Chương 6

Truy vấn trong CSDL phân tán

XỬ LÝ TRUY VẤN TRONG CSDL PHÂN TÁN

+ Vai trò của thể xử lý vấn tin phân tán là ánh xạ câu truy vấn cấp cao trên một CSDL phân tán vào một chuỗi các thao tác của đại số quan hệ trên các mảnh, thể hiện các bước:
  - Câu truy vấn phải được phân rã thành một chuỗi các phép toán quan hệ được gọi là vấn tin đại số.
  - Thứ hai, dữ liệu cần truy xuất phải được cục bộ hóa để các thao tác trên các quan hệ được chuyển thành các thao tác trên dữ liệu cục bộ (các mảnh).
  - Cuối cùng câu truy vấn đại số trên các mảnh phải được mở rộng để bao gồm các thao tác truyền thông và được tối ưu hóa để hàm chi phí là thấp nhất.
+ Hàm chi phí muốn nói đến các tính toán như thao tác xuất nhập đĩa, tài nguyên CPU, và mạng truyền thông.

1. Bài toán xử lý truy vấn

Có hai phương pháp tối ưu hóa cơ bản được sử dụng trong các bộ xử lý truy vấn:

- Phương pháp biến đổi đại số
- Chiến lược ước lượng chi phí.

- Phương pháp biến đổi đại số đơn giản hóa các câu truy vấn nhờ các phép biến đổi đại số nhằm hạ thấp chi phí trả lời câu vấn tin, độc lập với dữ liệu thực và cấu trúc vật lý của dữ liệu.
- Nhiệm vụ chính của thể xử lý truy vấn quan hệ là biến đổi câu vấn tin cấp cao thành một câu truy vấn tương đương ở cấp thấp hơn được diễn đạt bằng đại số quan hệ. Việc biến đổi này phải đạt được cả tính đúng đắn lẫn tính hiệu quả.

Ví dụ:

Xét một tập con của lược đồ CSDL đã được cho

NV(MNV, TênNV, Chức vụ)

PC (MNV, MDA, Nhiệm vụ, Thời gian)

Và một câu truy vấn đơn giản sau:

"Liệt kê tên của các nhân viên hiện đang quản lý một dự án"

Biểu thức truy vấn bằng phép tính quan hệ theo cú pháp của SQL là:

```
SELECT TênNV
FROM        NV, PC
WHERE NV.MNV=PC.MNV
AND     Nhiệmvụ="Quảnlý"
```

Thí dụ:

Hai biểu thức tương đương trong đại số quan hệ do biến đổi chính xác từ câu vấn tin trên là:

$$\pi_{TênNV}(\sigma_{Nhiệmvụ=”Quảnlý” \wedge NV.MNV=PC.MNV} (NV \times PC))$$

và

$$\pi_{TênNV}(NV |><|_{MNV}(\sigma_{Nhiệmvụ=” \ Quảnlý”} (PC)))$$

- Hiển nhiên là trong câu vấn tin thứ hai, chúng ta tránh sử dụng tích Descartes, vì thế tiêu dùng ít tài nguyên máy tính hơn câu vấn tin thứ nhất và vì vậy nên được giữ lại.
- Trong các hệ phân tán, đại số quan hệ không đủ để diễn tả các chiến lược thực thi. Nó phải được cung cấp thêm các phép toán trao đổi dữ liệu giữa các vị trí. Bên cạnh việc chọn thứ tự cho các phép toán đại số quan hệ, thể xử lý vấn tin phân tán cũng phải chọn các vị trí tốt nhất để xử lý dữ liệu, và có thể cả cách biến đổi dữ liệu.

Ví dụ 2:

Thí dụ này minh họa tầm quan trọng của việc chọn lựa vị trí và cách truyền dữ liệu của một câu vấn tin đại số. Chúng ta xét câu vấn tin của thí dụ trên:

$$\pi_{\text{TênNV}}(\text{NV}|><|_{\text{MNV}}(\sigma_{\text{Nhiệmvụ}=\text{"Quảnlý"}}(\text{PC})))$$

chúng ta giả sử rằng các quan hệ NV và PC được phân mảnh ngang như sau:

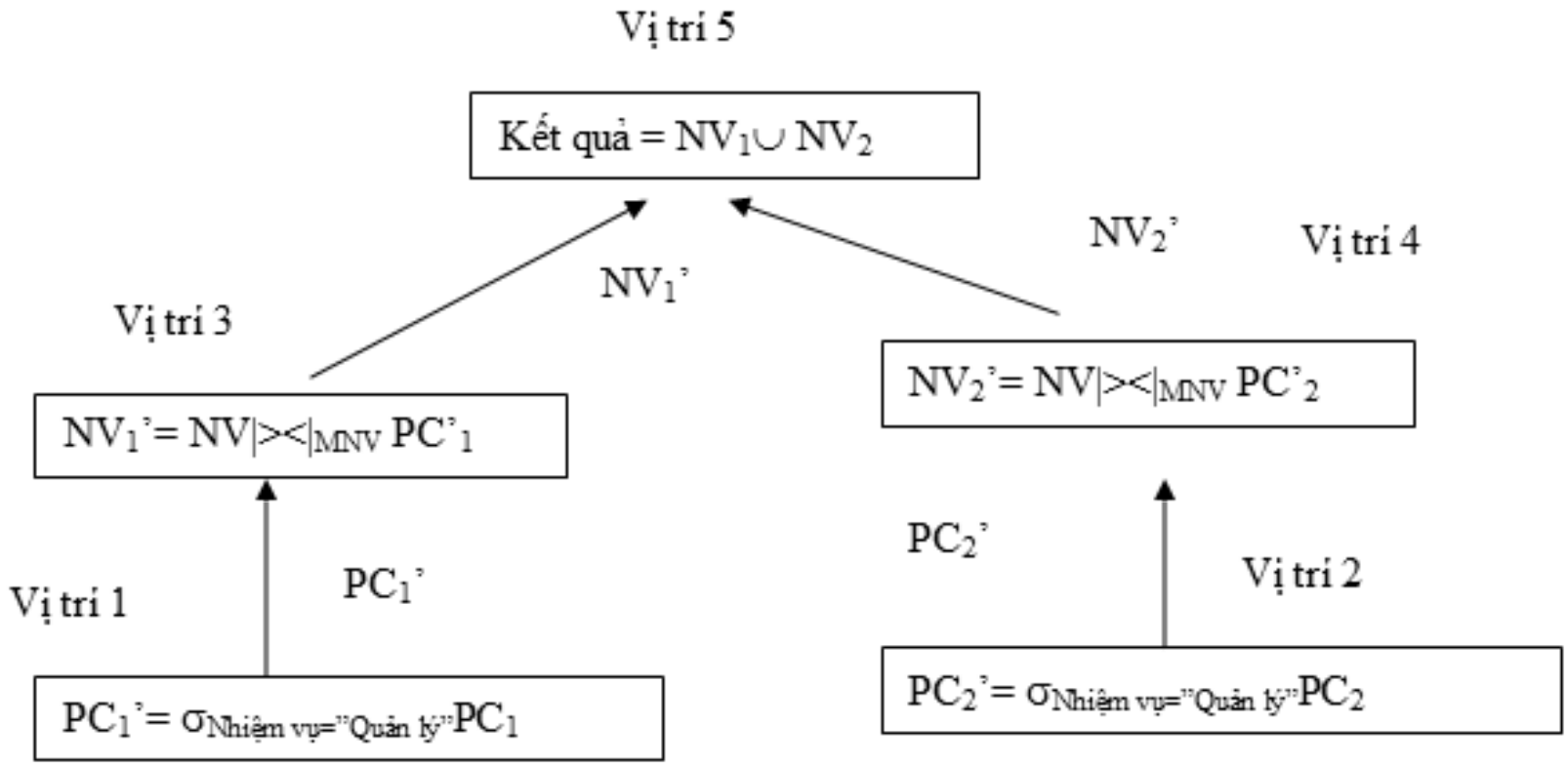
$$\text{NV1}=\sigma_{\text{MNV} \le \text{"E3"}}(\text{NV})$$

$$\text{NV2}=\sigma_{\text{MNV} > \text{"E3"}}(\text{NV})$$

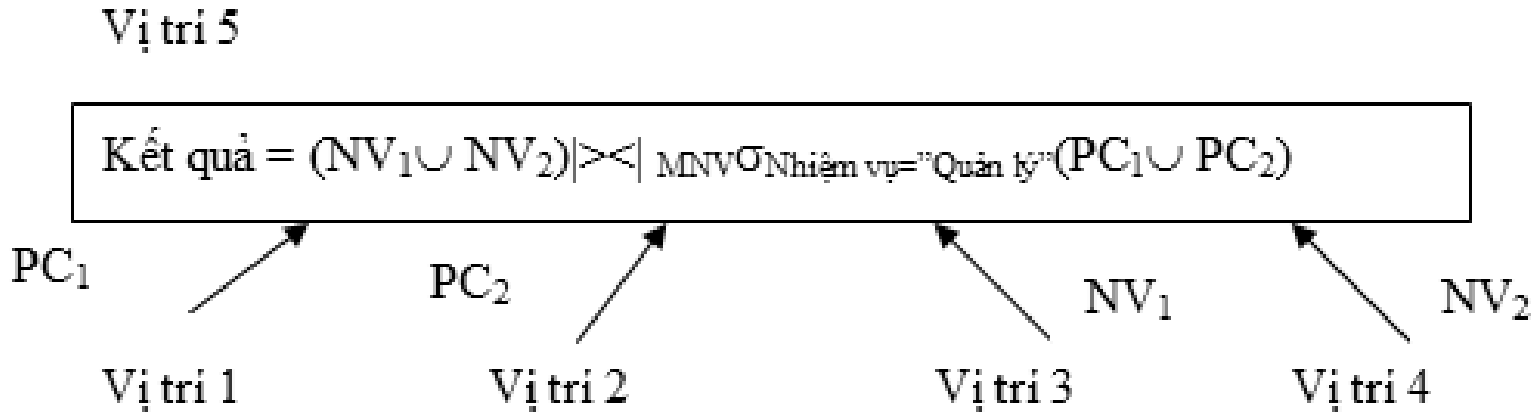
$$\text{PC1}=\sigma_{\text{MNV} \le \text{"E3"}}(\text{PC})$$

$$\text{PC2}=\sigma_{\text{MNV} \ge \text{"E3"}}(\text{PC})$$

Các mảnh PC1, PC2, NV1, NV2 theo thứ tự được lưu tại các vị trí 1, 2, 3 và 4 và kết quả được lưu tại vị trí 5



Hình 4.1a) Chiến lược a



Hình 4.1b) Chiến lược b

Mũi tên từ vị trí i đến vị trí j có nhãn R chỉ ra rằng quan hệ R được chuyển từ vị trí i đến vị trí j. Chiến lược a) sử dụng sự kiện là các quan hệ được phân mảnh theo cùng một cách để thực hiện song song các phép toán chọn và nối. Chiến lược b tập trung tất cả các dữ liệu tại vị trí lưu kết quả trước khi xử lý câu truy vấn.

Giả sử:

- Thao tác truy xuất một bộ (tuple access) được ký hiệu là tupacc, là một đơn vị và thao tác truyền một bộ (tuple transfer) tuptrans là 10 đơn vị.

- Các quan hệ NV và PC tương ứng có 400 và 1000 bộ, và có 20 giám đốc dự án thống nhất cho các vị trí.

- Các quan hệ PC và NV được gom cục bộ tương ứng theo các thuộc tính Nhiệmvụ và MNV. Vì vậy có thể truy xuất trực tiếp đến các bộ của PC dựa trên giá trị của thuộc tính Nhiệmvụ (tương ứng là MNV cho NV)

* Tổng chi phí của chiến lược a có thể được tính như sau:

1.Tạo ra PC' bằng cách chọn trên PC cần (10+10)* tupacc = 20

2. Truyền PC' đến vị trí của NV cần (10+10)*tuptrans = 200

3. Tạo NV' bằng cách nối PC' và NV' cần (10+10)*tupacc*2 = 40

4. Truyền NV' đến vị trí nhận kết quả cần (10+10)*tuptrans = 200

Tổng chi phí 460

* Tổng chi phí cho chiến lược b có thể được tính như sau:

1. Truyền NV đến vị trí 5 cần 400*tuptrans = 4.000

2. truyền PC đến vị trí 5 cần 1000*tuptrans =10.000

3. Tạo ra PC' bằng cách chọn trên PC cần 1000*tupacc = 1.000

4. Nối NVvà PC cần 400*20*tupacc = 8.000

Tổng chi phí là 23.000

Chỉ số đánh giá tiêu dùng tài nguyên

- Chỉ số đánh giá tiêu dùng tài nguyên là tổng chi phí (total cost) phải trả khi xử lý truy vấn.
- Tổng chi phí là tổng thời gian cần để xử lý các phép toán vấn tin tại các vị trí khác nhau và truyền dữ liệu giữa các vị trí.
- Một công cụ khác là thời gian đáp ứng của câu vấn tin, là thời gian cần thiết để chạy câu vấn tin.
- Trong môi trường CSDL phân tán, tổng chi phí cần phải giảm thiểu là chi phí CPU, chi phí xuất nhập và chi phí truyền.

Độ phức tạp của các phép toán quan hệ

+ Các phép toán chọn, Chiếu (Không loại bỏ trùng lặp) có độ phức tạp là O(n);

+ Các phép chiếu (Có loại bỏ trùng lặp), trùng lặp, nối, nối nửa, chia có độ phức tạp là O(n*logn);

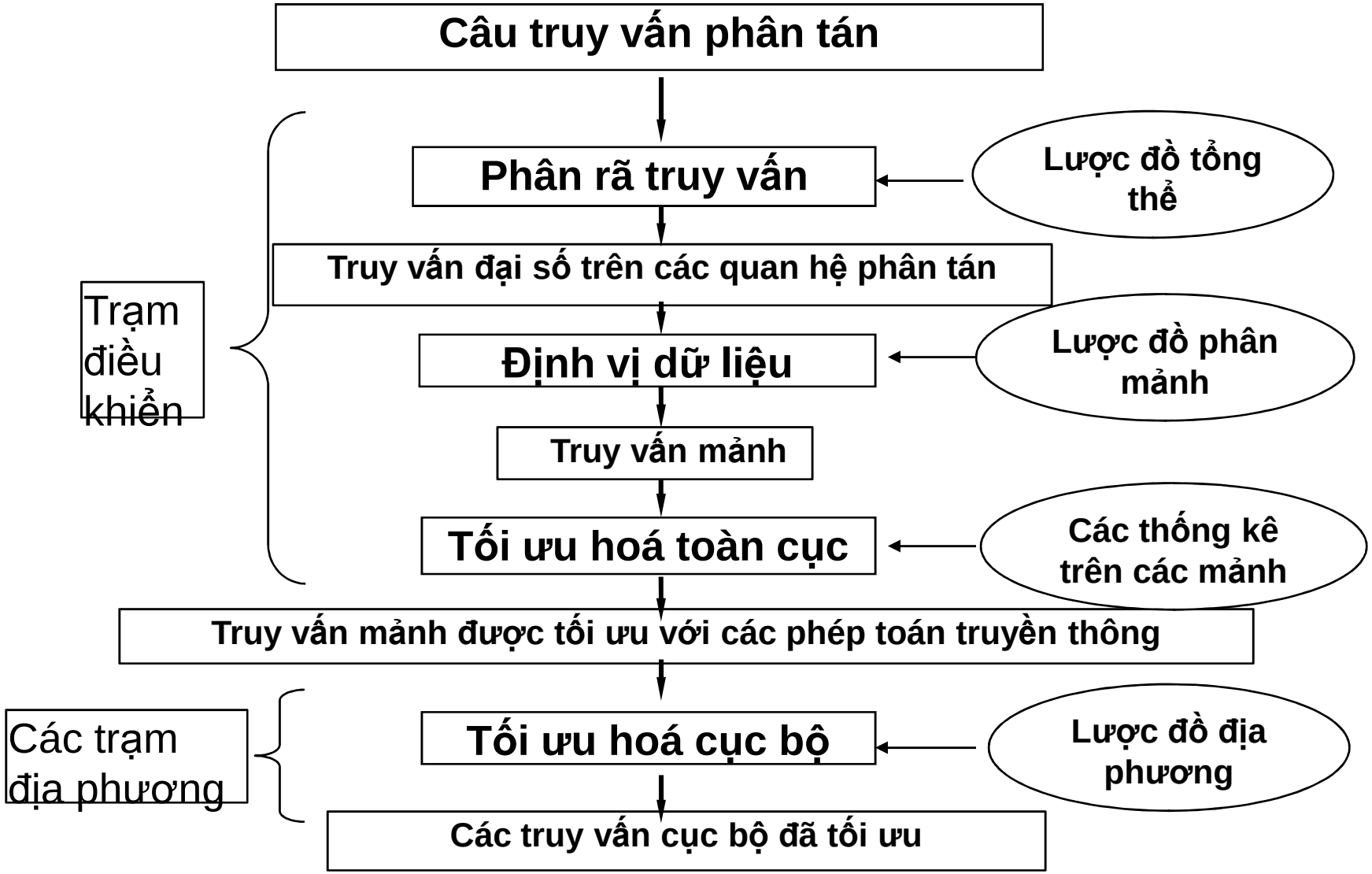
+ Tích Descartes có độ phức tạp là $O(n^2)$

(N biểu thị lực lượng của quan hệ nếu các bộ thu được độc lập với nhau)

Đánh giá:

- Các thao tác có tính chọn lựa làm giảm đi lực lượng cần phải thực hiện trước tiên.
- Các phép toán cần phải được sắp xếp để tránh thực hiện tích Descartes hoặc để lại thực hiện sau.

Xử lý truy vấn trong môi trường phân tán



Sơ đồ phân lớp chung cho xử lý truy vấn phân tán

Xử lý truy vấn trong môi trường phân tán

Phân rã truy vấn

Giai đoạn này chia làm bốn bước: chuẩn hoá, phân tích, loại bỏ dư thừa và viết lại.

1. Chuẩn hoá

Mục đích: chuyển đổi truy vấn thành một dạng chuẩn để thuận lợi cho các xử lý tiếp theo.

Với SQL, có hai dạng chuẩn cho các vị từ trong mệnh đề WHERE là:

Dạng chuẩn hội là hội ($\wedge$) của những phép toán tuyển ($\vee$):

$(p_{11} \vee p_{12} \vee \ldots \vee p_{1n}) \wedge \ldots \wedge (p_{m1} \vee p_{m2} \vee \ldots \vee p_{mn})$

Dạng chuẩn tuyển là tuyển ($\vee$) của những phép toán hội ($\wedge$):

$(p_{11} \wedge p_{12} \wedge \ldots \wedge p_{1n}) \vee \ldots \vee (p_{m1} \wedge p_{m2} \wedge \ldots \wedge p_{mn})$, trong đó p_{ij} là các biểu thức nguyên tố.

ĐẠI SỐ MỆNH ĐỀ

Bảng các tương đương logic thường dùng

Đặt T= hằng đúng, F = hằng sai

1. $p \wedge F \Leftrightarrow F$
2. $p \vee T \Leftrightarrow T$
3. $p \vee F \Leftrightarrow p$
4. $p \wedge T \Leftrightarrow p$
5. $p \wedge p \Leftrightarrow p$
6. $p \vee p \Leftrightarrow p$
7. $\neg(\neg p) \Leftrightarrow p$
8. $p \wedge \neg p \Leftrightarrow F$
9. $p \vee \neg p \Leftrightarrow T$
10. $p \wedge q \Leftrightarrow q \wedge p$

Bảng các tương đương logic thường dùng (tt)

11. $p \vee q \Leftrightarrow q \vee p$
12. $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
13. $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
14. $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
15. $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
16. $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$
17. $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$
18. $(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$
19. $p \vee (p \wedge q) = p$
20. $p \wedge (p \vee q) = p$

Xử lý truy vấn trong môi trường phân tán

Ví dụ minh họa: xét CSDL công ty phần mềm đã cho

Từ các quan hệ: E= E (MANV, TENNV, CHUCVU) và

G= HOSO (MANV, MADA, NHIEMVU, THOIGIAN).

Xét truy vấn:"*Tìm tên các nhân viên làm dự án có mã số J1 với thời gian 12 hoặc 24 tháng*" .

Truy vấn trên được biểu diễn trong SQL:

```
SELECT      E. TENNV
FROM        E, G
WHERE       E.MANV= G.MANV
      AND   G.MADA="J1"
      AND   THOIGIAN=12  OR  THOIGIAN=24
```

Điều kiện trong dạng chuẩn hội là:

$E.MANV=G.MANV \wedge G.MADA="J1" \wedge (THOIGIAN=12 \vee THOIGIAN=24)$

Điều kiện trong dạng chuẩn tuyển là:

$(E.MANV=G.MANV \wedge G.MADA="J1" \wedge THOIGIAN=12) \vee$

$(E.MANV=G.MANV \wedge G.MADA="J1" \wedge THOIGIAN=24)$

Xử lý truy vấn trong môi trường phân tán

2. Phân tích

Mục đích: Phát hiện ra những thành phần không đúng (sai kiểu hoặc sai ngữ nghĩa) và loại bỏ chúng sớm nhất nếu có thể.

***Truy vấn sai kiểu**:* nếu một thuộc tính bất kỳ hoặc tên quan hệ của nó không được định nghĩa trong lược đồ tổng thể, hoặc phép toán áp dụng cho các thuộc tính sai kiểu.

Ví dụ: truy vấn dưới đây là sai kiểu

```
SELECT    E#
FROM      E
WHERE     E.TENNV > 200
```

vì hai lý do:

- Thuộc tính E# không khai báo trong lược đồ
- Phép toán “>200” không thích hợp với kiểu chuỗi của thuộc tính E.TENNV

Xử lý truy vấn trong môi trường phân tán

Truy vấn sai ngữ nghĩa*:* nếu các thành phần của nó không tham gia vào việc tạo ra kết quả.

Để xác định truy vấn có sai về ngữ nghĩa hay không, ta dựa trên việc biểu diễn truy vấn như một đồ thị gọi là *đồ thị truy vấn*. Đồ thị này được xác định bởi các truy vấn liên quan đến phép chọn, chiếu và nối. Nếu đồ thị truy vấn mà không liên thông thì truy vấn là sai ngữ nghĩa

Xử lý truy vấn trong môi trường phân tán

Đồ thị truy vấn:

- Có một nút dùng để biểu diễn cho quan hệ kết quả
- Các nút khác biểu diễn cho các toán hạng trong câu truy vấn (các quan hệ)
- Cạnh nối giữa hai nút mà không phải là nút kết quả thì biểu diễn một **phép nối.**
- Cạnh có nút đích là nút kết quả thì biểu diễn một **phép chiếu**.
- Một nút không phải là nút kết quả có thể được gán nhãn bởi **phép chọn** hoặc **phép tự nối** (seft-join: nối của quan hệ với chính nó).

Đồ thị kết nối:

- Là một đồ thị con của đồ thị truy vấn (join graph), trong đó chỉ có phép nối.

Xử lý truy vấn trong môi trường phân tán

Ví dụ: Từ các quan hệ E=E (MANV, TENNV, CHUCVU) và G = HOSO (MANV, MADA, NHIEMVU, THOIGIAN) và J=DUAN (MADA, TENDA, NGANSACH).

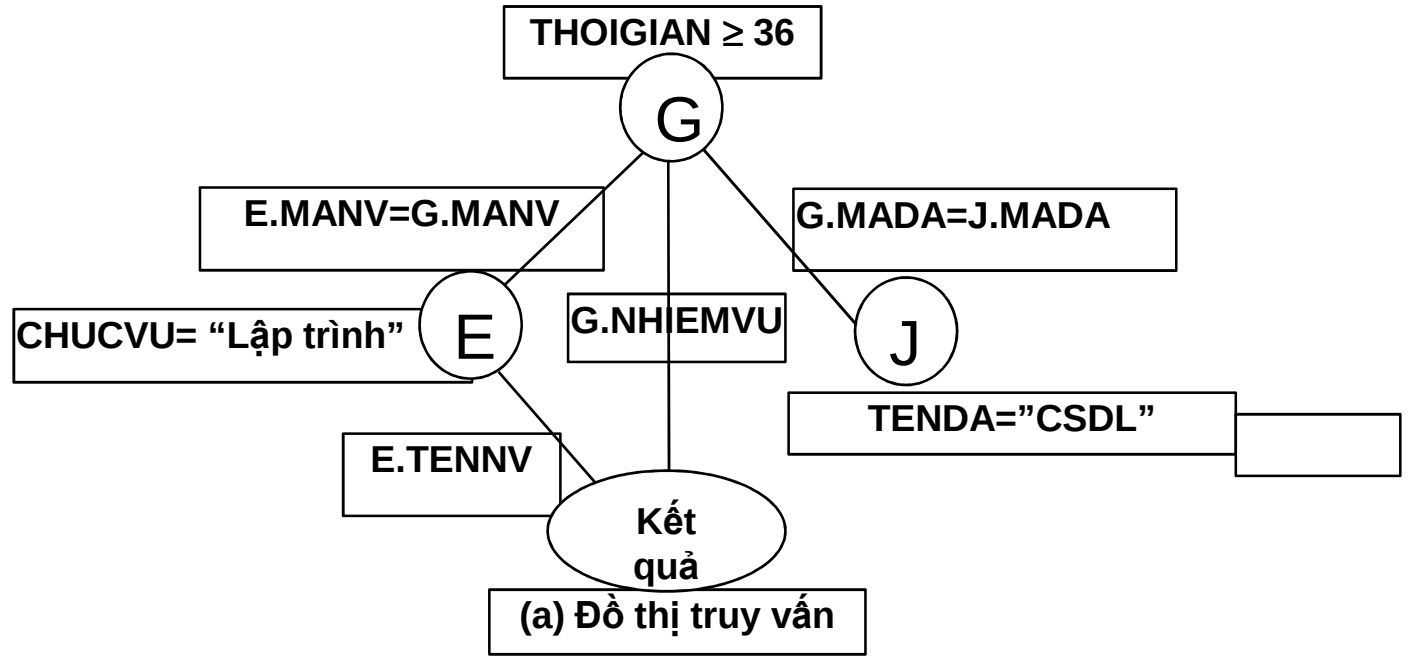
Hãy xác định "***Tên và nhiệm vụ các lập trình viên làm dự án CSDL có thời gian lớn hơn 3 năm.***"

Truy vấn SQL tương ứng là:

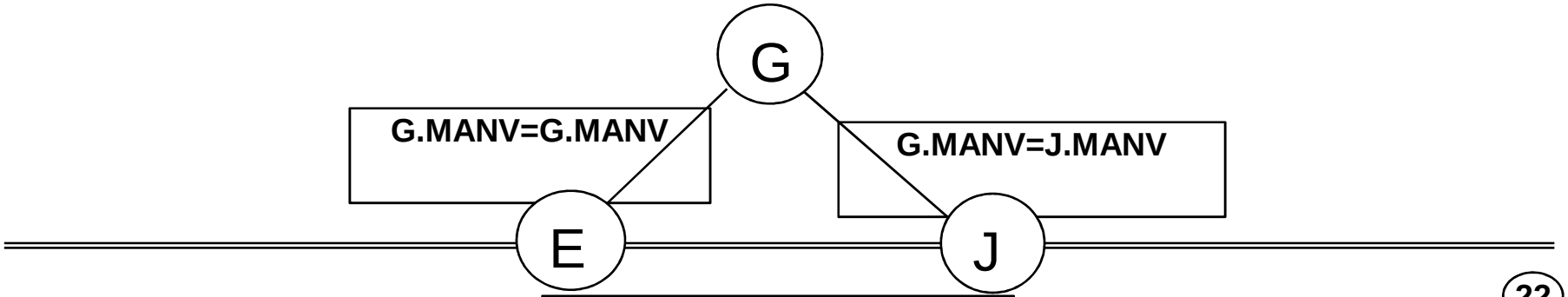
```
SELECT    E.TENNV, G.NHIEMVU
FROM      E, G, J
WHERE     E.MANV=G.MANV
          AND   G.MADA.= J.MADA
          AND   TENDA="CSDL"
          AND   THOIGIAN≥ 36
          AND   NHIEMVU="LTRINH"
```

Xử lý truy vấn trong môi trường phân tán

Đồ thị truy vấn và đồ thị kết nối tương ứng



(a) Đồ thị truy vấn



(b) Đồ thị kết nối tương ứng

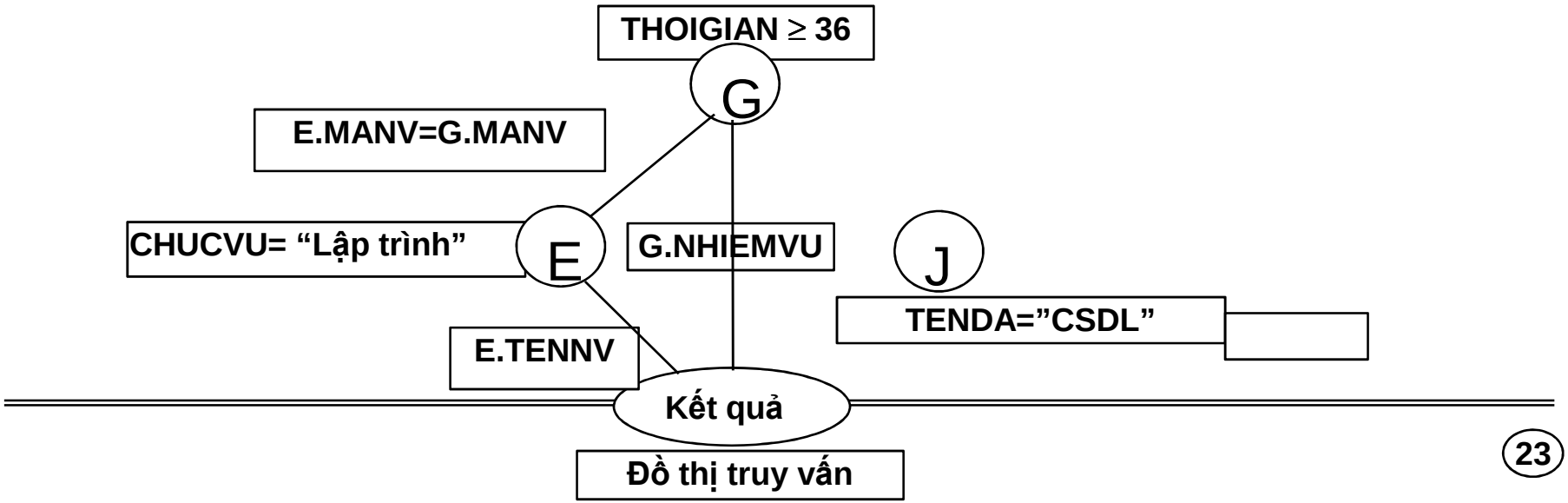
Xử lý truy vấn trong môi trường phân tán

Xét câu truy vấn SQL tương ứng:

SELECT E.TENNV, NHIEMVU
FROM E, G, J
WHERE E.MANV=G.MANV
AND TENDA=”CSDL”
AND THOIGIAN ≥ 36
AND CHUCVU=”Lập trình”

thiếu AND G.MADA=J.MADA

Truy vấn này là sai ngữ nghĩa vì đồ thị truy vấn của nó không liên thông.



Xử lý truy vấn trong môi trường phân tán

3. Loại bỏ dư thừa

- Điều kiện trong các truy vấn có thể có chứa các vị từ dư thừa.
- Một đánh giá sơ sài về một điều kiện dư thừa có thể dẫn đến lặp lại một số công việc.
- Sự dư thừa vị từ và dư thừa công việc có thể được loại bỏ bằng cách làm đơn giản hoá các điều kiện thông qua các luật luỹ đẳng sau:

1. $p \wedge p \Leftrightarrow p$
2. $p \vee true \Leftrightarrow true$
3. $p \vee p \Leftrightarrow p$
4. $p \wedge \neg p \Leftrightarrow false$
5. $p \wedge true \Leftrightarrow p$
6. $p \vee \neg p \Leftrightarrow true$
7. $p \vee false \Leftrightarrow p$
8. $p_1 \wedge (p_1 \vee p_2) \Leftrightarrow p_1$
9. $p \wedge false \Leftrightarrow false$
10. $p_1 \vee (p_1 \wedge p_2) \Leftrightarrow p_1$

Ví dụ: Đơn giản hoá câu truy vấn sau:

Xử lý truy vấn trong môi trường phân tán

SELECT E.CHUCVU
FROM E
WHERE (NOT(E.CHUCVU=”Lập trình”)
AND (E.CHUCVU=”Lập trình” **OR** E.CHUCVU=”Kỹ sư điện”)
AND NOT(E.CHUCVU=”Kỹ sư điện”)
OR E.TENNV=”Dũng”

Đặt p1:<CHUCVU=”Lập trình”>, p2:<CHUCVU=”Kỹ sư điện”>,
p3: <E.TENNV=”Dũng”>.
Các vị từ sau mệnh đề WHERE được mô tả lại:
p: $(\neg p1 \wedge (p1 \vee p2) \wedge \neg p2) \vee p3$
$\Leftrightarrow ((\neg p1 \wedge p1 \wedge \neg p2) \vee (\neg p1 \wedge p2 \wedge \neg p2)) \vee p3$ (áp dụng luật 7)
$\Leftrightarrow ((false \wedge \neg p2) \vee (\neg p1 \wedge false)) \vee p3$ (áp dụng luật 5)
$\Leftrightarrow (false \vee false) \vee p3$ (áp dụng luật 4)
$\Leftrightarrow P3$
Vậy câu truy vấn được biến đổi thành:
SELECT E.CHUCVU
FROM E
WHERE E.TENNV=”Dũng”

Xử lý truy vấn trong môi trường phân tán

4. Viết lại

Bước này được chia làm hai bước con như sau:

- Biến đổi trực tiếp truy vấn phép tính sang đại số quan hệ.
- Cấu trúc lại truy vấn đại số quan hệ để cải thiện hiệu quả thực hiện. Đại số quan hệ là một cây mà nút lá biểu diễn một quan hệ trong CSDL, các nút không lá là các quan hệ trung gian được sinh ra bởi các phép toán đại số quan hệ.

Xử lý truy vấn trong môi trường phân tán

Cách chuyển một truy vấn phép tính quan hệ thành một cây đại số quan hệ:

- Các nút lá khác nhau được tạo cho mỗi biến bộ khác nhau (tương ứng một quan hệ). Trong SQL các nút lá chính là các quan hệ trong mệnh đề FROM.
- Nút gốc được tạo ra bởi một phép chiếu lên các thuộc tính kết quả. Trong SQL nút gốc được xác định qua mệnh đề SELECT.
- Điều kiện (mệnh đề WHERE trong SQL) được biến đổi thành dãy các phép toán đại số thích hợp (phép chọn, nối, phép hợp, v.v...) đi từ lá đến gốc, có thể thực hiện theo thứ tự xuất hiện của các vị từ và các phép toán.

Xử lý truy vấn trong môi trường phân tán

Ví dụ:

Truy vấn *"Tìm tên các nhân viên không phải là "Dũng", làm việc cho dự án CSDL với thời gian một hoặc hai năm".*

Biểu diễn truy vấn này trong SQL là:

```
SELECT      E.TENNV
FROM        J, G, E
WHERE       G.MANV=E.MANV
      AND   G.MADA= J.MADA
      AND   E.TENNV <> "Dũng"
      AND   J.TENDA= "CSDL"
      AND   (THOIGIAN=12  OR  THOIGIAN=24)
```

Cơ sở dữ liệu của một công ty máy tính

NHANVIEN (E)

MANV	TENNV	CHUCVU
A1	Nam	Phân tích HT
A2	Trung	Lập trình viên
A3	Đông	Phân tích HT
A4	Bắc	Phân tích HT
A5	Tây	Lập trình viên
A6	Hùng	Kỹ sư điện
A7	Dũng	Phân tích HT
A8	Chiến	Thiết kế DL

HOSO(G)

MANV	MADA	NHIEMVU	THOIGIAN
A1	D1	Quản lý	12
A2	D1	Phân tích	34
A2	D2	Phân tích	6
A3	D3	Kỹ thuật	12
A3	D4	Lập trình	10
A4	D2	Quản lý	6
A5	D2	Quản lý	20
A6	D4	Kỹ thuật	36
A7	D3	Quản lý	48
A8	D3	Lập trình	15

DUAN (J)

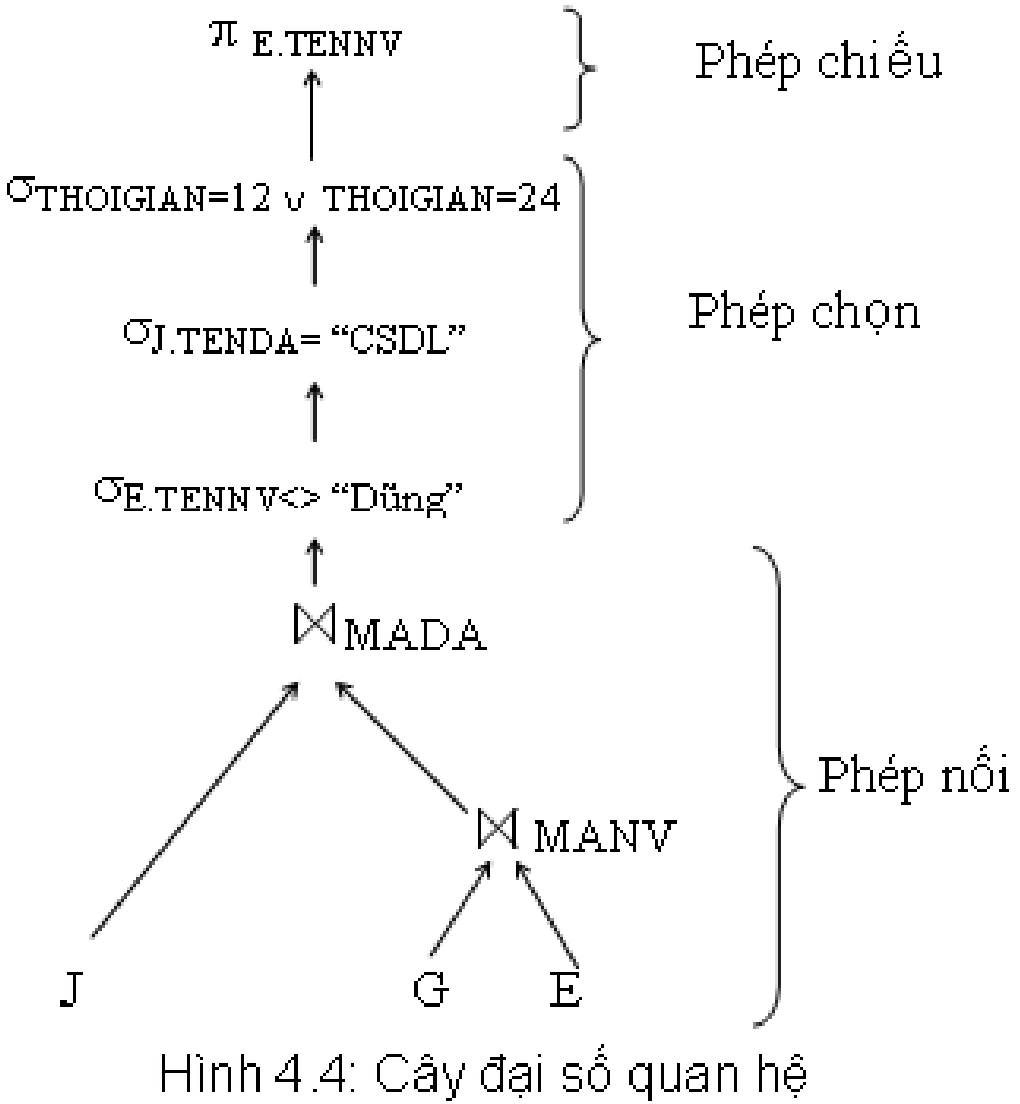
MADA	TENDA	NGANSACH
D1	CSDL	20000
D2	CÀI ĐẶT	12000
D3	BẢO TRÌ	28000
D4	PHÁT TRIỂN	25000

TLUONG (S)

CHUCVU	LUONG
Kỹ sư điện	1000
Phân tích HT	2500
Lập trình viên	3000
Thiết kế DL	4000

Xử lý truy vấn trong môi trường phân tán

```
SELECT E.TENNV
FROM   J, G, E
WHERE G.MANV=E.MANV
    AND G.MADA= J.MADA
    AND E.TENNV <> "Dũng"
    AND J.TENDA= "CSDL"
    AND (THOIGIAN=12  OR
                THOIGIAN=24)
```



Hình 4.4: Cây đại số quan hệ

Xử lý truy vấn trong môi trường phân tán

6 luật biến đổi phép toán đại số quan hệ:

Mục đích: dùng để biến đổi cây đại số quan hệ thành các cây tương đương (trong đó có thể có cây tối ưu).

Giả sử R, S, T là các quan hệ, R được định nghĩa trên toàn bộ thuộc tính $A=\{A_1, ..., A_n\}$, S được định nghĩa trên toàn bộ thuộc tính $B=\{B_1, ..., B_n\}$.

***1.Tính giao hoán của các phép toán hai ngôi*:**

Phép tích Decartes và phép nối hai quan hệ có tính giao hoán.

i. $\mathbf{R \times S \Leftrightarrow S \times R}$ ii. $\mathbf{R \bowtie S \Leftrightarrow S \bowtie R}$

2. *Tính kết hợp của các phép toán hai ngôi:*

Phép tích Decartes và phép nối hai quan hệ có tính kết hợp.

i. $\mathbf{(R \times S) \times T \Leftrightarrow R \times (S \times T)}$ ii. $\mathbf{(R \bowtie S) \bowtie T \Leftrightarrow R \bowtie (S \bowtie T)}$

Xử lý truy vấn trong môi trường phân tán

3. ***Tính luỹ đẳng của những phép toán một ngôi***

- Dãy các phép chiếu khác nhau trên cùng quan hệ được tổ hợp thành một phép chiếu và ngược lại:

$$\Pi_{A'}(\Pi_{A''}(R)) \Leftrightarrow \Pi_{A'}(R) \quad A', A'' \subseteq R \text{ và } A' \subseteq A''$$

- Dãy các phép chọn khác nhau $\sigma_{p_i(A_i)}$ trên cùng một quan hệ, với p_i là một vị từ được gán vào thuộc tính A_i , có thể được tổ hợp thành một phép chọn.

$$\sigma_{p_1(A_1)}(\sigma_{p_2(A_2)}(R)) = \sigma_{p_1(A_1) \wedge p_2(A_2)}(R)$$

Xử lý truy vấn trong môi trường phân tán

4. Phép chọn giao hoán với phép chiếu

$$\Pi_{A_1,\ldots,A_n}(\sigma_{p(A_p)}(R)) = \Pi_{A_1,\ldots,A_n}(\sigma_{p(A_p)}(\Pi_{A_1,\ldots,A_n,A_p}(R)))$$

Nếu A_p là thành viên của $\{A_1, \ldots, A_n\}$, biểu thức trên trở thành

$$\Pi_{A_1,\ldots,A_n}(\sigma_{p(A_p)}(R)) = \sigma_{p(A_p)}(\Pi_{A_1,\ldots,A_n,A_p}(R))$$

5. Phép chọn giao hoán với những phép toán hai ngôi

- Phép chọn với phép nhân: $\sigma_{p(A_p)}(R \times S) \Leftrightarrow \sigma_{p(A_p)}(R) \times S$
- Phép chọn với phép nối:

$$\sigma_{p(A_i)}(R \bowtie_{(A_i,B_k)} S) \Leftrightarrow \sigma_{p(A_i)}(R) \bowtie_{(A_i,B_k)} S$$

- Phép chọn với phép hợp: Nếu R và T cùng bộ thuộc tính.

$$\sigma_{p(A_i)}(R \cup T) \Leftrightarrow \sigma_{p(A_i)}(R) \cup \sigma_{p(A_i)}(T)$$

Xử lý truy vấn trong môi trường phân tán

6. Phép chiếu giao hoán với những phép toán hai ngôi

- ***Phép chiếu và tích Decartes***:
  Nếu C=A'∪B' với A'⊆ A, B'⊆ B, và A, B là tập các thuộc tính trên quan hệ R, S ta có:

$$\Pi_C(R \times S) = \Pi_{A'}(R) \times \Pi_{B'}(S)$$

- ***Phép chiếu và phép nối***:

$$\Pi_C(R \bowtie_{p(A_i, B_j)} S) = \Pi_{A'}(R) \bowtie_{p(A_i, B_j)} \Pi_{B'}(S)$$

- ***Phép chiếu và phép hợp:***

$$\Pi_C(R \cup S) = \Pi_{A'}(R) \cup \Pi_{B'}(S)$$

Chú ý: Việc sử dụng sáu luật trên có khả năng sinh ra nhiều cây đại số quan hệ tương đương nhau. Vấn đề là xác định cho được cây tối ưu.

Xử lý truy vấn trong môi trường phân tán

Chú ý:

Trong giai đoạn tối ưu, sự so sánh các cây có thể thực hiện dựa trên chi phí dự đoán của chúng. Tuy nhiên, nếu số lượng các cây quá lớn thì cách tiếp cận này sẽ không hiệu quả. Chúng ta có thể dùng 6 luật trên để cấu trúc lại cây, nhằm loại bỏ những cây đại số quan hệ "tồi".

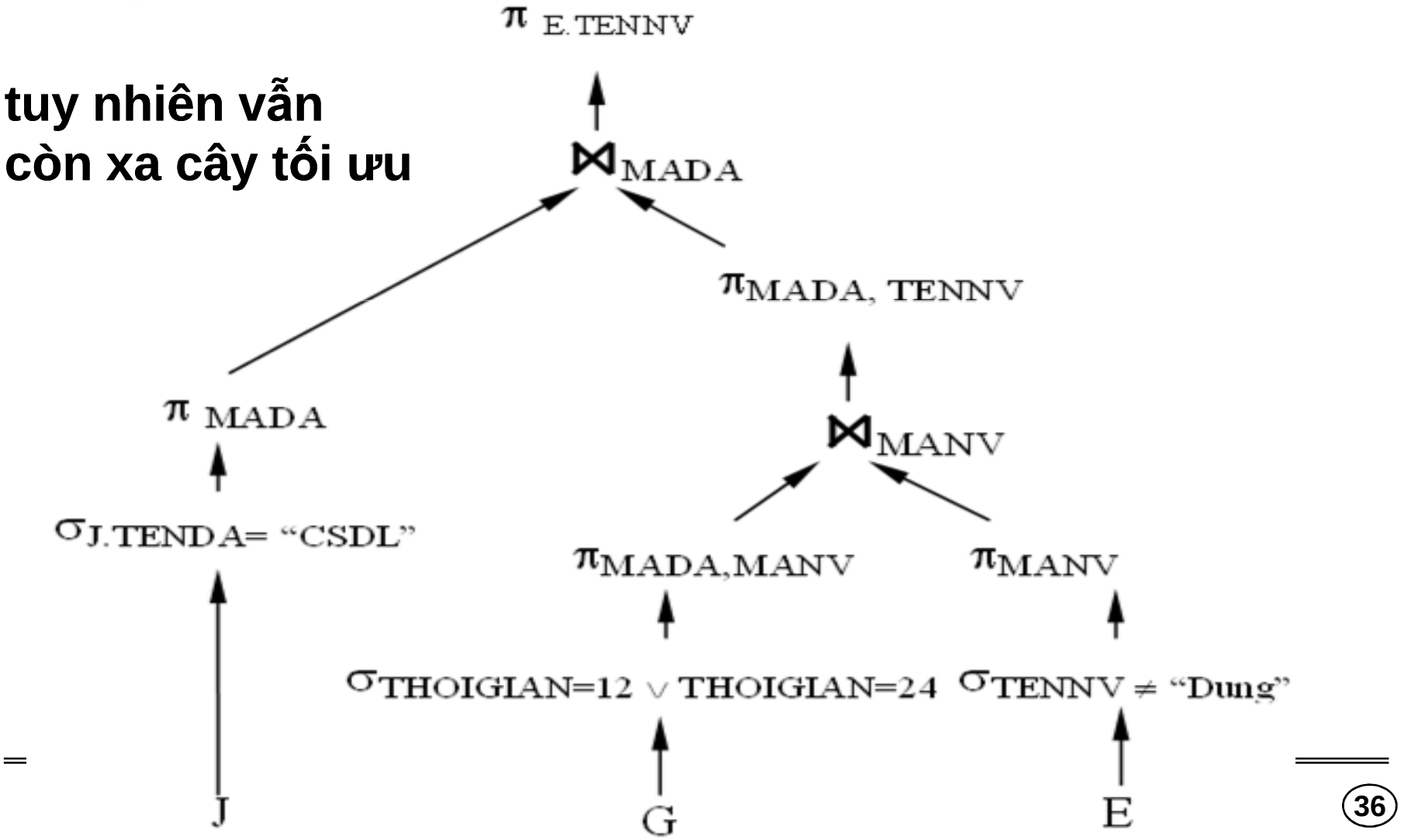
Các luật trên có thể sử dụng theo bốn cách như sau:

- Phân rã các phép toán một ngôi, đơn giản hóa biểu thức truy vấn .
- Nhóm các phép toán một ngôi trên cùng một quan hệ để giảm số lần thực hiện.
- Giao hoán các phép toán một ngôi với các phép toán hai ngôi để ưu tiên cho một số phép toán (chẳng hạn phép chọn).
- Sắp thứ tự các phép toán hai ngôi trong thực hiện truy vấn.

Xử lý truy vấn trong môi trường phân tán

Ví dụ: Cấu trúc lại cây truy vấn ở ví dụ trên, cho ra cây kết quả tốt hơn cây ban đầu.

tuy nhiên vẫn còn xa cây tối ưu



=

Xử lý truy vấn trong môi trường phân tán

2. Định vị dữ liệu phân tán-Tối ưu hóa cục bộ

- Lớp định vị biến đổi một truy vấn đại số quan hệ tổng thể thành một truy vấn đại số được biểu thị trên các mảnh vật lý.
- Sử dụng thông tin được lưu trữ trên các lược đồ phân mảnh để định vị.
- Chương trình đại số quan hệ xây dựng lại quan hệ tổng thể từ các phân mảnh của nó gọi là *chương trình định vị*.
- Truy vấn có được từ chương trình định vị gọi là truy vấn ban đầu.
- ***Chú ý***: Trong phần dưới đây, với *mỗi kiểu phân mảnh* chúng ta sẽ biểu diễn một *kỹ thuật rút gọn* để sinh ra truy vấn được tối ưu và đơn giản hoá.

Xử lý truy vấn trong môi trường phân tán

1. Rút gọn theo phân mảnh ngang nguyên thuỷ

Xét quan hệ E(MANV,TENNV,CHUCVU). Tách quan hệ này thành ba mảnh ngang E_1, E_2 và E_3 như sau:

$E_1=\sigma_{MANV \leq "E3"}(E)$ $E_2=\sigma_{"E3"< MANV \leq "E6"}(E)$ $E_3=\sigma_{MANV > "E6"}(E)$

- Chương trình định vị cho quan hệ E: $E = E_1 \cup E_2 \cup E_3$.
- Dạng ban đầu của bất kỳ truy vấn nào được xác định trên E là có được bằng cách thay thế nó bởi $E_1 \cup E_2 \cup E_3$.
- Việc rút gọn các truy vấn trên các quan hệ đã được phân mảnh ngang bao gồm việc *xác định câu truy vấn*, sau khi đã cấu trúc lại cây con. Điều này sẽ sinh ra một số quan hệ rỗng, và sẽ *loại bỏ* chúng.
- Phân mảnh ngang có thể được khai thác để làm đơn giản cả phép chọn và phép nối.

Xử lý truy vấn trong môi trường phân tán

a. ***Rút gọn với phép chọn***: cho một quan hệ R được phân mảnh ngang thành $R_1, R_2, ..., R_n$ với $R_j = \sigma_{p_j}(R)$

Luật 1: $\sigma_{p_j}(R_j) = \phi$ nếu $\forall x \in R : \neg(p_i(x) \wedge p_j(x))$. Trong đó, p_i, p_j là vị từ chọn, x là bộ dữ liệu, p(x) là vị từ p chiếm giữ x.

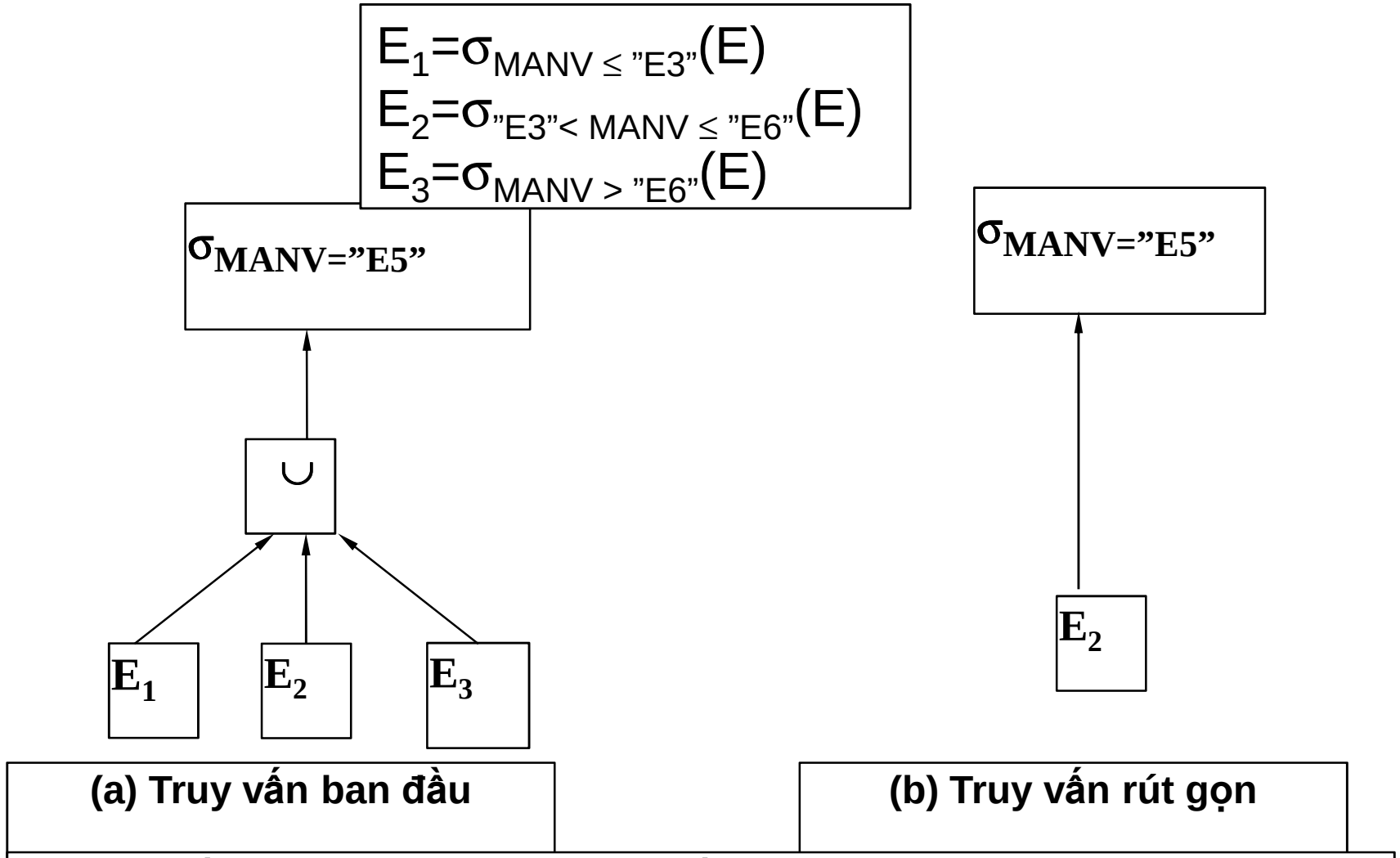
Ví dụ: Hãy rút gọn truy vấn

SELECT *
FROM E
WHERE MANV=”E5”

Với E được tách thành ba mảnh ngang E_1, E_2 và E_3 :

$E_1 = \sigma_{MANV \leq \text{”E3”}}(E)$ $E_2 = \sigma_{\text{”E3”} < MANV \leq \text{”E6”}}(E)$ $E_3 = \sigma_{MANV > \text{”E6”}}(E)$

Xử lý truy vấn trong môi trường phân tán



Rút gọn bằng cách sử dụng tính chất giao hoán phép chọn với phép hợp, chúng ta thấy vị từ chọn đối lập với vị từ E1 và E3 nên sinh ra các quan hệ rỗng.

Xử lý truy vấn trong môi trường phân tán

b. Rút gọn với phép nối

- Các phép nối trên quan hệ đã được phân mảnh ngang có thể đơn giản khi chúng được phân mảnh theo thuộc tính nối.
- Việc rút gọn được thực hiện dựa trên tính phân phối giữa phép nối và phép hợp và loại bỏ các phép nối vô ích.
- Với tính chất, $(R_1 \cup R_2) \bowtie R_3 = (R_1 \bowtie R_3) \cup (R_2 \bowtie R_3)$, R_i là các phân mảnh. Chúng ta có thể xác định được các phép nối vô ích của các mảnh khi các điều kiện nối mâu thuẫn nhau. Sau đó, dùng luật 2 dưới đây để loại bỏ các phép nối vô ích.

Xử lý truy vấn trong môi trường phân tán

Luật 2: **$R_i \bowtie R_j = \phi$ nếu $\forall x \in R_i, \forall y \in R_j : \neg (p_i(x) \wedge p_j(y))$.** Trong đó R_i, R_j được xác định theo các vị từ p_i, p_j trên cùng thuộc tính.

Nhận xét:

- Việc xác định các phép nối vô ích được thực hiện bằng cách chỉ xem xét các vị từ mảnh.
- Truy vấn rút gọn không phải luôn tốt hơn hoặc đơn giản hơn truy vấn ban đầu.
- Một thuận lợi của truy vấn rút gọn là những phép nối có thể thực hiện song song.

Xử lý truy vấn trong môi trường phân tán

Ví dụ: Giả sử quan hệ E được phân mảnh thành các mảnh

$E_1=\sigma_{MANV \le "E3"}(E)$ $E_2=\sigma_{"E3"< MANV \le "E6"}(E)$ $E_3=\sigma_{MANV > "E6"}(E)$

Quan hệ G được phân làm hai mảnh:

$G_1=\sigma_{MANV\le"E3"}(G)$ và $G_2=\sigma_{MANV>"E3"}(G)$.

Nhận xét:

- E_1 và G_1 được định nghĩa bởi cùng vị từ.
- Vị từ định nghĩa G_2 là hợp của các định nghĩa của những vị từ E_2 và E_3.

Xét truy vấn

```
SELECT    *
FROM      E, G
WHERE     E.MANV=G.MANV
```

Xử lý truy vấn trong môi trường phân tán

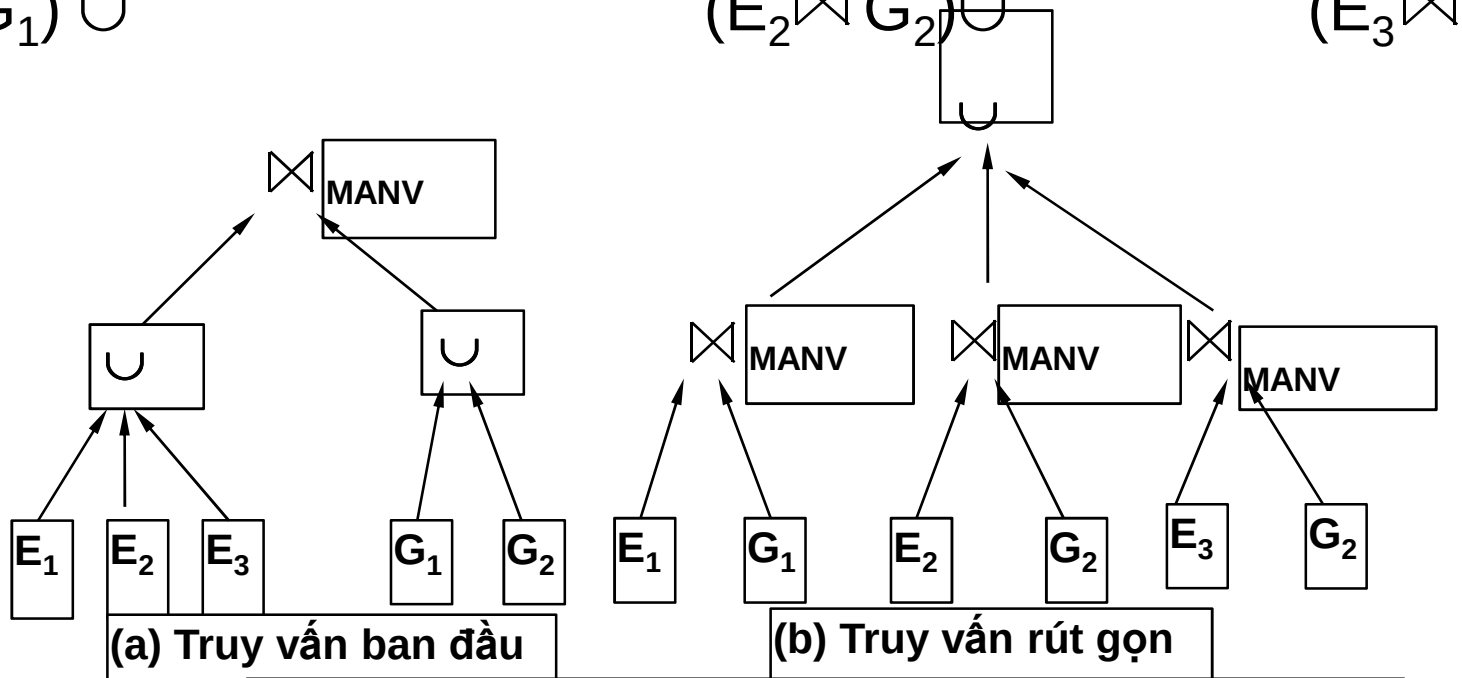
$E_1=\sigma_{MANV \le "E3"}(E)$ $E_2=\sigma_{"E3"< MANV \le "E6"}(E)$ $E_3=\sigma_{MANV > "E6"}(E)$

$G_1=\sigma_{MANV\le"E3"}(G)$ $G_2=\sigma_{MANV>"E3"}(G)$.

$E \bowtie G = (E_1 \cup E_2 \cup E_3) \bowtie (G_1 \cup G_2)$

$= (E_1 \bowtie G_1) \cup (E_1 \bowtie G_2) \cup (E_2 \bowtie G_1) \cup (E_2 \bowtie G_2) \cup (E_3 \bowtie G_1) \cup (E_3 \bowtie G_2)$

$= (E_1 \bowtie G_1) \cup (E_2 \bowtie G_2) \cup (E_3 \bowtie G_2)$



Hình 4.8: Sự rút gọn phân mảnh ngang với phép nối

Xử lý truy vấn trong môi trường phân tán

2. Rút gọn phân mảnh dọc

- Chức năng của việc phân mảnh dọc là tách quan hệ dựa vào thuộc tính của các phép chiếu.
- Vì phép toán xây dựng lại đối với phân mảnh dọc là nối, nên chương trình định vị một quan hệ đã được phân mảnh dọc là nối của các mảnh trong vùng thuộc tính chung.

Ví dụ: Quan hệ E được phân mảnh dọc thành E_1, E_2, với thuộc tính khoá MANV được lặp lại như sau:

$E_1 = \Pi_{MANV,TENNV}(E)$ và $E_2 = \Pi_{MANV,CHUCVU}(E)$

Chương trình định vị là: $E = E_1 \bowtie_{MANV} E_2$

- Các truy vấn trên phân mảnh dọc có thể rút gọn bằng cách xác định các quan hệ trung gian vô ích và loại bỏ các cây con chứa chúng.
- Các phép chiếu trên một phân mảnh dọc không có thuộc tính chung với các thuộc tính chiếu (ngoại trừ khóa của quan hệ) là vô ích, mặc dù các quan hệ là khác rỗng.

Xử lý truy vấn trong môi trường phân tán

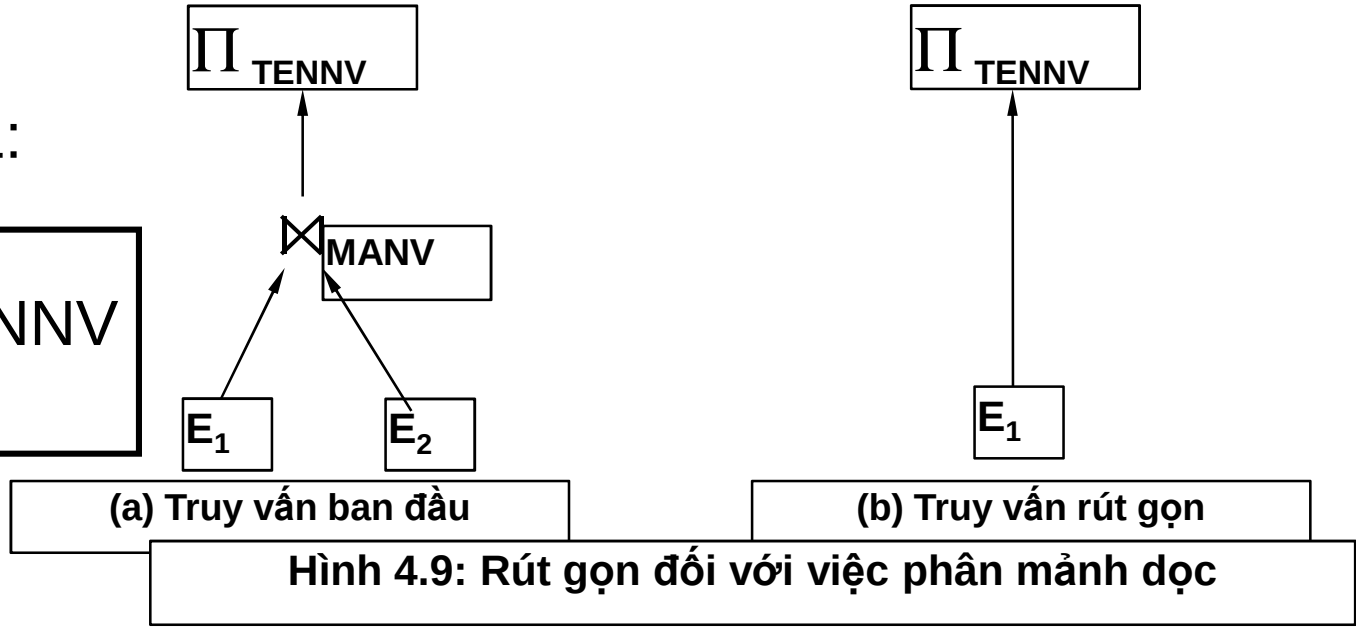
Luật 3: $\Pi_{D,K}(R_i)$ là vô ích nếu $D \cap A' = \phi$. Trong đó, quan hệ R xác định trên $A=\{A_1, ..., A_n\}$; $R = \Pi_{A'}(R)$, $A' \subseteq A$, K là khoá của quan hệ, $K \subset A$, D là tập các thuộc tính chiếu, $D \subset A$.

Ví dụ: Với quan hệ E được phân mảnh dọc như sau:

$E_1 = \Pi_{MANV,TENNV}(E)$ và $E_2 = \Pi_{MANV,CHUCVU}(E)$

Xét truy vấn SQL:

```
SELECT    TENNV
FROM      E
```



Hình 4.9: Rút gọn đối với việc phân mảnh dọc

Nhận xét: phép chiếu trên E_2 là vô ích vì TENNV không có trong E_2, nên phép chiếu chỉ cần gán vào E_1

Xử lý truy vấn trong môi trường phân tán

3. Rút gọn theo phân mảnh gián tiếp

- Sự phân mảnh ngang gián tiếp là một cách tách hai quan hệ để việc xử lý nối của các phép chọn và phép nối
- Nếu quan hệ R phụ thuộc vào sự phân mảnh ngang gián tiếp nhờ quan hệ S, thì các mảnh của R và S, mà có cùng giá trị thuộc tính nối sẽ được định vị tại cùng trạm. Ngoài ra, S có thể được phân mảnh tùy thuộc vào vị từ chọn.
- Khi các bộ của R được đặt tuỳ theo những bộ của S, thì sự phân mảnh gián tiếp chỉ nên sử dụng mối quan hệ một nhiều từ S$\rightarrow$R (i.e. với một bộ của S có thể phù hợp với n bộ của R, Nhưng với một bộ của R chỉ phù hợp với một bộ của S).
- Truy vấn trên các phân mảnh gián tiếp cũng có thể rút gọn được, nếu các vị từ phân mảnh mâu thuẫn nhau thì phép nối sẽ đưa ra quan hệ rỗng.
- Chương trình định vị một quan hệ đã được phân mảnh ngang gián tiếp là hợp của các mảnh.

Xử lý truy vấn trong môi trường phân tán

Ví dụ: Cho mối quan hệ một nhiều từ E đến G, quan hệ G (MANV, MADA, NHIEMVU, THOIGIAN) có thể được phân mảnh gián tiếp theo những luật sau:

$G_1 = G \bowtie_{MANV} E_1$ và $G_2 = G \bowtie_{MANV} E_2$.

Trong đó E được phân mảnh ngang như sau:

$E_1 = \sigma_{CHUCVU=\text{"Lập trình"}}(E)$ và $E_2 = \sigma_{CHUCVU\neq\text{"Lập trình"}}(E)$

Chương trình định vị cho một quan hệ đã được phân mảnh gián tiếp là hợp của các mảnh $G=G_1 \cup G_2$.

Để rút gọn các truy vấn trên phân mảnh gián tiếp này, phép nối sẽ đưa ra quan hệ rỗng nếu các vị từ phân mảnh mâu thuẫn nhau.

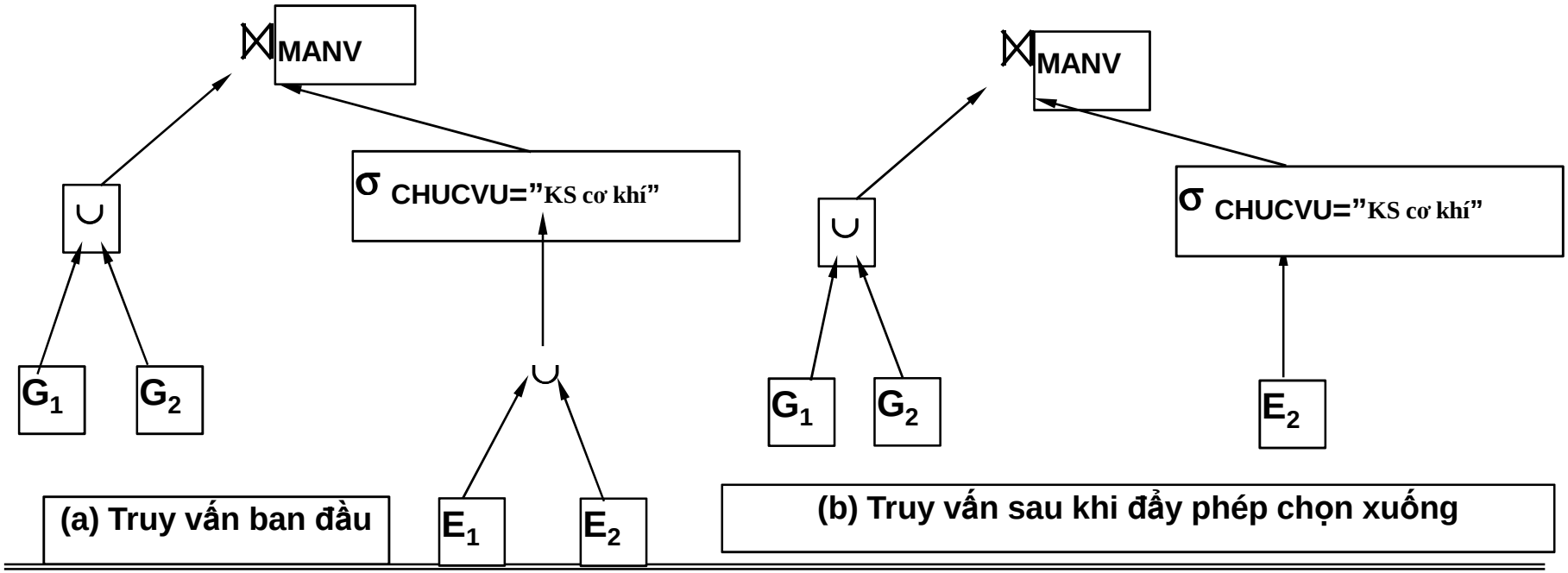
Ví dụ vị từ G_1 và E_2 mâu thuẫn nhau, nên $G_1 \bowtie E_2 = \phi$.

$G_1 = G \bowtie_{MANV} E_1$ và $G_2 = G \bowtie_{MANV} E_2$.

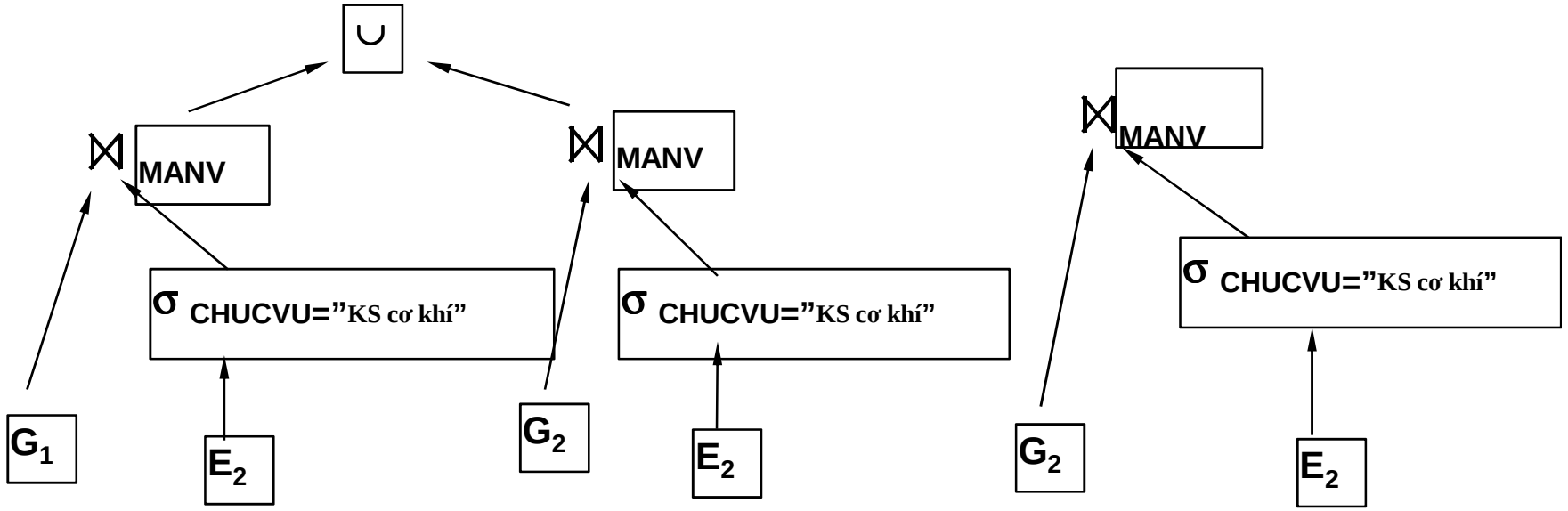
$E_1 = \sigma_{CHUCVU=\text{"Lập trình"}}(E)$ và $E_2 = \sigma_{CHUCVU\neq\text{"Lập trình"}}(E)$

Ví dụ: Xét truy vấn

```
SELECT      *
FROM        E, G
WHERE       G.MANV=E.MANV
        AND CHUCVU="KS cơ khí"
```



Chú ý: $(G_1 \cup G_2) \bowtie \sigma_{CHUCVU="ks\ cơ\ khí"}(E_2)$
$= (G_1 \bowtie \sigma_{CHUCVU="ks\ cơ\ khí"}(E_2)) \cup (G_2 \bowtie \sigma_{CHUCVU="ks\ cơ\ khí"}(E_2))$



(c) Truy vấn sau khi đẩy phép hợp lên

(d) Truy vấn đã rút gọn

Hình 4.10: Rút gọn của phân mảnh gián tiếp

Nhận xét:

- Truy vấn ban đầu trên các mảnh E_1, E_2, G_1 và G_2 tương ứng hình 4.10a.
- Bằng cách đẩy phép chọn xuống các mảnh E_1 và E_2, được truy vấn rút gọn ở hình 4.10b.
- Phân phối các phép nối với phép hợp, chúng ta thu được cây hình 4.10c.
- Cây con bên trái đưa ra một quan hệ rỗng, nên cây rút gọn có được trong hình 4.10d.

Xử lý truy vấn trong môi trường phân tán

4. Rút gọn theo phân mảnh hỗn hợp

- Sự phân mảnh hỗn hợp là sự kết hợp giữa phân dọc và phân mảnh ngang.
- Mục đích của phân mảnh hỗn hợp là hỗ trợ các truy vấn liên quan đến phép chiếu, phép chọn và phép nối
- Chương trình định vị cho một quan hệ đã phân mảnh hỗn hợp là phép hợp và phép nối của các mảnh.

Ví dụ: Xét quan hệ E được phân mảnh hỗn hợp như sau:

$E_1=\sigma_{MANV \leq "E4"}(\Pi_{MANV,TENNV}(E))$, $E_2=\sigma_{MANV > "E4"}(\Pi_{MANV,TENNV}(E))$

$E_3=\Pi_{MANV,CHUCVU}(E)$

Chương trình định vị là: $E = (E_1 \cup E_2) \bowtie_{MANV} E_3$

Xử lý truy vấn trong môi trường phân tán

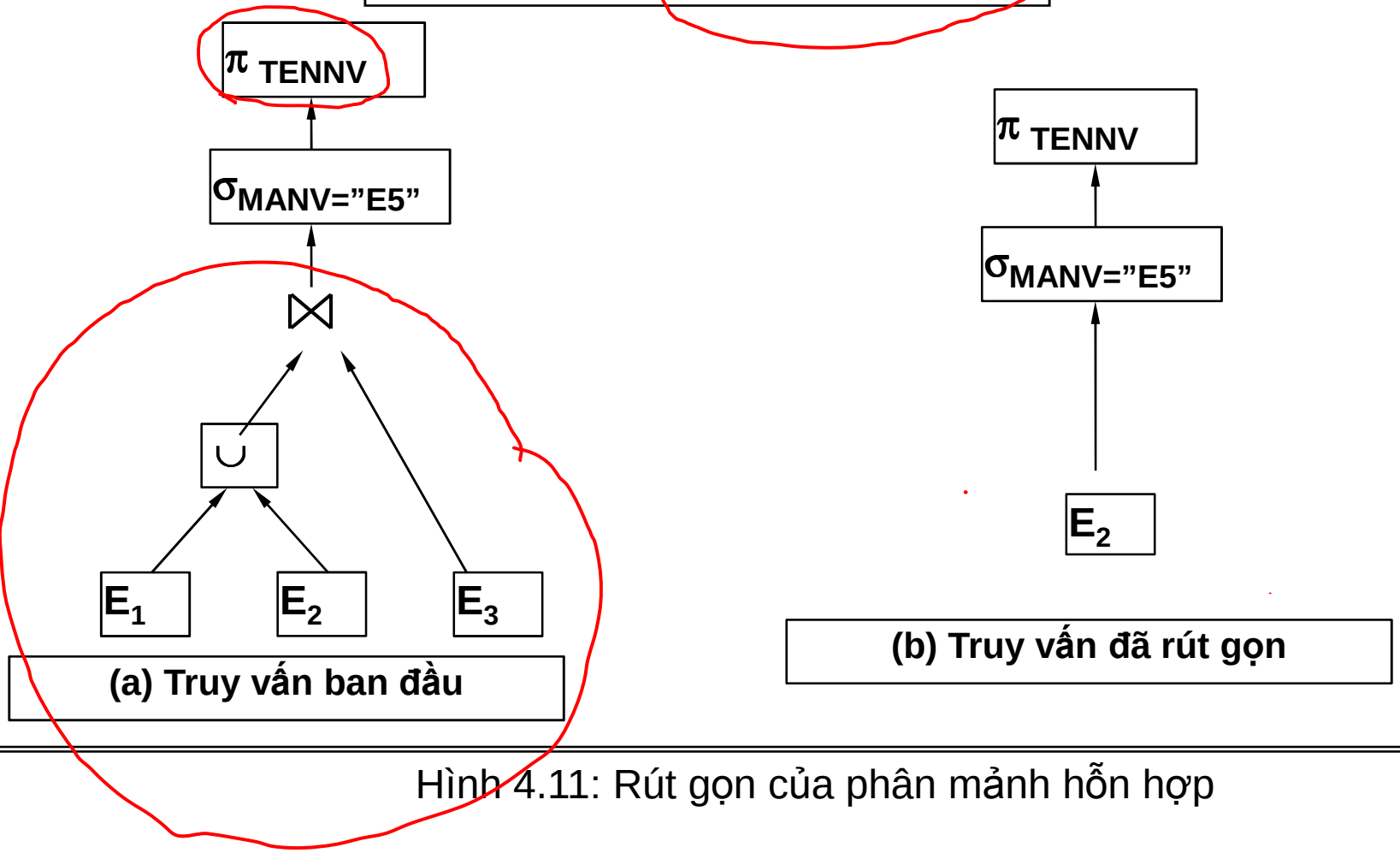
Các truy vấn trên các mảnh hỗn hợp có thể được rút gọn bằng cách kết hợp các luật sử dụng trong phân mảnh ngang nguyên thủy, phân mảnh dọc, phân mảnh ngang gián tiếp, tương ứng như sau:

1. Loại bỏ các quan hệ rỗng sinh bởi sự mâu thuẫn giữa các phép chọn trên các phân mảnh ngang.
2. Loại bỏ các quan hệ vô ích sinh bởi các phép chiếu trên các phân mảnh dọc.
3. Phân phối các phép nối với các phép hợp để tách và loại bỏ các phép nối vô ích.

Ví dụ: $E_1=\sigma_{MANV \le "E4"}(\Pi_{MANV,TENNV}(E))$, $E_2=\sigma_{MANV > "E4"}(\Pi_{MANV,TENNV}(E))$

$E_3=\Pi_{MANV,CHUCVU}(E)$

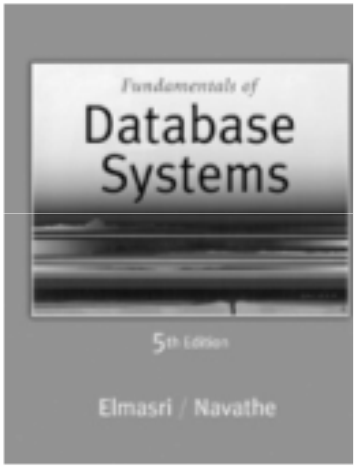
```
SELECT    TENNV
FROM      E
WHERE     MANV="E5"
```



Hình 4.11: Rút gọn của phân mảnh hỗn hợp

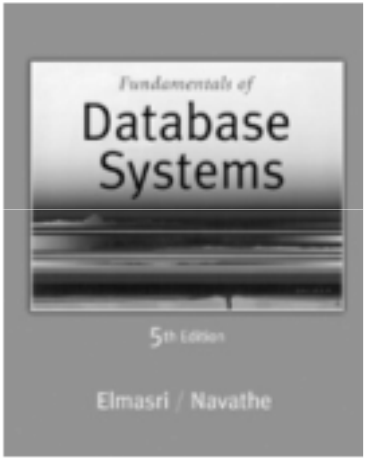
Chapter 17

Introduction to Transaction Processing Concepts and Theory



Chapter 18

Concurrency Control Techniques



Chương 6

Truy vấn trong CSDL phân tán