

# THIẾT KẾ HỆ CSDL QUAN HỆ - XÂY DỰNG CSDL VẬT LÝ

---

Nguyễn Đình Hóa

[dinhhoa@gmail.com](mailto:dinhhoa@gmail.com)

094-280-7711

# Câu truy vấn SQL

- Nội dung môn học tập trung vào các câu truy vấn  
Select-Project-Join

Select <attribute list>

From <relation list>

Where <condition list>

- Ví dụ: với lược đồ R(A, B, C):

Select B

From R

Where A = 'x' AND C > 20;

# Câu truy vấn SQL

Ví dụ về kết nối tự nhiên giữa hai lược đồ  
R(A,B,C) và S(C,D,E):

Select B, D

From R, S

Where R.A = 'x' AND S.E = 3 AND R.C = S.C;

R	A	B	C	S	C	D	E
	a	1	10		10	x	3
	b	1	20		20	y	3
	x	2	10		30	z	3
	d	2	35		40	x	1
	e	3	45		50	y	4

Select B,D

From R,S

Where R.A = 'x'

AND S.E = 3

AND R.C=S.C

Answer

B	D
2	x

# Câu truy vấn được thực hiện ra sao?

Select B, D

From R, S

Where R.A = 'x' AND S.E = 3 AND R.C = S.C;

**Giải pháp 1**

- Thực hiện tích Đề các
- Chọn bản ghi
- Chiếu.

$R \times S$ 

Select B,D

From R,S

Where R.A = 'x'

AND S.E = 3

AND R.C=S.C;

R.A	R.B	R.C	S.C	S.D	S.E
a	1	10	10	x	3
a	1	10	20	y	3
.					
.					
x	2	10	10	x	3
.					
.					

Bingo! →

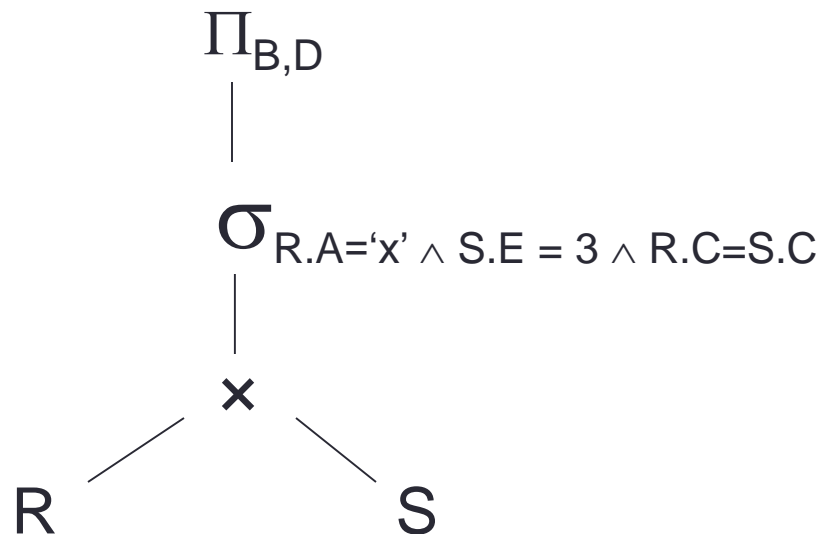
Got one...

Answer

B	D
2	x

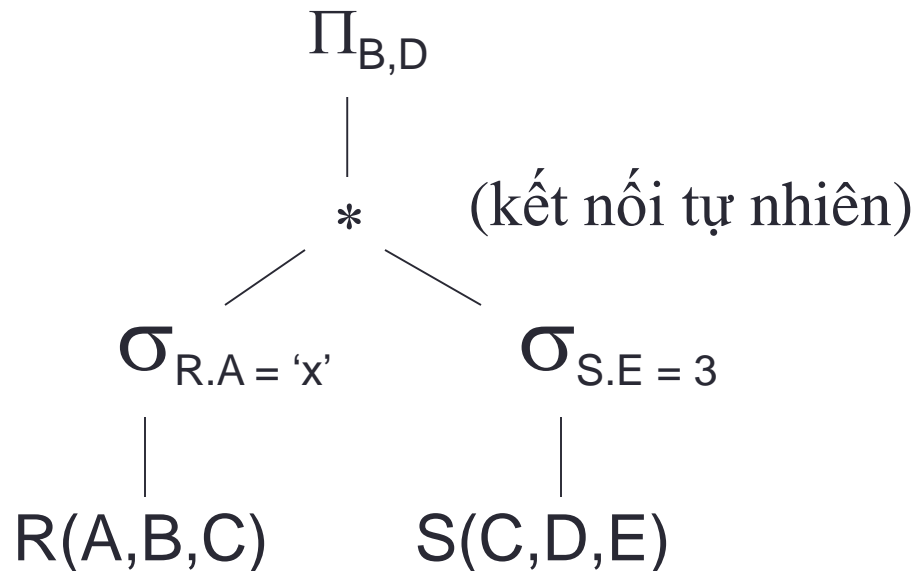
# Mô tả bằng đại số quan hệ

Giải pháp 1:



$$\Pi_{B,D} [\sigma_{R.A='x' \wedge S.E=3 \wedge R.C=S.C} (R \times S)]$$

## Giải pháp 2

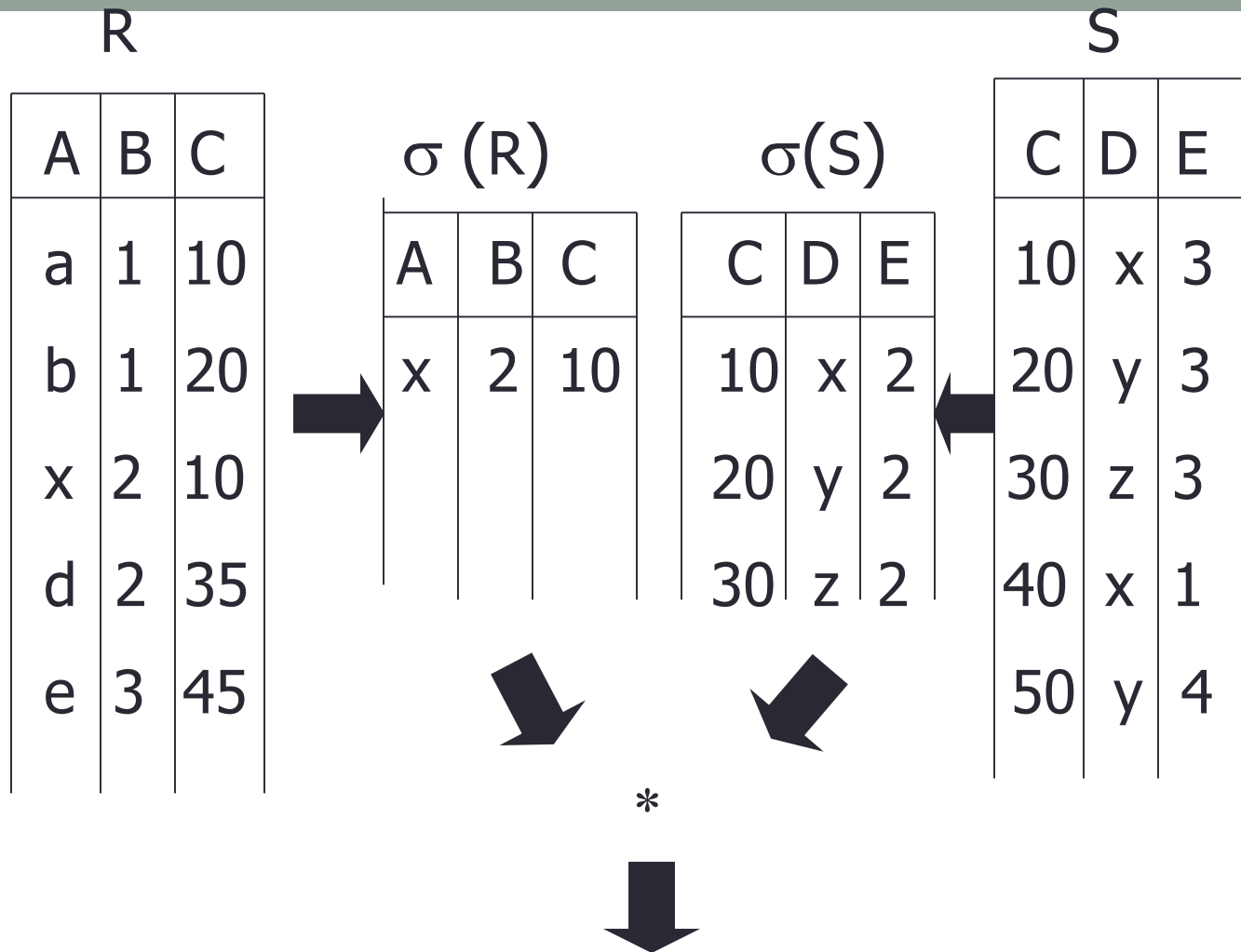


Select B,D

From R,S

Where R.A = 'x' AND S.E = 3 AND R.C=S.C;





Select B,D

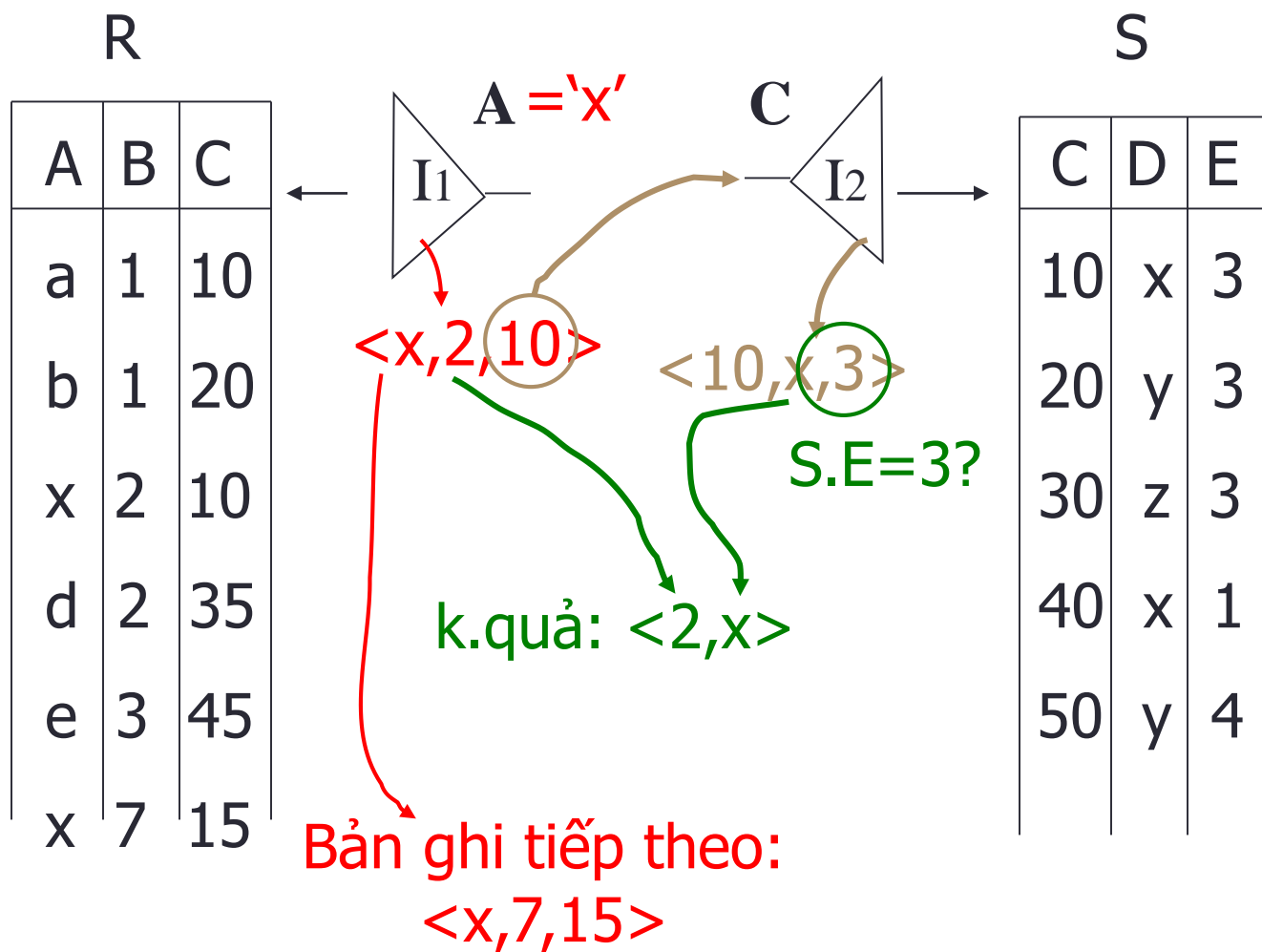
From R,S

Where R.A = 'x' AND S.E = 3 AND R.C=S.C;

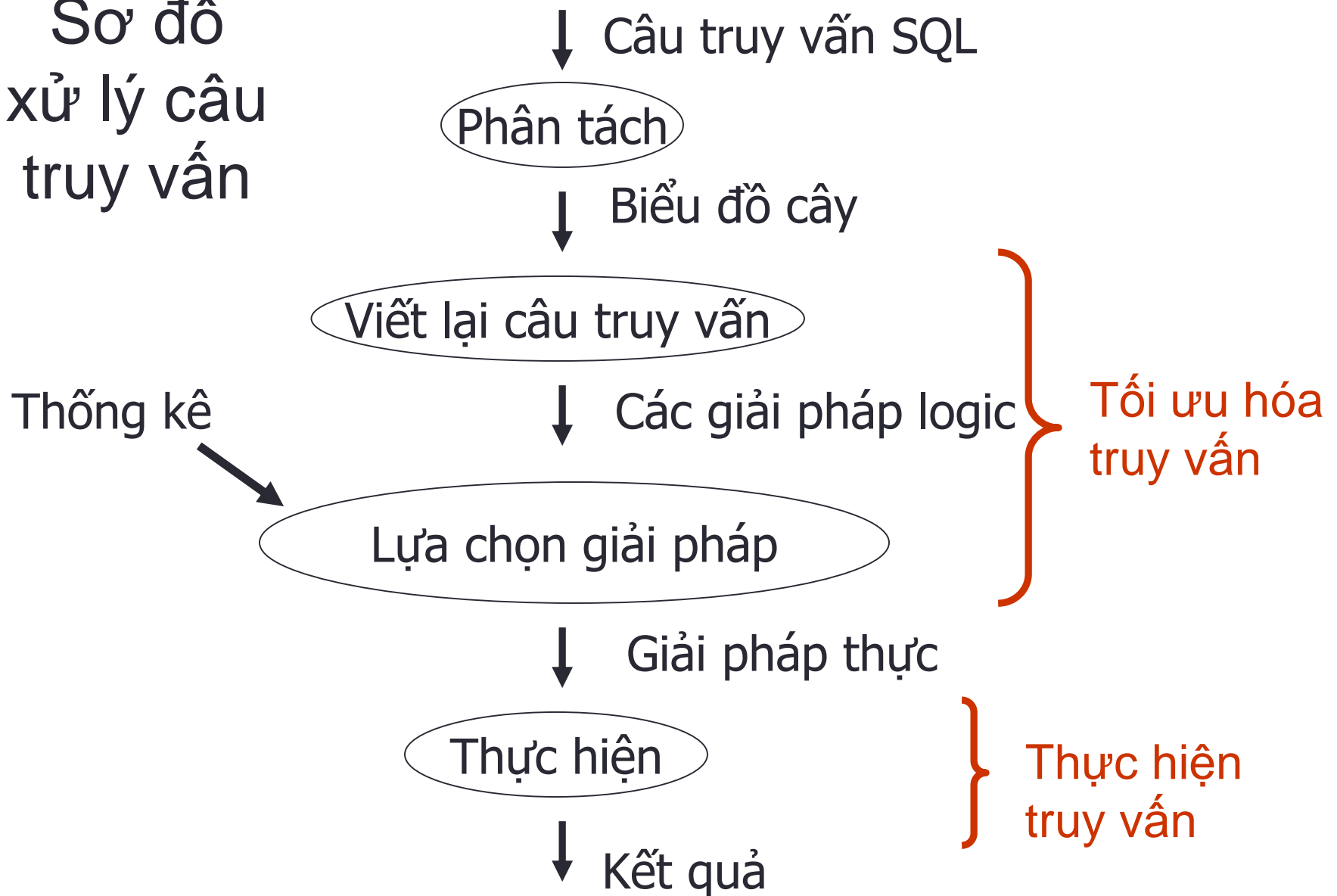
## Giải pháp 3

Sử dụng các thuộc tính R.A và S.C:

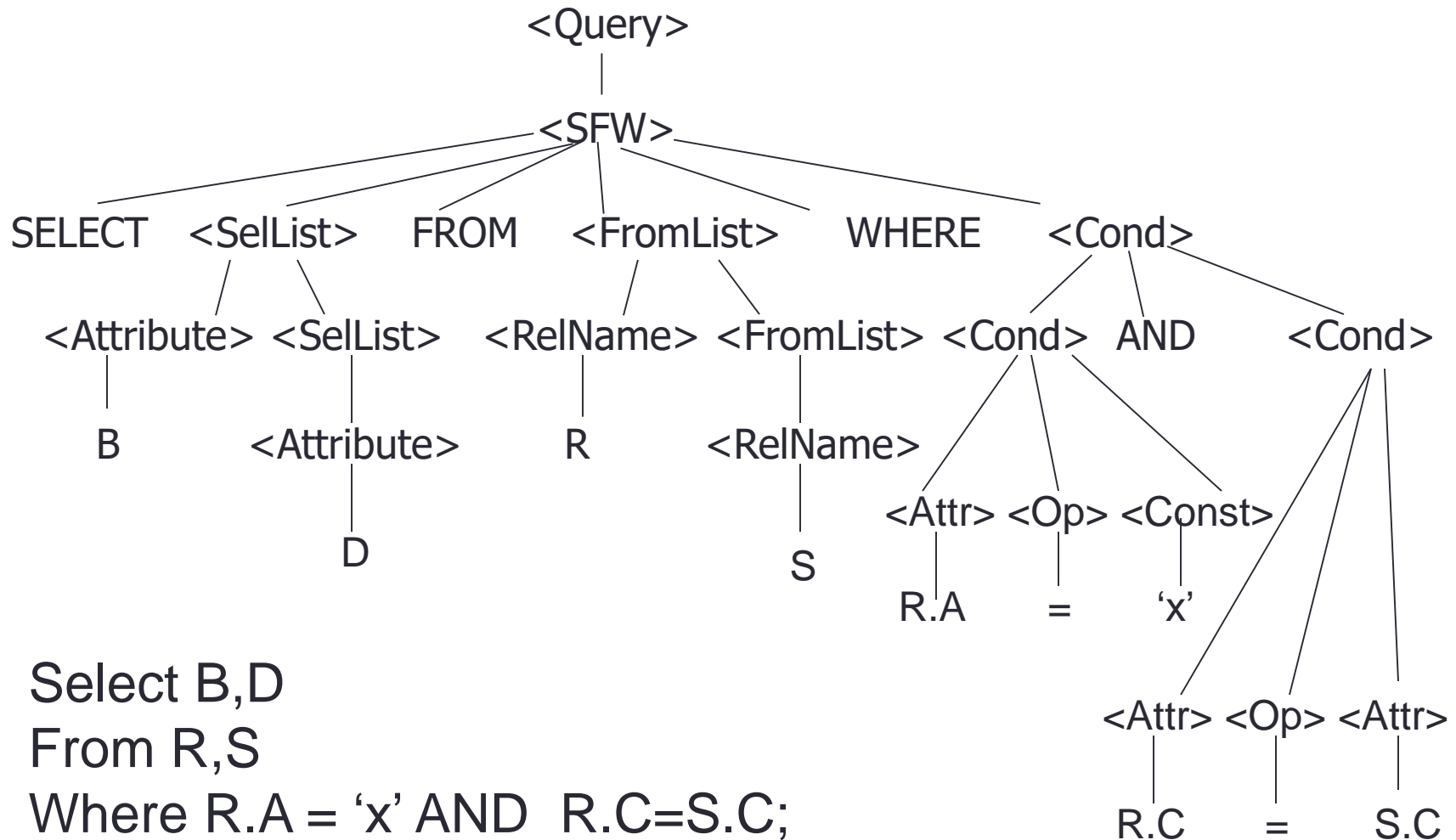
- (1) Dùng R.A để lựa chọn các bản ghi trong R có chứa  $R.A = 'x'$
- (2) Ứng với mỗi bản ghi đã tìm được bằng R.C, sử dụng S.C để tìm các bản ghi trong S có  $S.C = 3$ .
- (3) Loại bỏ các bản ghi có  $S.E \neq 3$
- (4) Kết nối tự nhiên các bản ghi đã tìm được trong R và S, chiếu lên các thuộc tính B, D.

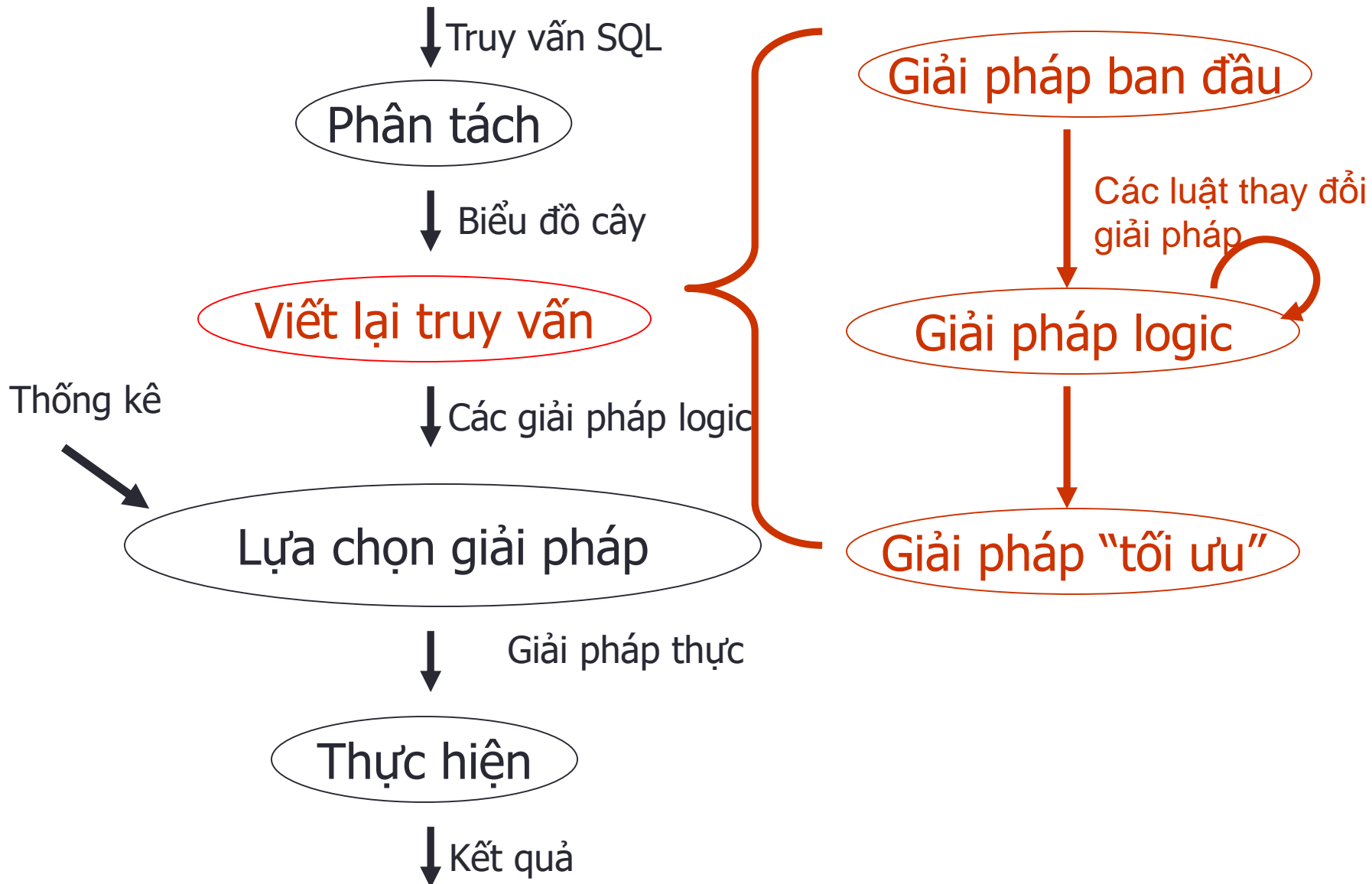


Sơ đồ  
xử lý câu  
truy vấn



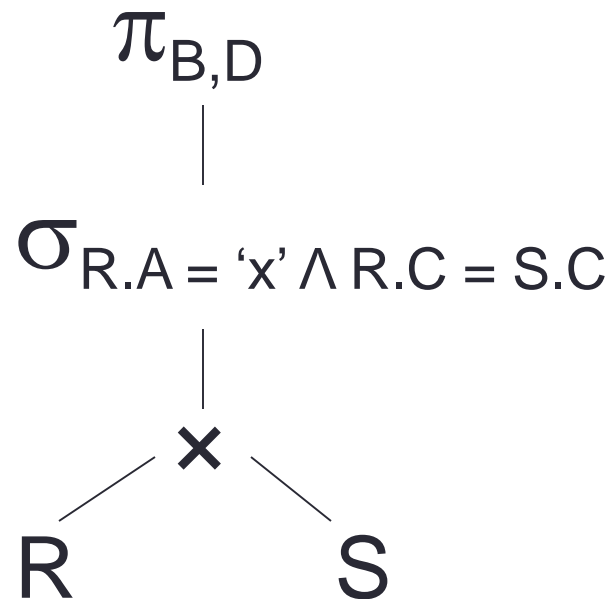
## Ví dụ: Cây phân tách





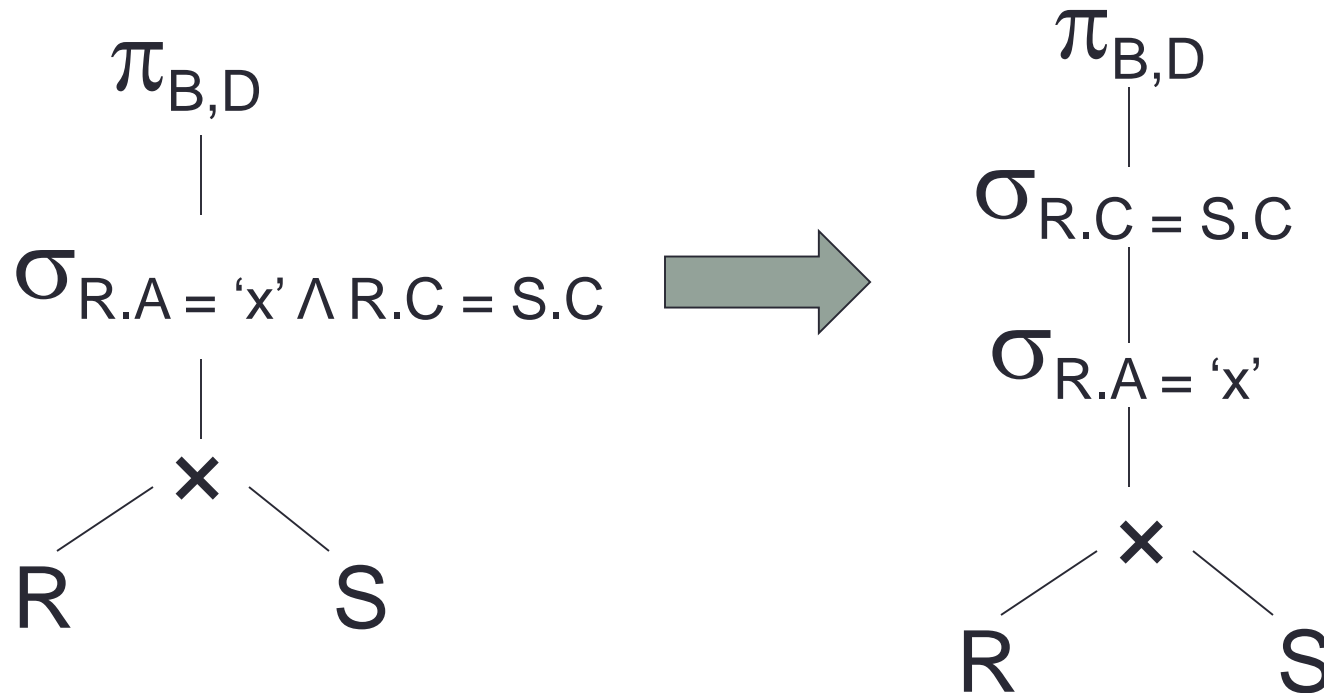
# Giải pháp ban đầu

Select B,D  
From R,S  
Where R.A = 'x'  
AND R.C=S.C;



Đại số quan hệ:  $\Pi_{B,D} [\sigma_{R.A='x' \wedge R.C = S.C} (R \times S)]$

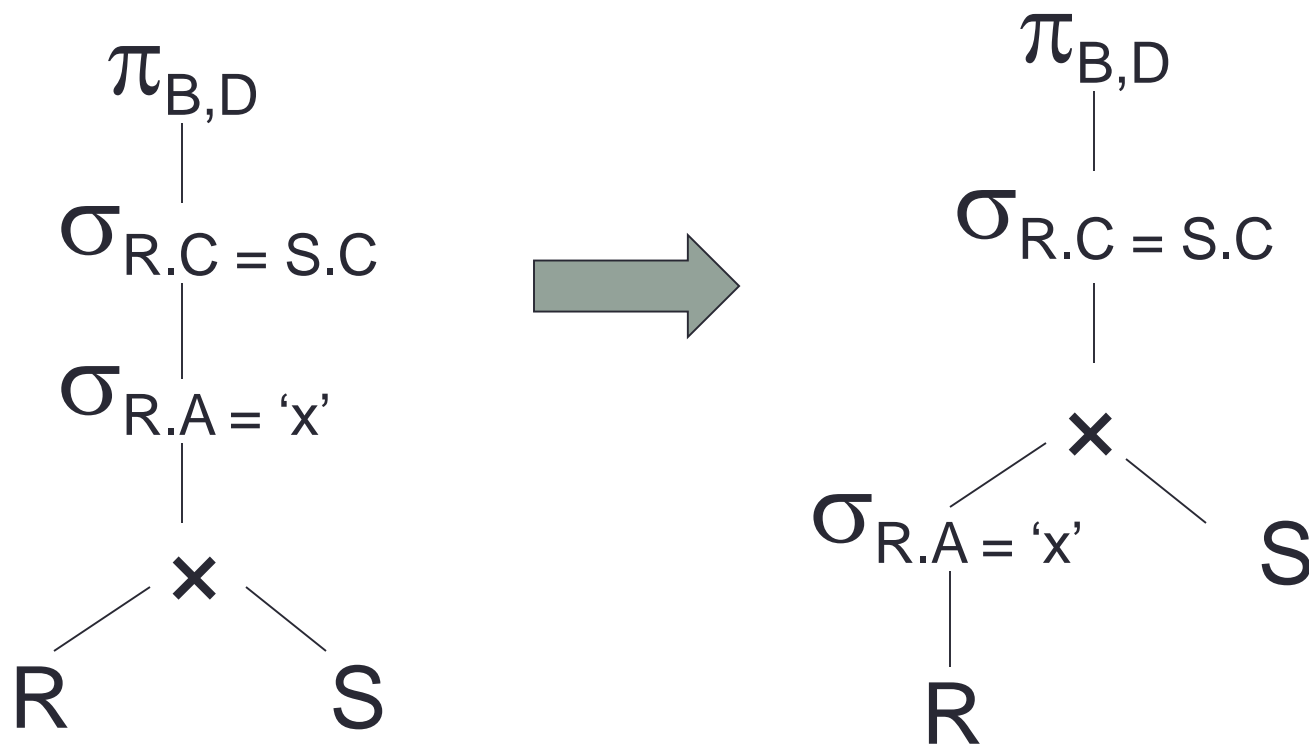
# Luật thay đổi giải pháp 1



$$\Pi_{B,D} [\sigma_{R.C=S.C} [\sigma_{R.A='x'}(R \bowtie S)]]$$

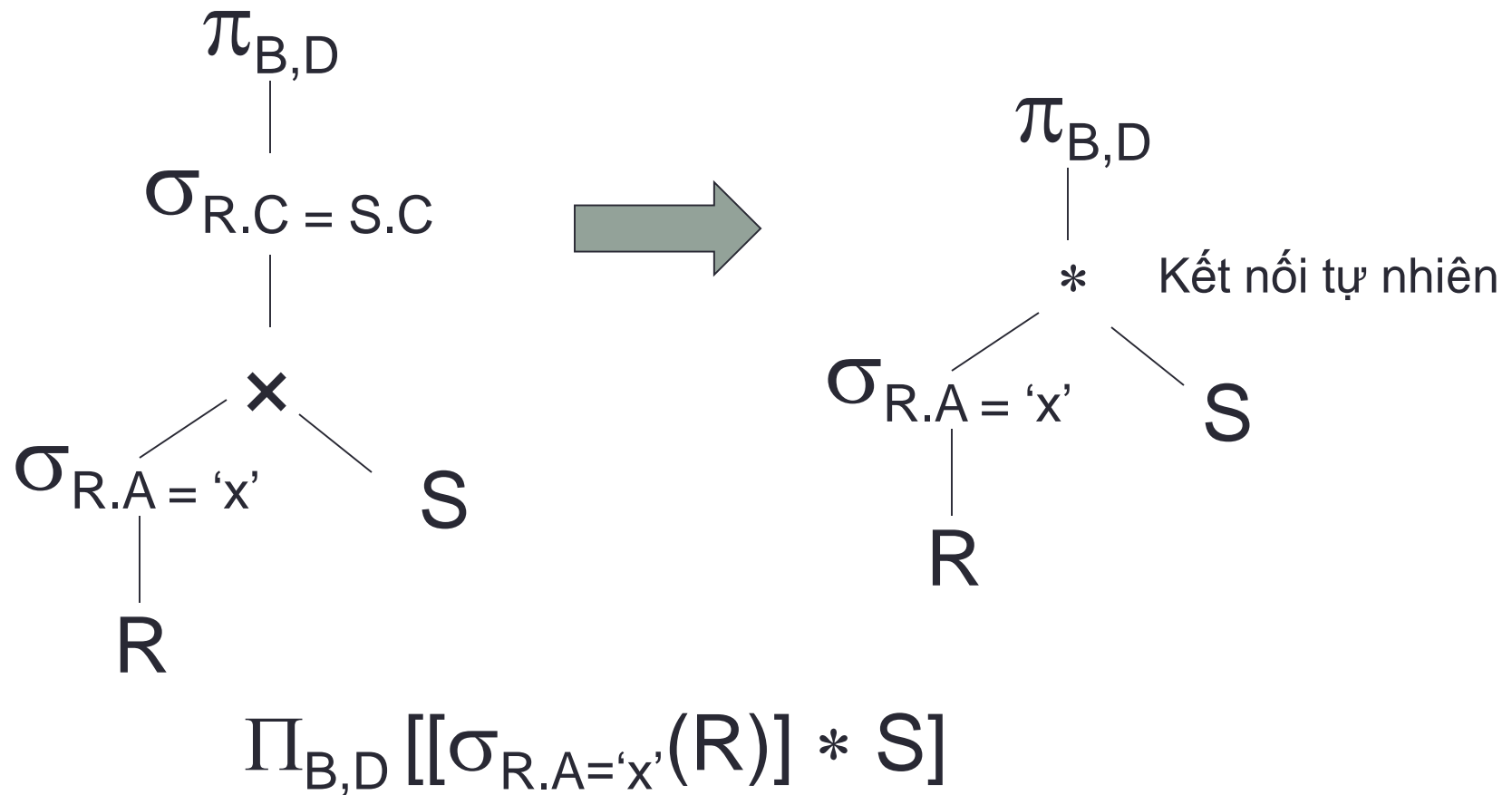


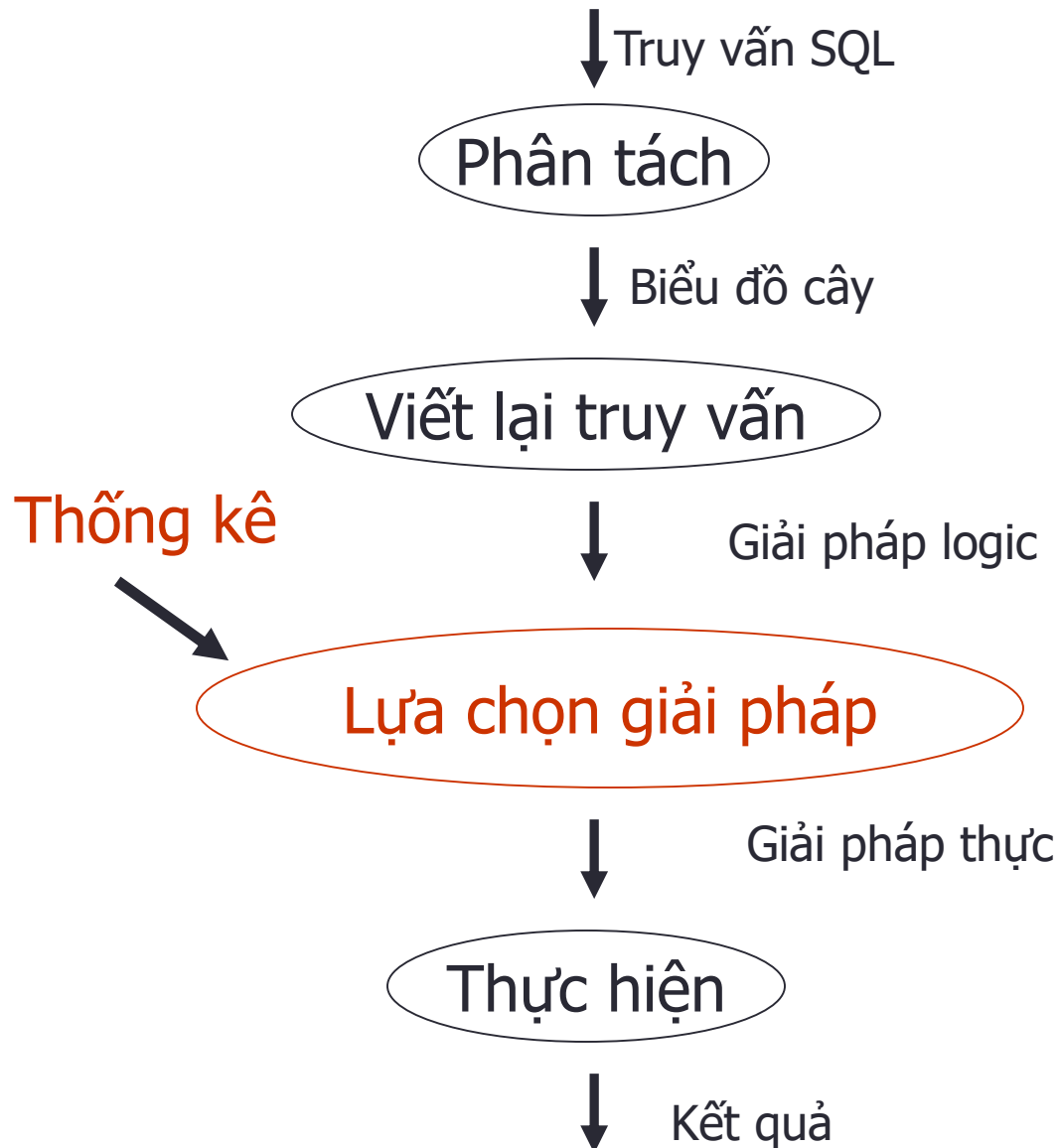
## Luật thay đổi giải pháp 2



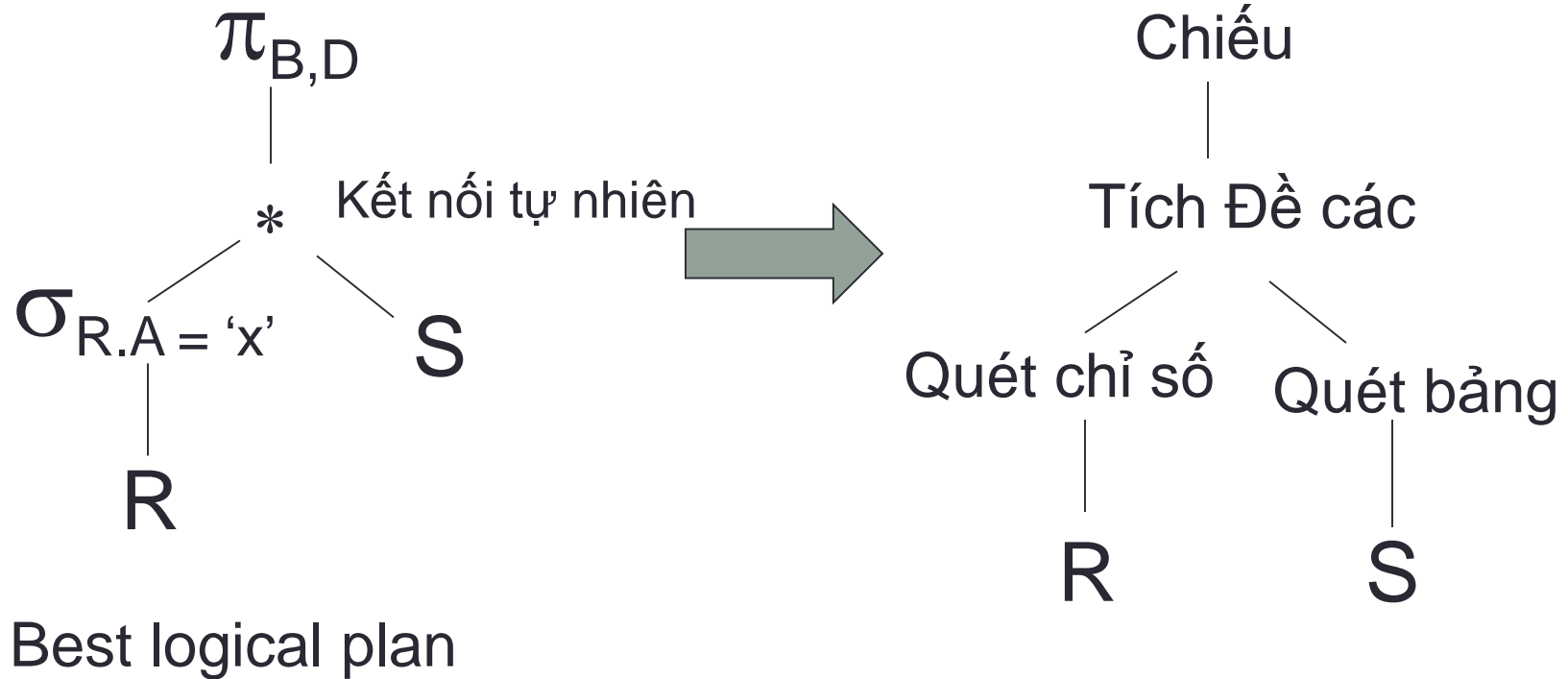
$$\Pi_{B,D} [\sigma_{R.C=S.C} [\sigma_{R.A='x'}(R)] \times S]$$

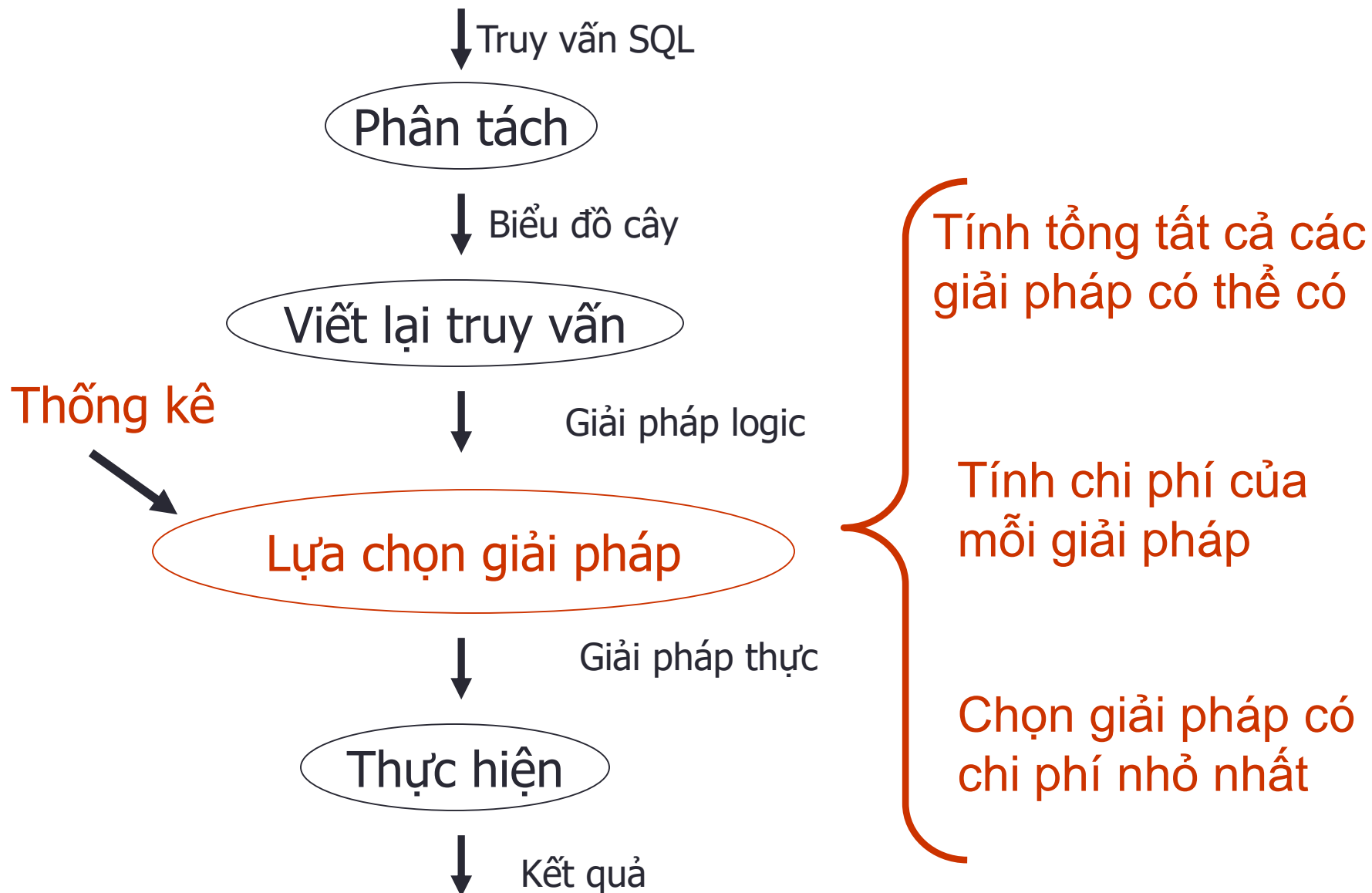
# Luật thay đổi giải pháp 3



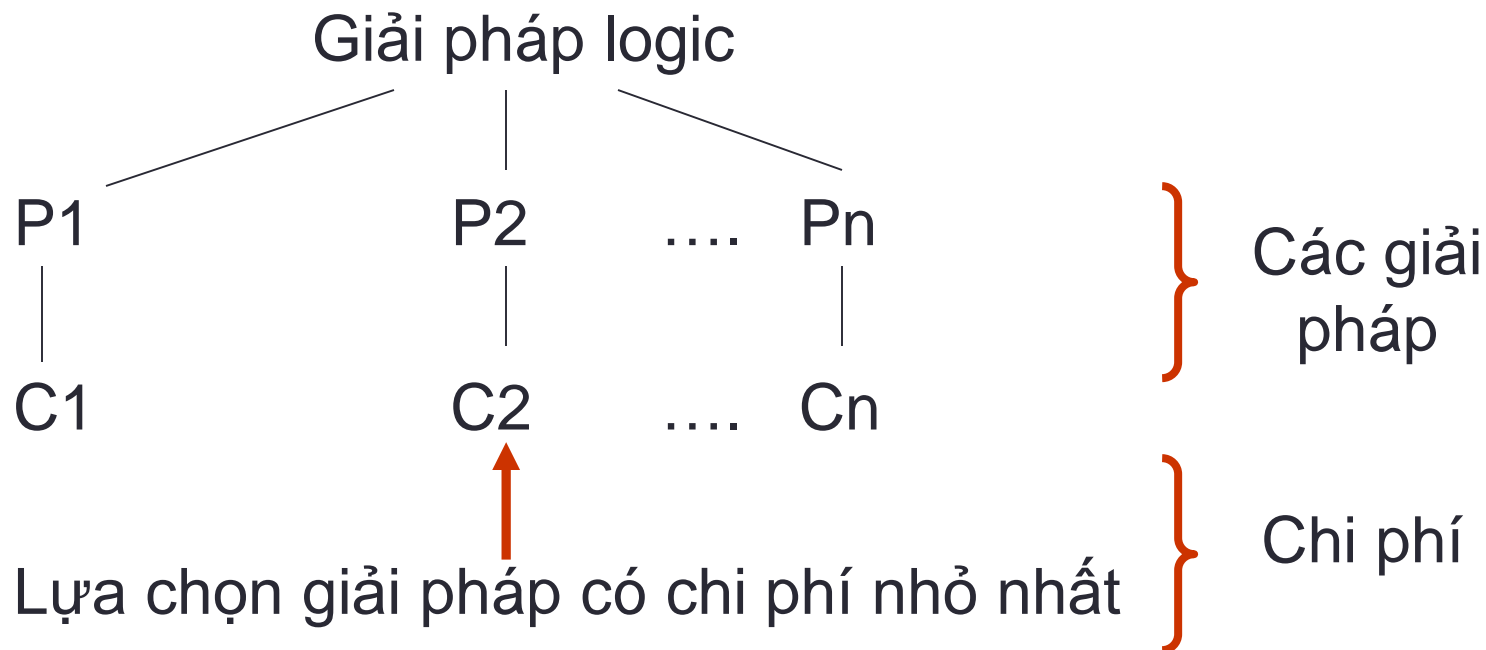


# Chuyển đổi giải pháp

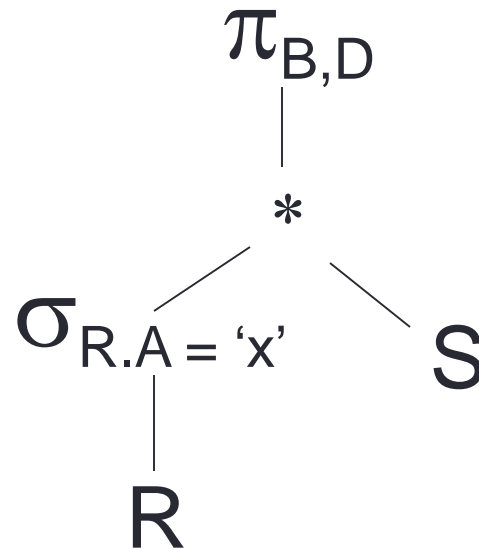




# Lựa chọn giải pháp

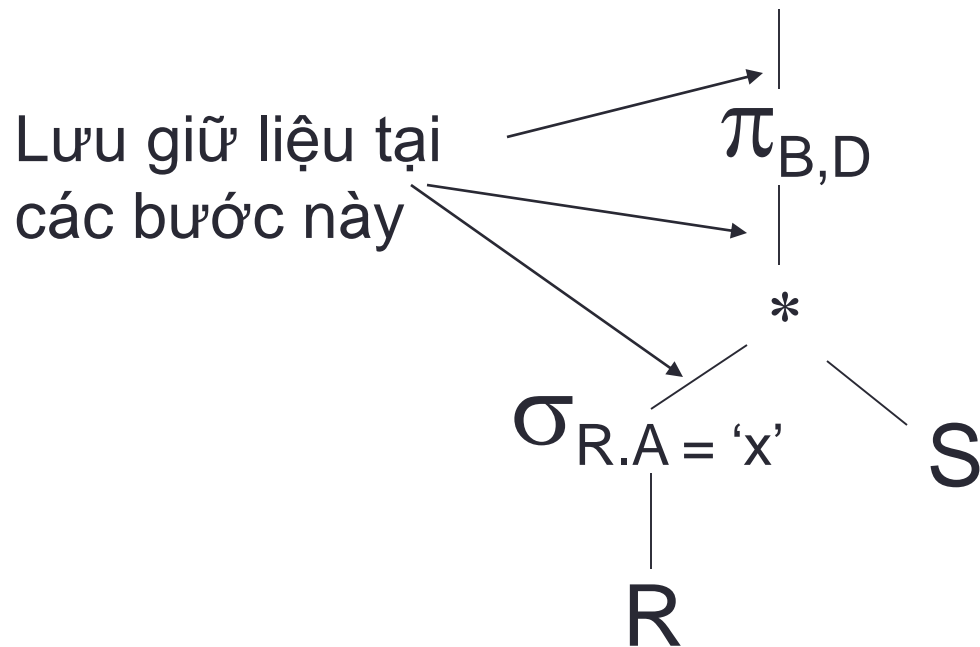


# Thực hiện giải pháp



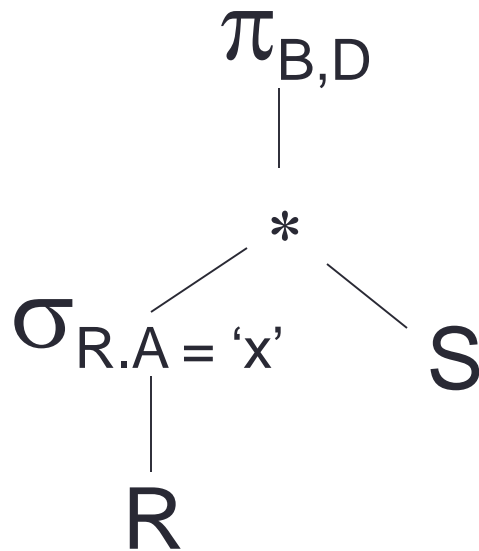
- **Theo chặng:** Thực hiện từng câu lệnh rồi ghi kết quả ra đĩa, các câu lệnh tiếp theo đọc lại kết quả của các câu lệnh trước đó.
- **Theo luồng:** Kết quả của câu lệnh trước là đầu vào trực tiếp cho các câu lệnh tiếp theo

# Theo chặng





# Theo luồng



→ tại mỗi câu lệnh đều có các thủ tục:

- `Open()`
- `GetNext()`
- `Close()`

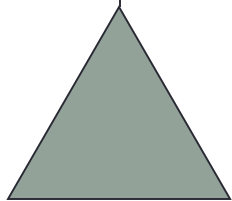
# Ví dụ: thao tác tại bước quét bảng R

```
Open() {  
    /** initialize variables */  
    b = first block of R;  
    t = first tuple in block b;  
}
```

```
Close() {  
    /** nothing to be done */  
}
```

```
GetNext() {  
    IF (t is past last tuple in block b) {  
        set b to next block;  
        IF (there is no next block)  
            /** no more tuples */  
            RETURN EOT;  
        ELSE t = first tuple in b;  
    }  
    /** return current tuple */  
    oldt = t;  
    set t to next tuple in block b;  
    RETURN oldt;  
}
```

# Thao tác thực hiện lệnh Chọn

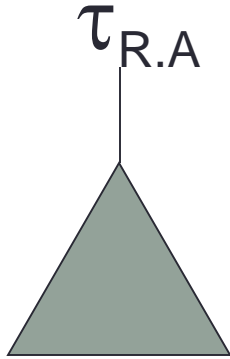
$$\sigma_{R.A = 'x'}$$


```
Open() {
    /** initialize child */
    Child.Open();
}
```

```
Close() {
    /** inform child */
    Child.Close();
}
```

```
GetNext() {
    LOOP:
        t = Child.GetNext();
        IF (t == EOT) {
            /** no more tuples */
            RETURN EOT;
        }
        ELSE IF (t.A == 'x')
            RETURN t;
    ENDLOOP:
}
```

# Thao tác thực hiện lệnh Sắp xếp



```

getNext() {
    IF (more tuples)
        RETURN next tuple in order;
    ELSE RETURN EOT;
}

```

```

Open() {
    /** Bulk of the work is here */
    Child.Open();
    Read all tuples from Child
    and sort them
}

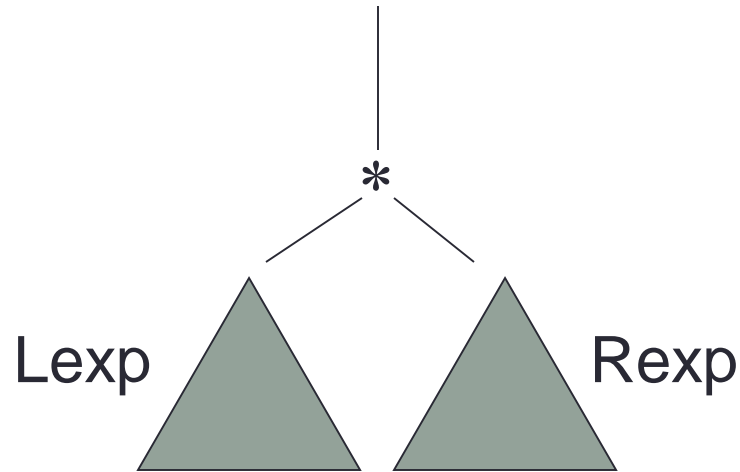
```

```

Close() {
    /** inform child */
    Child.Close();
}

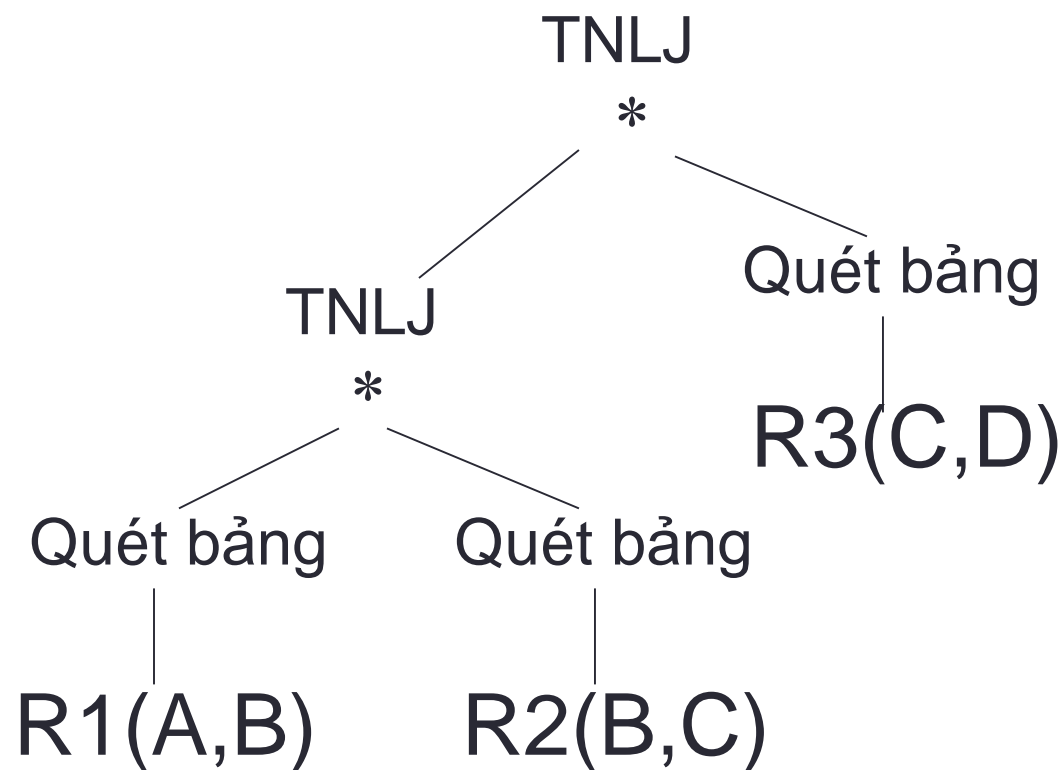
```

# Thao tác thực hiện lệnh Kết nối



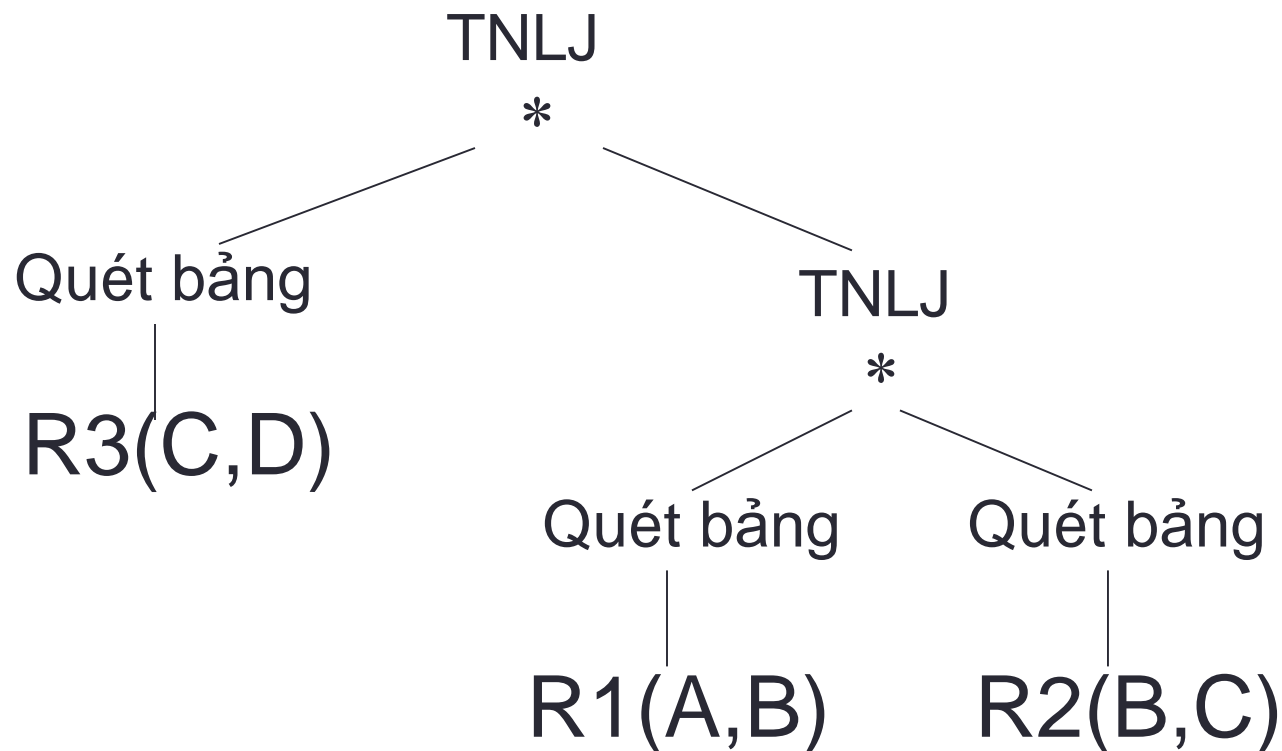
- TNLJ (conceptually)
  - for each  $r \in \text{Lexp}$  do
    - for each  $s \in \text{Rexp}$  do
      - if  $\text{Lexp.C} = \text{Rexp.C}$ , output  $r, s$

# Ví dụ: Cây phát triển trái



Thứ tự thực hiện hàm getNext() là gì?

# Ví dụ: Cây phát triển phải



# Tính toán chi phí cho các giải pháp

- Các chi phí cần tính đến:
  - Thời gian hoàn thành câu lệnh
  - Lượng thông tin vào/ra
  - Số lần gọi hàm getNext()
- Cân nhắc lựa chọn giữa: Đơn giản về tính toán Vs. Độ tối ưu (thời gian, tài nguyên) trong quá trình thực hiện.



# Tại sao phải viết lại câu truy vấn?

- **Giảm kích thước câu lệnh**
  - Loại bỏ các điều kiện/câu lệnh dư thừa
  - Áp dụng các luật xử lý nhằm tăng hiệu quả câu lệnh
- **Tiền xử lý**
  - Chuyển đổi câu truy vấn về dạng dễ xử lý nhất.

➔ Giảm thiểu thời gian xử lý của hệ thống mà không làm giảm chất lượng của kết quả đầu ra

# Một vài quy tắc viết lại câu lệnh

- Chuyển đổi giữa các giải pháp logic
- Tìm các phép toán đại số quan hệ tương đương
- Giảm các kết quả trung gian
- Thực hiện sớm các phép chiếu
- Tránh sử dụng phép tích Đề các

## Một số phép toán đại số quan hệ tương đương

$$R * S = S * R \quad \text{Giao hoán}$$

$$(R * S) * T = R * (S * T) \quad \text{Kết hợp}$$

Tương tự cho các phép toán: Tích Đề các, Hợp, Giao

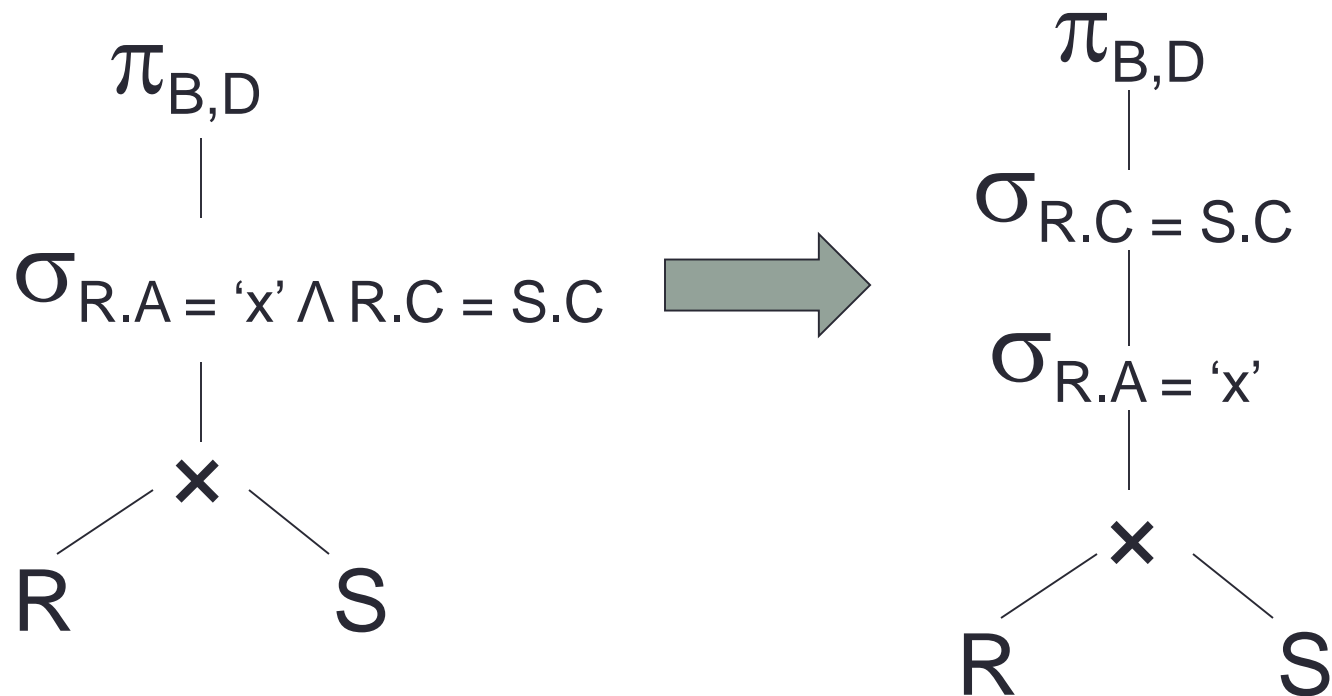
$$R \times S = S \times R$$

$$(R \times S) \times T = R \times (S \times T)$$

$$R \cup S = S \cup R$$

$$R \cup (S \cup T) = (R \cup S) \cup T$$

## Luật viết lại câu lệnh 1



$$\Pi_{B,D} [\sigma_{R.C=S.C} [\sigma_{R.A='c'}(R \times S)]]$$

## Lưu ý: phép Chiếu

Gọi:  $X$  = tập các thuộc tính

$Y$  = tập các thuộc tính

$$XY = X \cup Y$$

$$\pi_{xy}(R) = \pi_x[\pi_y(R)]$$


## Lưu ý: phép chọn kết hợp với phép kết nối tự nhiên

Gọi  $p$  = tập các thuộc tính của  $R$

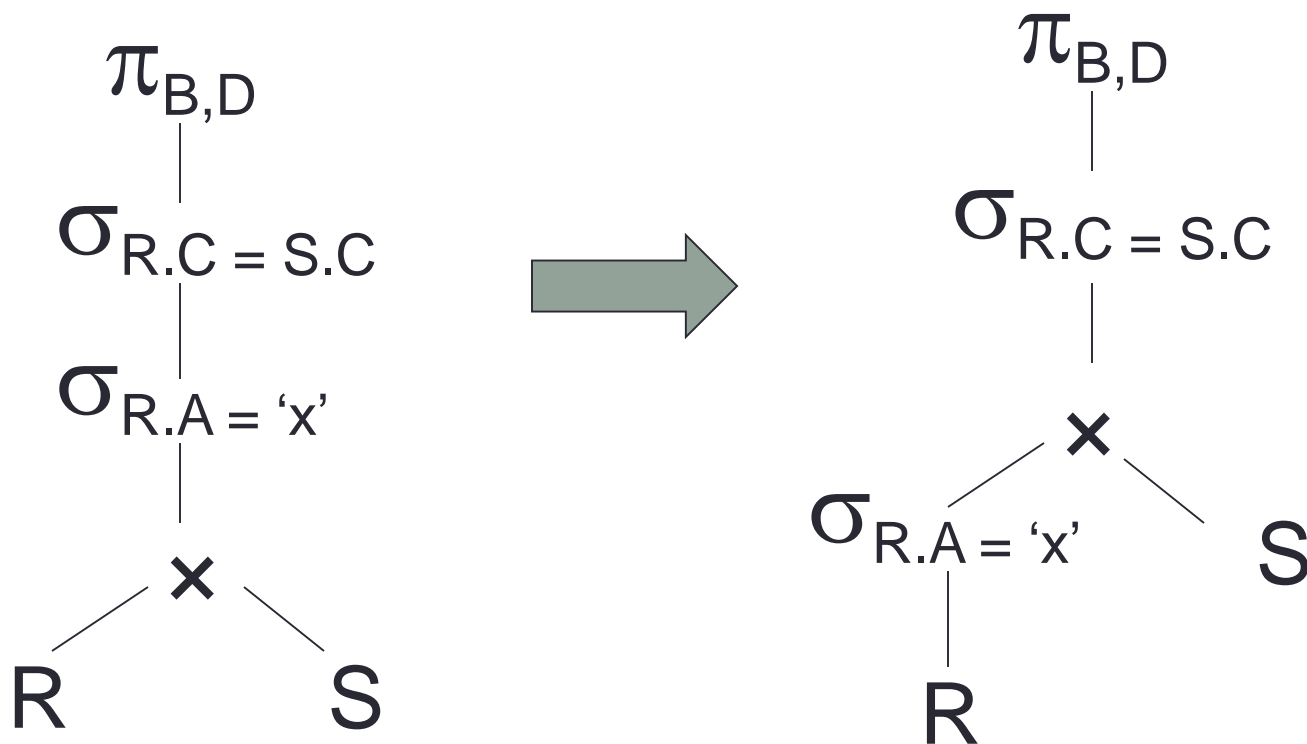
$q$  = tập các thuộc tính của  $S$

$m$  = tập các thuộc tính của cả  $R$  và  $S$

$$\sigma_p (R * S) = [\sigma_p (R)] * S$$

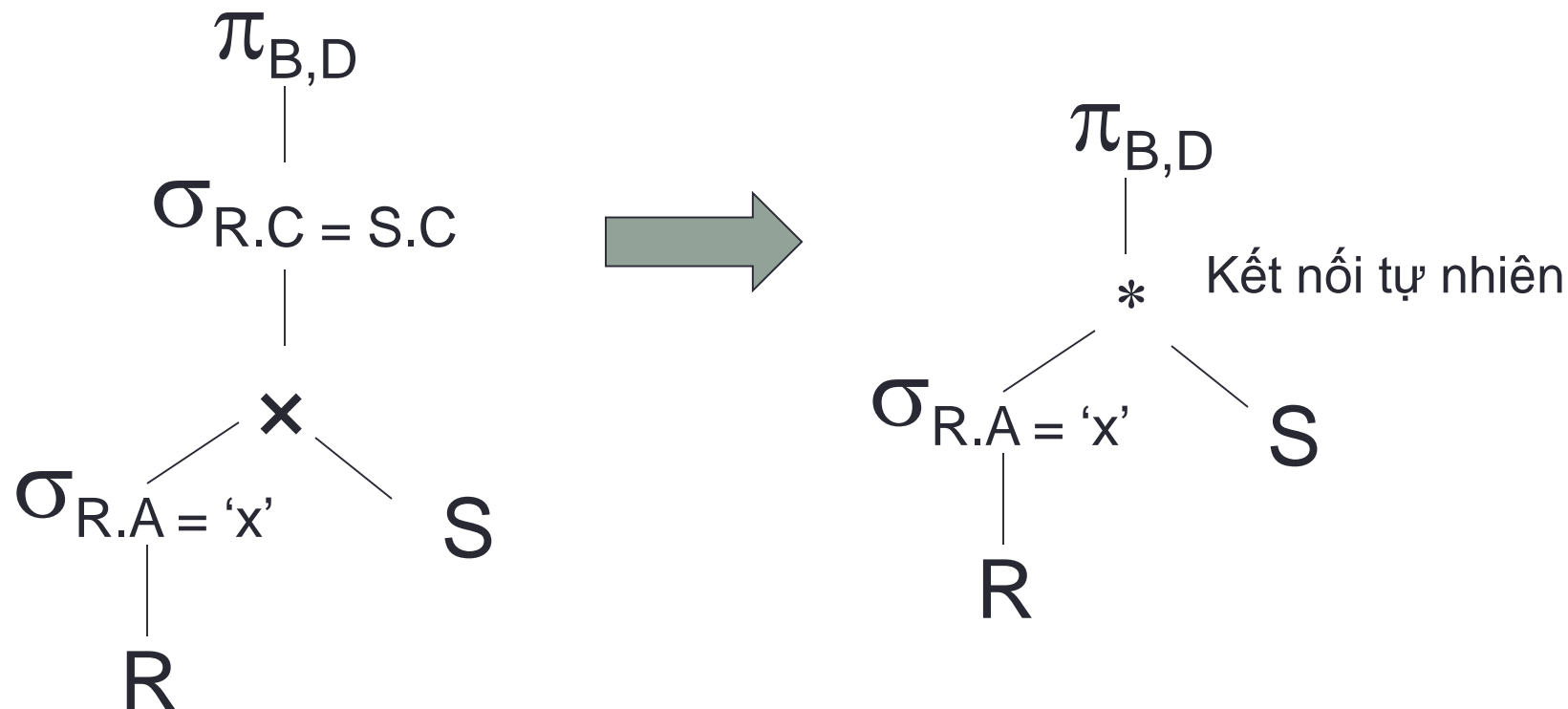
$$\sigma_q (R * S) = R * [\sigma_q (S)]$$

## Luật viết lại câu lệnh 2



$$\Pi_{B,D} [\sigma_{R.C=S.C} [\sigma_{R.A='x'}(R)] \times S]$$

## Luật viết lại câu lệnh 3



$$\Pi_{B,D} [[\sigma_{R.A="c"}(R)] * S]$$



Lưu ý: phép Chọn kết hợp với phép Kết nối tự nhiên

$$\sigma_{p \wedge q} (R * S) = \sigma_p (R) * \sigma_q (S)$$

$$\sigma_{p \wedge q \wedge m} (R * S) = \sigma_m [\sigma_p (R) * \sigma_q (S)]$$

$$\sigma_{p \vee q} (R * S) = [\sigma_p (R) * S] \cup [R * \sigma_q (S)]$$

## Một số chuyển đổi

- $\sigma_{p1 \wedge p2} (R) \rightarrow \sigma_{p1} [\sigma_{p2} (R)]$
- $\sigma_p (R * S) \rightarrow [\sigma_p (R)] * S$
- $R * S \rightarrow S * R$
- $\pi_x [\sigma_p (R)] \rightarrow \pi_x \{ \sigma_p [\pi_{xp} (R)] \}$

Mẹo: Thực hiện phép Chiếu càng sớm càng tốt

Ví dụ:  $R(A,B,C,D,E)$

$P: (A=2) \wedge (B=\text{"long"})$

$$\pi_E \{ \sigma_P (R) \} \quad \text{vs.} \quad \pi_E \{ \sigma_P [\pi_{ABE}(R)] \}$$

## Chốt lại:

- Không có phương pháp chuyển đổi tối ưu cho mọi câu truy vấn.
- Một vài phương pháp chuyển đổi là tốt như:
  - Giảm số kết quả trung gian
  - Tránh dùng phép Tích Đề các
  - Thực hiện Kết nối tự nhiên trong các câu truy vấn phụ.

## Ví dụ: tránh dùng Tích Đề các

```
Select B,D  
From R,S,T,U  
Where R.A = S.B  
      AND R.C=T.C  
      AND R.D = U.D;
```

- Biểu đồ cây dạng nào phù hợp trong trường hợp này?
- Nếu không thể tránh dùng Tích Đề các, hãy cố dùng nó cuối cùng.

# Một vài quy tắc khác để viết lại câu lệnh

- Chuyển đổi giữa các giải pháp logic
- Tìm các phép toán đại số quan hệ tương đương
- Giảm các kết quả trung gian
- Thực hiện sớm các phép chiếu
- Tránh sử dụng phép tích Đề các
- Sử dụng biểu đồ (cây) phát triển trái
- Thực hiện kết nối tự nhiên trong các câu truy vấn phụ
- Sử dụng các ràng buộc giới hạn