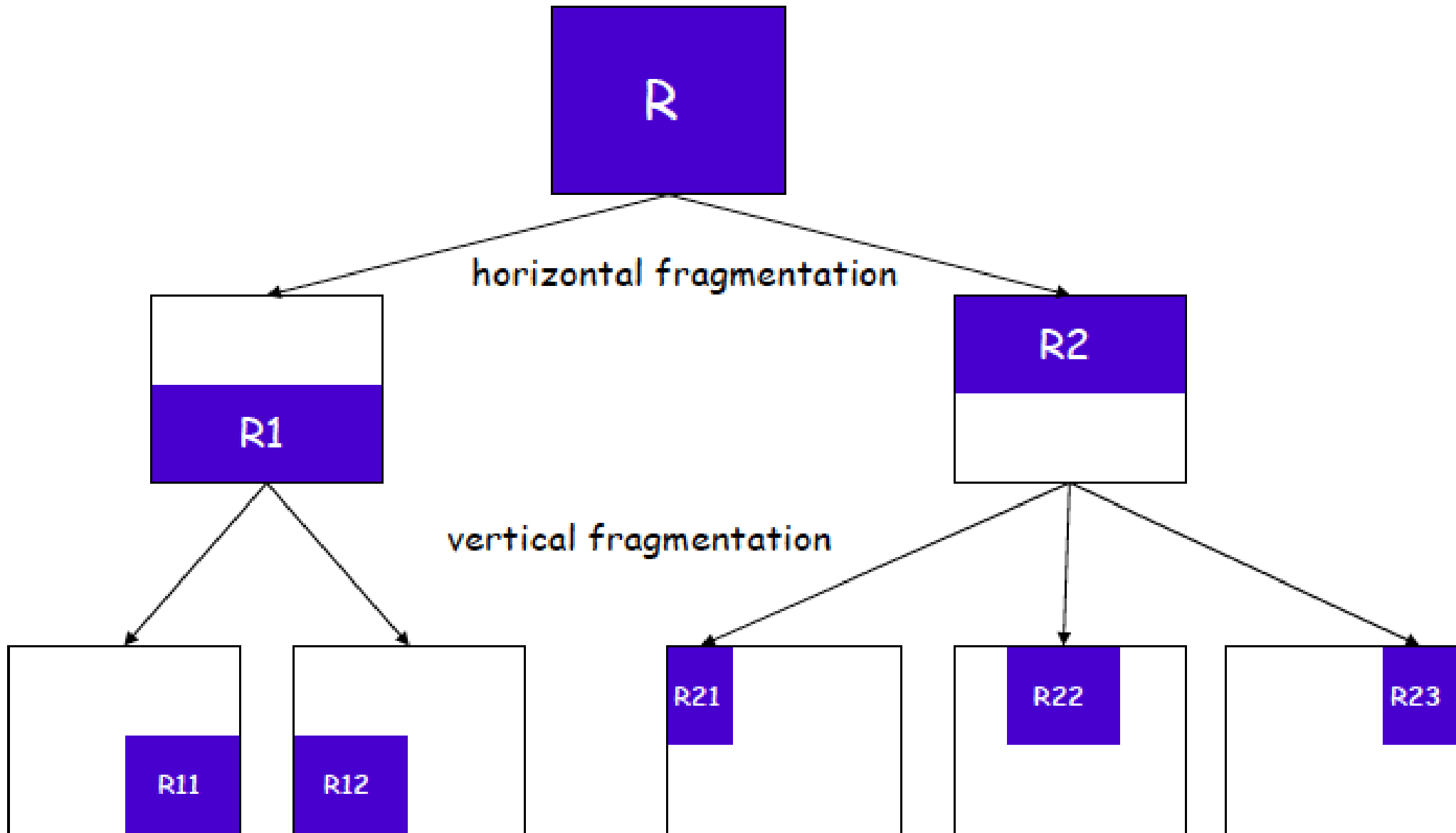
- $T_1 = \{C, C4\} = \pi_{C,C4}(TEMP)$

- $T_2 = \{C, C1, C2, C3\} = \pi_{C,C1,C2,C3}(TEMP)$

Phân đoạn lai

- Phân đoạn ngang hoặc phân đoạn dọc một lược đồ CSDL không đáp ứng đầy đủ các yêu cầu các ứng dụng.
 - Áp dụng phân đoạn dọc lên các đoạn ngang hoặc ngược lại gọi là phân đoạn lai
- Phân đoạn lai còn gọi là phân đoạn hỗn hợp (mixed fragmentation) hoặc phân đoạn lồng nhau (nested fragmentation)

Phân đoạn lại



Nội dung

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



Lược đồ cấp phát

- Việc đặt đoạn nào trên nút nào được quyết định dựa theo nguồn gốc của các câu truy vấn đã dùng để phân đoạn.
 - Đối với mỗi câu truy vấn:
 - Chúng ta biết một tập các nút mà có thể sinh ra câu truy vấn này
 - Chúng ta có một tập các đoạn liên quan đến câu truy vấn này
- => Mục đích là để đặt các đoạn trên các nút mà chúng được sử dụng nhiều nhất, và để giảm thiểu việc truyền dữ liệu giữa các nút.

Lược đồ cấp phát

- Để định nghĩa một lược đồ định vị/cấp phát, cần tìm:
 - **Ưu tiên 1:** Các câu truy vấn tìm kiếm được sinh ra ở đâu
 - **Ưu tiên 2:** Các câu cập nhật được thực hiện ở đâu ?
- Phân đoạn có thể được thực hiện với nhân bản (replication) hoặc không nhân bản.
 - Nhân bản làm tăng hiệu suất truy vấn và tính sẵn có của dữ liệu,
 - Nhưng gây tốn kém khi xem xét việc cập nhật (thêm, sửa, xóa) của các đoạn nhân bản.

Cấp phát - Ví dụ

- Xét quan hệ đầu bếp và các câu truy vấn:

DAUBEP(**ID**, Ho, Ten, NamKN)

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

R2: SELECT * FROM DAUBEP WHERE NamKN = 1;

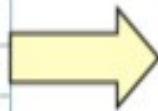
R3: SELECT ID, ho FROM DAUBEP WHERE NamKN =2 AND ten = 'Jean'

=> Tìm giải pháp cấp phát đơn giản các đoạn

Cấp phát - Ví dụ

- Giả sử rằng sau khi áp dụng các PP phân đoạn ngang và dọc, ta được các đoạn như sau:

ID	Ho	Ten	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



F11	F12	F13	F14																				
<table><tr><th>ID</th></tr><tr><td>13</td></tr></table>	ID	13	<table><tr><th>ID</th><th>Ho</th></tr><tr><td>13</td><td>DURDUR</td></tr></table>	ID	Ho	13	DURDUR	<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>13</td><td>2</td></tr></table>	ID	NamKN	13	2	<table><tr><th>ID</th><th>Ten</th></tr><tr><td>13</td><td>Jean</td></tr></table>	ID	Ten	13	Jean						
ID																							
13																							
ID	Ho																						
13	DURDUR																						
ID	NamKN																						
13	2																						
ID	Ten																						
13	Jean																						
F21	F22																						
<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>20</td><td>1</td></tr></table>	ID	NamKN	20	1	<table><tr><th>ID</th><th>Ho</th><th>Ten</th></tr><tr><td>20</td><td>DURALEX</td><td>Jean</td></tr></table>	ID	Ho	Ten	20	DURALEX	Jean												
ID	NamKN																						
20	1																						
ID	Ho	Ten																					
20	DURALEX	Jean																					
F31	F32																						
<table><tr><th>ID</th><th>Ho</th></tr><tr><td>26</td><td>DUGENOU</td></tr></table>	ID	Ho	26	DUGENOU	<table><tr><th>ID</th><th>Ten</th><th>NamKN</th></tr><tr><td>26</td><td>Jean</td><td>2</td></tr></table>	ID	Ten	NamKN	26	Jean	2												
ID	Ho																						
26	DUGENOU																						
ID	Ten	NamKN																					
26	Jean	2																					
F41																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>12</td><td>DUPONT</td><td>Jean</td><td>1</td></tr><tr><td>17</td><td>DUBOIS</td><td>Robert</td><td>1</td></tr><tr><td>22</td><td>DUBALAI</td><td>Aline</td><td>1</td></tr><tr><td>9</td><td>DURALUMIN</td><td>Roberte</td><td>1</td></tr></table>	ID	Ho	Ten	namKN	12	DUPONT	Jean	1	17	DUBOIS	Robert	1	22	DUBALAI	Aline	1	9	DURALUMIN	Roberte	1			
ID	Ho	Ten	namKN																				
12	DUPONT	Jean	1																				
17	DUBOIS	Robert	1																				
22	DUBALAI	Aline	1																				
9	DURALUMIN	Roberte	1																				
F51																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>34</td><td>DUPONT</td><td>Jeanne</td><td>2</td></tr><tr><td>11</td><td>DURAND</td><td>Aline</td><td>2</td></tr><tr><td>38</td><td>DURACUIRE</td><td>ROBERT</td><td>2</td></tr></table>	ID	Ho	Ten	namKN	34	DUPONT	Jeanne	2	11	DURAND	Aline	2	38	DURACUIRE	ROBERT	2							
ID	Ho	Ten	namKN																				
34	DUPONT	Jeanne	2																				
11	DURAND	Aline	2																				
38	DURACUIRE	ROBERT	2																				

Cấp phát - Ví dụ

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

=> F13, F21

F11	F12	F13	F14																				
<table><tr><th>ID</th></tr><tr><td>13</td></tr></table>	ID	13	<table><tr><th>ID</th><th>Ho</th></tr><tr><td>13</td><td>DURDUR</td></tr></table>	ID	Ho	13	DURDUR	<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>13</td><td>2</td></tr></table>	ID	NamKN	13	2	<table><tr><th>ID</th><th>Ten</th></tr><tr><td>13</td><td>Jean</td></tr></table>	ID	Ten	13	Jean						
ID																							
13																							
ID	Ho																						
13	DURDUR																						
ID	NamKN																						
13	2																						
ID	Ten																						
13	Jean																						
F21	F22																						
<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>20</td><td>1</td></tr></table>	ID	NamKN	20	1	<table><tr><th>ID</th><th>Ho</th><th>Ten</th></tr><tr><td>20</td><td>DURALEX</td><td>Jean</td></tr></table>	ID	Ho	Ten	20	DURALEX	Jean												
ID	NamKN																						
20	1																						
ID	Ho	Ten																					
20	DURALEX	Jean																					
F31	F32																						
<table><tr><th>ID</th><th>Ho</th></tr><tr><td>26</td><td>DUGENOU</td></tr></table>	ID	Ho	26	DUGENOU	<table><tr><th>ID</th><th>Ten</th><th>NamKN</th></tr><tr><td>26</td><td>Jean</td><td>2</td></tr></table>	ID	Ten	NamKN	26	Jean	2												
ID	Ho																						
26	DUGENOU																						
ID	Ten	NamKN																					
26	Jean	2																					
F41																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>12</td><td>DUPONT</td><td>Jean</td><td>1</td></tr><tr><td>17</td><td>DUBOIS</td><td>Robert</td><td>1</td></tr><tr><td>22</td><td>DUBALAI</td><td>Aline</td><td>1</td></tr><tr><td>9</td><td>DURALUMIN</td><td>Roberte</td><td>1</td></tr></table>	ID	Ho	Ten	namKN	12	DUPONT	Jean	1	17	DUBOIS	Robert	1	22	DUBALAI	Aline	1	9	DURALUMIN	Roberte	1			
ID	Ho	Ten	namKN																				
12	DUPONT	Jean	1																				
17	DUBOIS	Robert	1																				
22	DUBALAI	Aline	1																				
9	DURALUMIN	Roberte	1																				
F51																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>34</td><td>DUPONT</td><td>Jeanne</td><td>2</td></tr><tr><td>11</td><td>DURAND</td><td>Aline</td><td>2</td></tr><tr><td>38</td><td>DURACUIRE</td><td>ROBERT</td><td>2</td></tr></table>	ID	Ho	Ten	namKN	34	DUPONT	Jeanne	2	11	DURAND	Aline	2	38	DURACUIRE	ROBERT	2							
ID	Ho	Ten	namKN																				
34	DUPONT	Jeanne	2																				
11	DURAND	Aline	2																				
38	DURACUIRE	ROBERT	2																				

Cấp phát - Ví dụ

R2: SELECT * FROM
DAUBEP WHERE
NamKN = 1;

=> F21, F22, F41

F11	F12	F13	F14																				
<table><tr><th>ID</th></tr><tr><td>13</td></tr></table>	ID	13	<table><tr><th>ID</th><th>Ho</th></tr><tr><td>13</td><td>DURDUR</td></tr></table>	ID	Ho	13	DURDUR	<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>13</td><td>2</td></tr></table>	ID	NamKN	13	2	<table><tr><th>ID</th><th>Ten</th></tr><tr><td>13</td><td>Jean</td></tr></table>	ID	Ten	13	Jean						
ID																							
13																							
ID	Ho																						
13	DURDUR																						
ID	NamKN																						
13	2																						
ID	Ten																						
13	Jean																						
F21	F22																						
<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>20</td><td>1</td></tr></table>	ID	NamKN	20	1	<table><tr><th>ID</th><th>Ho</th><th>Ten</th></tr><tr><td>20</td><td>DURALEX</td><td>Jean</td></tr></table>			ID	Ho	Ten	20	DURALEX	Jean										
ID	NamKN																						
20	1																						
ID	Ho	Ten																					
20	DURALEX	Jean																					
F31	F32																						
<table><tr><th>ID</th><th>Ho</th></tr><tr><td>26</td><td>DUGENOU</td></tr></table>	ID	Ho	26	DUGENOU	<table><tr><th>ID</th><th>Ten</th><th>NamKN</th></tr><tr><td>26</td><td>Jean</td><td>2</td></tr></table>	ID	Ten	NamKN	26	Jean	2												
ID	Ho																						
26	DUGENOU																						
ID	Ten	NamKN																					
26	Jean	2																					
F41																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>12</td><td>DUPONT</td><td>Jean</td><td>1</td></tr><tr><td>17</td><td>DUBOIS</td><td>Robert</td><td>1</td></tr><tr><td>22</td><td>DUBALAI</td><td>Aline</td><td>1</td></tr><tr><td>9</td><td>DURALUMIN</td><td>Roberte</td><td>1</td></tr></table>				ID	Ho	Ten	namKN	12	DUPONT	Jean	1	17	DUBOIS	Robert	1	22	DUBALAI	Aline	1	9	DURALUMIN	Roberte	1
ID	Ho	Ten	namKN																				
12	DUPONT	Jean	1																				
17	DUBOIS	Robert	1																				
22	DUBALAI	Aline	1																				
9	DURALUMIN	Roberte	1																				
F51																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>34</td><td>DUPONT</td><td>Jeanne</td><td>2</td></tr><tr><td>11</td><td>DURAND</td><td>Aline</td><td>2</td></tr><tr><td>38</td><td>DURACUIRE</td><td>ROBERT</td><td>2</td></tr></table>				ID	Ho	Ten	namKN	34	DUPONT	Jeanne	2	11	DURAND	Aline	2	38	DURACUIRE	ROBERT	2				
ID	Ho	Ten	namKN																				
34	DUPONT	Jeanne	2																				
11	DURAND	Aline	2																				
38	DURACUIRE	ROBERT	2																				

Cấp phát - Ví dụ

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

=> F12, F31

F11	F12	F13	F14																				
<table><tr><th>ID</th></tr><tr><td>13</td></tr></table>	ID	13	<table><tr><th>ID</th><th>Ho</th></tr><tr><td>13</td><td>DURDUR</td></tr></table>	ID	Ho	13	DURDUR	<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>13</td><td>2</td></tr></table>	ID	NamKN	13	2	<table><tr><th>ID</th><th>Ten</th></tr><tr><td>13</td><td>Jean</td></tr></table>	ID	Ten	13	Jean						
ID																							
13																							
ID	Ho																						
13	DURDUR																						
ID	NamKN																						
13	2																						
ID	Ten																						
13	Jean																						
F21	F22																						
<table><tr><th>ID</th><th>NamKN</th></tr><tr><td>20</td><td>1</td></tr></table>	ID	NamKN	20	1	<table><tr><th>ID</th><th>Ho</th><th>Ten</th></tr><tr><td>20</td><td>DURALEX</td><td>Jean</td></tr></table>	ID	Ho	Ten	20	DURALEX	Jean												
ID	NamKN																						
20	1																						
ID	Ho	Ten																					
20	DURALEX	Jean																					
F31	F32																						
<table><tr><th>ID</th><th>Ho</th></tr><tr><td>26</td><td>DUGENOU</td></tr></table>	ID	Ho	26	DUGENOU	<table><tr><th>ID</th><th>Ten</th><th>NamKN</th></tr><tr><td>26</td><td>Jean</td><td>2</td></tr></table>	ID	Ten	NamKN	26	Jean	2												
ID	Ho																						
26	DUGENOU																						
ID	Ten	NamKN																					
26	Jean	2																					
F41																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>12</td><td>DUPONT</td><td>Jean</td><td>1</td></tr><tr><td>17</td><td>DUBOIS</td><td>Robert</td><td>1</td></tr><tr><td>22</td><td>DUBALAI</td><td>Aline</td><td>1</td></tr><tr><td>9</td><td>DURALUMIN</td><td>Roberte</td><td>1</td></tr></table>	ID	Ho	Ten	namKN	12	DUPONT	Jean	1	17	DUBOIS	Robert	1	22	DUBALAI	Aline	1	9	DURALUMIN	Roberte	1			
ID	Ho	Ten	namKN																				
12	DUPONT	Jean	1																				
17	DUBOIS	Robert	1																				
22	DUBALAI	Aline	1																				
9	DURALUMIN	Roberte	1																				
F51																							
<table><tr><th>ID</th><th>Ho</th><th>Ten</th><th>namKN</th></tr><tr><td>34</td><td>DUPONT</td><td>Jeanne</td><td>2</td></tr><tr><td>11</td><td>DURAND</td><td>Aline</td><td>2</td></tr><tr><td>38</td><td>DURACUIRE</td><td>ROBERT</td><td>2</td></tr></table>	ID	Ho	Ten	namKN	34	DUPONT	Jeanne	2	11	DURAND	Aline	2	38	DURACUIRE	ROBERT	2							
ID	Ho	Ten	namKN																				
34	DUPONT	Jeanne	2																				
11	DURAND	Aline	2																				
38	DURACUIRE	ROBERT	2																				

Cấp phát - Ví dụ

- Giả sử rằng có 2 nút A và B
 - R1 sinh ra bởi A hoặc B
 - R2 sinh ra chỉ bởi A
 - R3 sinh ra chỉ bởi B
 - Cho cả 3 câu truy vấn, các đoạn sau có liên quan
 - $R1 \rightarrow$ đoạn F13, F21
 - $R2 \rightarrow$ F21, F22, F41
 - $R3 \rightarrow$ F12, F31
 - Đối với các đoạn còn lại:
 - Đoạn F21 có thể liên quan R1 hoặc R2
 - Đoạn F11 liên quan tất cả các câu truy vấn
 - Đoạn F14, F32, F51 không liên quan câu truy vấn nào
- => lựa chọn các đoạn để các site cân bằng

Cấp phát - Ví dụ

- Giả sử rằng có 2 nút A và B
 - R1 sinh ra bởi A hoặc B, R2 sinh ra chỉ bởi A và R3 sinh ra chỉ bởi B
 - Cho cả 3 câu truy vấn, các đoạn sau có liên quan
 - R1 → đoạn F13, F21
 - R2 → F21, F22, F41
 - R3 → F12, F31
 - Đoạn F21 có thể liên quan R1 hoặc R2
 - Đoạn F11 liên quan tất cả các câu truy vấn
 - Đoạn F14, F32, F51 không liên quan câu truy vấn nào
 - Trên các site:
 - Site A: F13, F21, F22, F41, F11, F51
 - Site B: F12, F31, F14, F32
- => Các đoạn có thể kết hợp lại
- Site A: F11, F13, F2, F4, F5
 - Site B: F12, F14, F3

=> Trong trường hợp phương pháp cấp phát đơn giản không thoả mãn, các kỹ thuật mạnh hơn (phức tạp hơn) có thể được sử dụng.

Cấp phát (Allocation)

- Cho:
 - $F = \{F_1, F_2, \dots, F_n\}$ là một tập các đoạn
 - $S = \{S_1, S_2, \dots, S_m\}$ là tập các nút trong một hệ thống phân tán
 - $Q = \{q_1, q_2, \dots, q_q\}$ là một tập các ứng dụng đang chạy trên S .

=> Vấn đề cấp phát liên quan đến việc tìm kiếm sự phân bố "tối ưu" của F trên S .

- Sự tối ưu có thể được định nghĩa đối với hai vấn đề:
 - Chi phí tối thiểu
 - Hiệu suất

Chi phí tối thiểu

- Hàm chi phí bao gồm:
 - Chi phí lưu trữ mỗi F_i tại một nút S_j ,
 - Chi phí truy vấn F_i tại nút S_j ,
 - Chi phí cập nhật F_i tại tất cả các nút nơi nó được lưu giữ, và
 - Chi phí truyền thông dữ liệu.
- Vấn đề cấp phát, vì vậy thử tìm một lược đồ cấp phát để giảm thiểu hàm chi phí kết hợp.

Hiệu suất

- Chiến lược phân bổ được thiết kế để duy trì hiệu suất
- Hai vấn đề phổ biến cần xem xét:
 - Giảm thiểu thời gian đáp ứng và
 - Tối đa hóa *thông lượng** (throughput) hệ thống tại mỗi nút.

** là lượng thông tin hữu ích được truyền đi trên mạng trong một đơn vị thời gian*

Các yêu cầu về thông tin

- Ở giai đoạn cấp phát chúng ta cần các dữ liệu định lượng về
 - cơ sở dữ liệu,
 - các ứng dụng chạy trên CSDL,
 - mạng truyền thông,
 - khả năng xử lý, và
 - các giới hạn lưu trữ của mỗi nút trên mạng.

Thông tin CSDL

- Sự chọn lọc các đoạn (selectivity)
 - Sự chọn lọc một đoạn F_j đối với truy vấn q_i là số bộ của F_j cần truy cập để xử lý q_i . Giá trị này sẽ được ý hiệu là $sel_i(F_j)$.
- Kích thước của một đoạn
 - Kích thước của đoạn F_j cho bởi công thức

$$size(F_j) = card(F_j) * length(F_j)$$

Trong đó: $length(F_j)$ là kích thước (byte) của mỗi bộ trong đoạn F_j

Thông tin ứng dụng

- RR_{ij} : số truy cập **đọc** của một truy vấn q_i đến đoạn F_j
- UR_{ij} : số truy cập **cập nhật** của truy vấn q_i đến đoạn F_j
- UM : ma trận với các thành phần u_{ij} cho biết câu truy vấn nào **cập nhật** các đoạn nào,
- RM : một ma trận với các thành phần r_{ij} cho biết câu truy vấn nào truy xuất **đọc** các đoạn nào

$$u_{ij} = \begin{cases} 1, & \text{nếu } q_i \text{ cập nhật đoạn } F_j \\ 0, & \text{ngược lại} \end{cases}$$

$$r_{ij} = \begin{cases} 1, & \text{nếu } q_i \text{ đọc đoạn } F_j \\ 0, & \text{ngược lại} \end{cases}$$

- Vector O với các giá trị $o(i)$ chỉ ra nút gốc (nguồn gốc) của câu truy vấn q_i

Thông tin nút (site)

- Với mỗi nút, chúng ta cần biết:
 - Khả năng lưu trữ
 - khả năng xử lý
- Các giá trị này có thể được tính bằng các hàm phức tạp hoặc ước tính đơn giản
 - USC_k : chi phí đơn vị cho việc lưu trữ dữ liệu tại một nút S_k
 - LPC_k : chi phí xử lý một đơn vị dữ liệu tại nút S_k

Thông tin mạng

- Chi phí giao tiếp (frame) giữa 2 nút,
 - g_{ij} biểu thị sự chi phí giao tiếp cho mỗi frame giữa các nút S_i và S_j .
- Kích thước của frame: $fsize$

Mô hình cấp phát

- Mô hình cấp phát (allocation model) nhằm:
 - Giảm thiểu tổng chi phí xử lý và lưu trữ
 - Trong khi cố gắng thoả mãn các giới hạn về thời gian đáp ứng.
- Mô hình có dạng chung:

$\min(\text{Total Cost})$

tuỳ thuộc vào:

- Ràng buộc về thời gian đáp ứng
- Ràng buộc về lưu trữ
- Ràng buộc về xử lý

Mô hình cấp phát

- Hàm tổng chi phí có 2 thành phần: **lưu trữ** và **xử lý truy vấn**:

$$TOC = \sum_{S_k \in S} \sum_{F_j \in F} STC_{jk} + \sum_{q_i \in Q} QPC_i$$

- Chi phí lưu trữ của đoạn F_j tại nút S_k :**

$$STC_{jk} = USC_k * size(F_j) * x_{ij}$$

Trong đó: USC_k là chi phí đơn vị cho việc lưu trữ dữ liệu tại một nút S_k

$$x_{ij} = \begin{cases} 1, & \text{nếu } F_j \text{ được lưu trữ tại } S_k \\ 0, & \text{ngược lại} \end{cases}$$

- Chi phí xử lý truy vấn** cho câu truy vấn q_i bao gồm hai thành phần: **chi phí xử lý** (processing Cost - PC) và **chi phí truyền tải** (transmission cost - TC)

$$QPC_i = PC_i + TC_i$$

Mô hình cấp phát

- Chi phí xử lý là tổng của 3 thành phần: chi phí truy cập (AC - access cost), chi phí ràng buộc toàn vẹn (IE - integrity constraint), chi phí điều khiển cạnh tranh (CC - concurency control)

$$PC_i = AC_i + IE_i + CC_i$$

Trong đó:

$$AC_i = \sum_{s_k \in S} \sum_{F_j \in F} (UR_{ij} + RR_{ij}) * x_{ij} * LPC_k$$

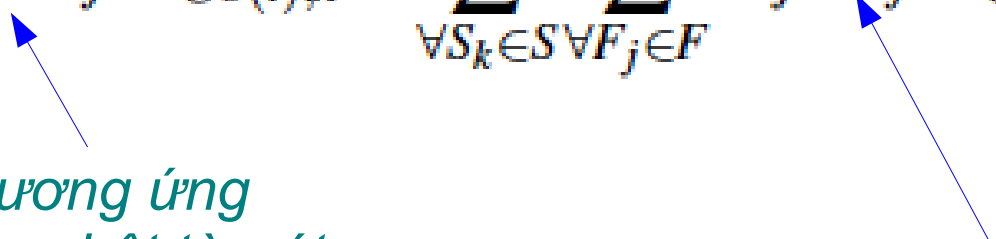
- IE và CC có thể được tính tương tự

Mô hình cấp phát

- Chi phí truyền tải bao gồm hai thành phần:
 - Chi phí xử lý cập nhật (TCU) và
 - Chi phí xử lý truy xuất (TCR)

$$TC_i = TCU_i + TCR_i$$

Trong đó:

$$TCU_i = \sum_{\forall S_k \in S} \sum_{\forall F_j \in F} u_{ij} * x_{jk} * g_{o(i),k} + \sum_{\forall S_k \in S} \sum_{\forall F_j \in F} u_{ij} * x_{jk} * g_{k,o(i)}$$


- Thành phần đầu tiên tương ứng việc gửi thông điệp cập nhật từ nút có nguồn gốc $o(i)$ của q_i cho tất cả các bản sao đoạn cần được cập nhật.
- Thành phần thứ hai dành cho xác nhận.

Mô hình cấp phát

- Chi phí truyền tải bao gồm hai thành phần (tt):

$$TCR_i = \sum_{\forall F_j \in F} \min_{S_k \in S} (r_{ij} * x_{jk} * g_{o(i),k} + r_{ij} * x_{jk} * \frac{sel_i(F_j) * length(F_j)}{fsize} * g_{k,o(i)})$$

Thành phần đầu tiên trong đại diện cho chi phí gửi yêu cầu tìm kiếm tới các site có bản sao của các đoạn cần truy cập.

Thành phần thứ hai tương ứng cho việc gửi các kết quả từ các site này đến các site gốc.

Phương trình trên chỉ ra rằng trong số tất cả các site với các bản sao của cùng một đoạn, chỉ các nút mà mang lại tổng chi phí truyền tải nhỏ nhất được lựa chọn cho thực hiện hoạt động.

Mô hình cấp phát

- Các ràng buộc

- Ràng buộc thời gian đáp ứng cho một truy vấn q_i

Thời gian thực hiện $q_i \leq$ thời gian đáp ứng cho phép tối đa cho q_i

- Ràng buộc lưu trữ cho một nút S_k

$$\sum_{F_j \in F} (\text{yêu cầu lưu trữ của } F_j \text{ tại } S_k) \leq \text{khả năng lưu trữ của } S_k$$

- Ràng buộc xử lý cho nút S_k

$$\sum_{q_i \in Q} (\text{Tải xử lý của } q_i \text{ tại } S_k) \leq \text{khả năng xử lý của } S_k$$

Mô hình cấp phát

- Phương pháp giải pháp
 - Sự phức tạp của vấn đề cấp phát đoạn là NP-complete
=> Sử dụng các phương pháp heuristic khác nhau để giảm sự phức tạp
 - *Vấn đề cấp phát* tương tự các vấn đề trong các lĩnh vực khác :
 - Knapsack problem solution [Ceri et al., 1982] ,
 - Branch and-bound techniques [Fisher and Hochbaum, 1980] ,
 - network flow algorithms [Chang and Liu, 1982] .
 - Các giải pháp của các lĩnh vực khác có thể được sử dụng lại

Tóm tắt

- Thiết kế CSDL phân tán là tiếp cận trên xuống
- Xây dựng lược đồ toàn cục → quyết định việc phân tán dữ liệu như thế nào.
- Phân đoạn ngang sử dụng phép chọn
 - Xây dựng tập các vị từ minterm từ các vị từ đơn giản
- Phân đoạn dọc sử dụng phép chiếu
 - Xây dựng ma trận $AA \rightarrow CA \rightarrow$ phân rã.
- Phân đoạn lai kết hợp cả hai tiếp cận trên
- Kiểm tra tính đúng đắn của phân đoạn (3 luật)
- Cấp phát đoạn.