

## **Chương 3**

# **Các mức trong suốt phân tán**



# Nội dung

---

- ❖ Kiến trúc tham khảo của CSDL phân tán.
- ❖ Các loại phân mảnh dữ liệu.
- ❖ Các điều kiện đúng dẫn để phân mảnh dữ liệu.
- ❖ Phân mảnh ngang chính.
- ❖ Phân mảnh ngang dẫn xuất.
- ❖ Phân mảnh dọc.
- ❖ Phân mảnh hỗn hợp.
- ❖ Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc.

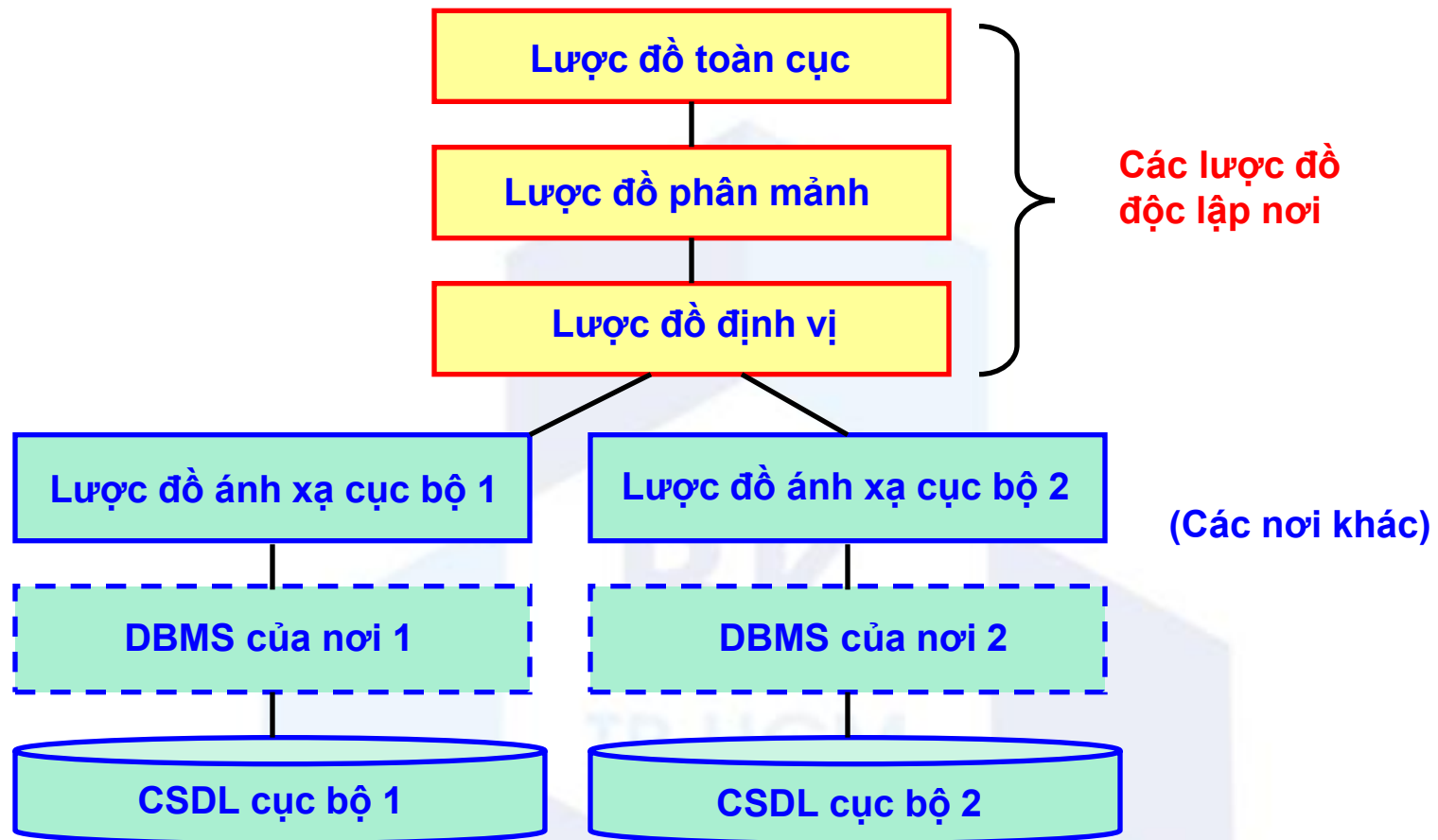
# Nội dung

---

- ❖ Tính trong suốt phân tán dùng cho ứng dụng cập nhật.
- ❖ Các tác vụ cơ bản truy xuất CSDL phân tán.



# Kiến trúc tham khảo của CSDL phân tán



Hình 3.1. Kiến trúc tham khảo dùng cho các CSDL phân tán.

# Kiến trúc tham khảo của CSDL phân tán

## ❖ Trong suốt phân mảnh (*fragmentation transparency*)

- ▶ Không nhìn thấy các mảnh.
- ▶ Nhìn thấy các quan hệ toàn cục (*global relation*).
- ▶ Nhìn thấy lược đồ toàn cục (*global schema*).

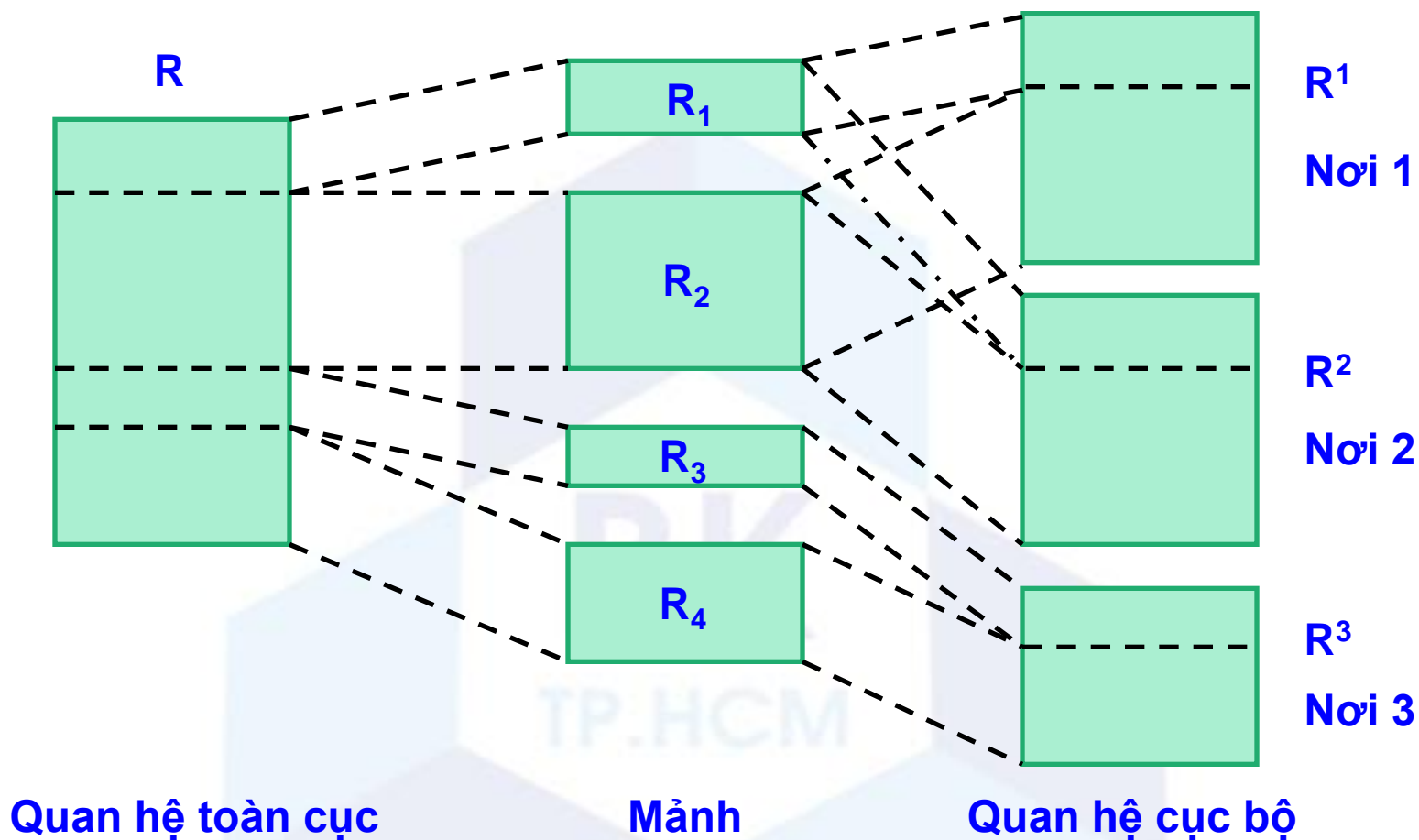
## ❖ Trong suốt vị trí (*location transparency*)

- ▶ Không nhìn thấy các quan hệ cục bộ.
- ▶ Nhìn thấy các mảnh (*fragment*).
- ▶ Nhìn thấy lược đồ phân mảnh (*fragmentation schema*).

# Kiến trúc tham khảo của CSDL phân tán

- ❖ **Trong suốt nhân bản (*replication transparency*)**
  - ▶ Nhìn thấy các mảnh.
  - ▶ Không nhìn thấy sự nhân bản của các mảnh.
- ❖ **Trong suốt ánh xạ cục bộ (*local mapping transparency*)**
  - ▶ Nhìn thấy các quan hệ cục bộ (*local relation*).
  - ▶ Không nhìn thấy CSDL vật lý.
  - ▶ Nhìn thấy lược đồ định vị (*allocation schema*).
- ❖ **Không trong suốt (*no transparency*)**
  - ▶ Lược đồ ánh xạ cục bộ (*local mapping schema*)

# Kiến trúc tham khảo của CSDL phân tán



Hình 3.2. Các mảnh và các hình ảnh vật lý của một quan hệ toàn cục.

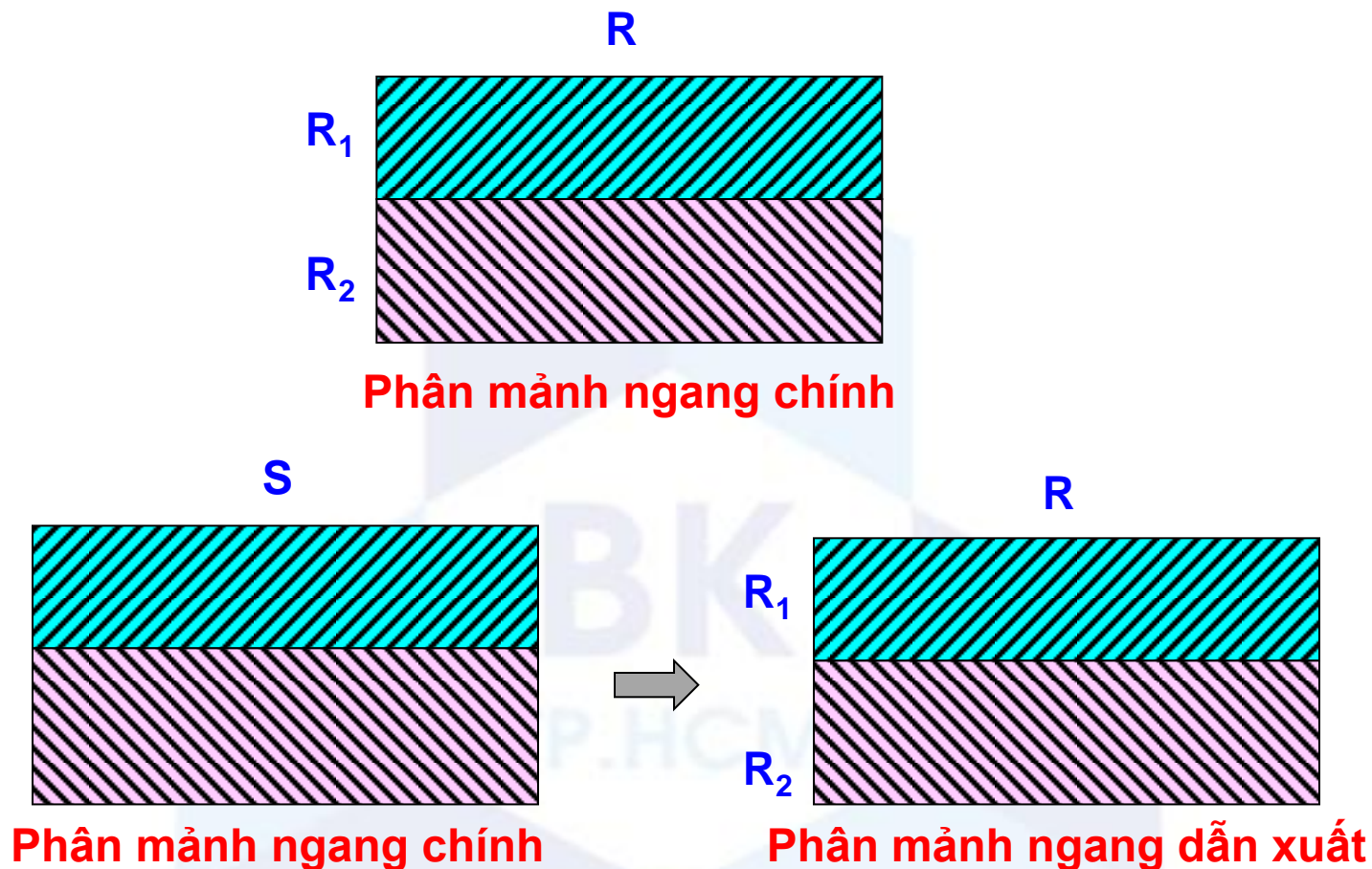
# Các loại phân mảnh dữ liệu

- ❖ **Phân mảnh ngang (*horizontal fragmentation*)**
  - ▶ **Phân mảnh ngang chính (*primary horizontal fragmentation*)**
  - ▶ **Phân mảnh ngang dẫn xuất (*derived horizontal fragmentation*)**





# Các loại phân mảnh dữ liệu



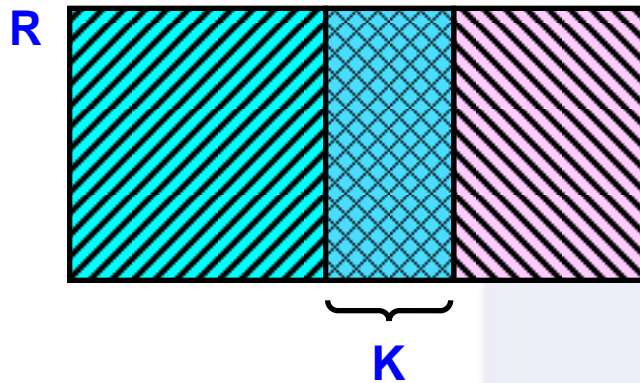
Hình 3.3. Các loại phân mảnh ngang.

# Các loại phân mảnh dữ liệu

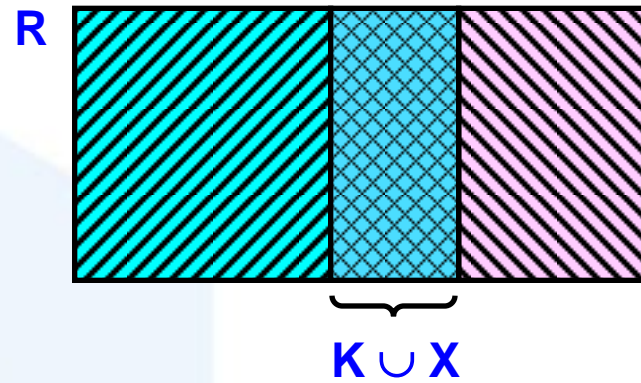
## ❖ Phân mảnh dọc (*vertical fragmentation*)

- ▶ Phân mảnh dọc gom tụ (*vertical clustering fragmentation*)
  - Phân mảnh dư thừa (*redundant fragmentation*)
  - Phân mảnh không dư thừa (*non-redundant fragmentation*)
- ▶ Phân mảnh dọc tách biệt (*vertical partitioning fragmentation*)

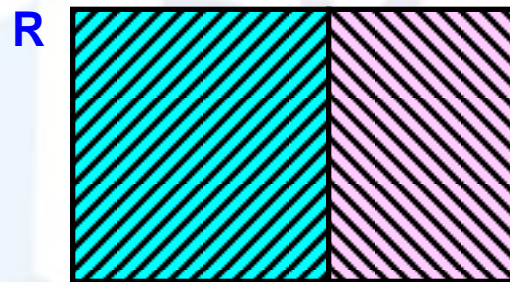
# Các loại phân mảnh dữ liệu



Phân mảnh gom tụ không dư thừa



Phân mảnh gom tụ có dư thừa



Phân mảnh dọc tách biệt

Hình 3.4. Các loại phân mảnh dọc.

# Các loại phân mảnh dữ liệu

## ❖ Phân mảnh hỗn hợp (*mixed fragmentation*)

- ▶ Kết hợp giữa phân mảnh ngang và phân mảnh dọc.



Hình 3.5. Phân mảnh hỗn hợp.

# Các điều kiện đúng đắn

- ❖ Quan hệ  $R$  được phân rã thành các mảnh  $R_1, R_2, \dots, R_n$
- ❖ **Điều kiện đầy đủ (*completeness condition*)**
  - ▶ Mọi mục dữ liệu trong  $R$  phải có trong một hoặc nhiều mảnh  $R_i$
  - ▶ **Phân mảnh ngang:**  
$$\forall u \in R, \exists i \in [1, n]: u \in R_i$$
  - ▶ **Phân mảnh dọc:**  
$$\forall A \in Attr(R), \exists i \in [1, n]: A \in Attr(R_i)$$

# Các điều kiện đúng đắn

## ❖ Điều kiện tái tạo (*reconstruction condition*)

- ▶ Luôn luôn có thể xác định một phép toán quan hệ  $\nabla$  sao cho:

$$R = \nabla R_i \quad \forall R_i \in F_R \text{ với } F_R = \{R_1, R_2, \dots, R_n\}$$

- ▶ Phân mảnh ngang:

$$R = R_1 \cup R_2 \cup \dots \cup R_n$$

- ▶ Phân mảnh dọc:

$$R = R_1 \triangleright \triangleleft R_2 \triangleright \triangleleft \dots \triangleright \triangleleft R_n$$

# Các điều kiện đúng đắn

## ❖ Điều kiện tách biệt (*disjointness condition*)

- ▶ Nếu mục dữ liệu  $d_i$  có trong  $R_i$  thì nó không có trong bất kỳ mảnh  $R_k$  khác ( $k \neq i$ ).

- ▶ **Phân mảnh ngang:**

$$\forall i \neq k \text{ và } i, k \in [1, n]: R_i \cap R_k = \emptyset$$

hoặc

$$\forall u \in R_i \text{ và } \forall i \neq k \text{ và } i, k \in [1, n]: u \notin R_k$$

# Phân mảnh ngang chính

- ❖ **Phân mảnh ngang chính** (*primary horizontal fragmentation*) của một quan hệ toàn cục là sự phân chia các bộ của quan hệ này thành các tập hợp con dựa vào các thuộc tính của quan hệ này, mỗi tập hợp con được gọi là **mảnh ngang chính** (*primary horizontal fragment*).
- ❖ Mỗi mảnh ngang được tạo bởi một phép chọn trên quan hệ toàn cục.
- ❖ Vị từ được dùng làm điều kiện chọn của phép chọn để tạo ra một mảnh ngang được gọi là **vị từ định tính** (*qualification*) của mảnh ngang đó.



# Phân mảnh ngang chính

## ❖ Ví dụ

### ▶ Quan hệ toàn cục:

sinhviên (mãsv, họ tên, khoa)

### ▶ Các mảnh ngang:

$\text{sinhviên}_1 = \sigma_{\text{khoa} = \text{'CNTT'}} \text{sinhviên}$

$\text{sinhviên}_2 = \sigma_{\text{khoa} = \text{'DIEN'}} \text{sinhviên}$

### ▶ Các vị từ định tính:

$q_1: \text{khoa} = \text{'CNTT'}$

$q_2: \text{khoa} = \text{'DIEN'}$

# Phân mảnh ngang chính

## ❖ Ví dụ

### ► Xét các điều kiện đúng đắn

- Điều kiện đầy đủ: được thỏa mãn nếu 'CNTT' và 'DIEN' chỉ là các giá trị có thể có của thuộc tính *khoa*.
- Điều kiện tái tạo:  
$$\text{sinhviên} = \text{sinhviên}_1 \cup \text{sinhviên}_2$$
- Điều kiện tách biệt: được thỏa mãn vì *mãsv* là khóa của *sinhviên*.

# Phân mảnh ngang chính

- ❖ Để thỏa mãn điều kiện đầy đủ thì tập hợp các vị từ định tính của tất cả các mảnh ngang phải là đầy đủ, ít nhất là đối với tập hợp các giá trị cho phép.
- ❖ Điều kiện tái tạo luôn luôn được thỏa mãn bằng cách sử dụng phép hợp.
- ❖ Để thỏa mãn điều kiện tách biệt thì các vị từ định tính phải loại trừ nhau.

# Phân mảnh ngang dẫn xuất

- ❖ **Phân mảnh ngang dẫn xuất** (*derived horizontal fragmentation*) của một quan hệ toàn cục là sự phân chia các bộ của quan hệ này thành các tập hợp con dựa vào phân mảnh ngang của một quan hệ khác (được gọi là **quan hệ chủ**), mỗi tập hợp con được gọi là **mảnh ngang dẫn xuất** (*derived horizontal fragment*).
- ❖ Vị từ định tính của mảnh ngang dẫn xuất bao gồm **điều kiện kết** và **vị từ định tính của mảnh ngang chủ** tương ứng.

# Phân mảnh ngang dẫn xuất

## ❖ Ví dụ

### ► Quan hệ toàn cục:

điểmthi (mãsv, mãmh, mấbm, điểm)

### ► Các mảnh ngang dẫn xuất:

$\text{điểmthi}_1 = \text{điểmthi} \triangleright \text{mãsv} = \text{mãsv} \text{ sinhviên}_1$

$\text{điểmthi}_2 = \text{điểmthi} \triangleright \text{mãsv} = \text{mãsv} \text{ sinhviên}_2$

### ► Các vị từ định tính:

$q_1: \text{sinhviên.mãsv} = \text{điểmthi.mãsv} \text{ AND}$   
 $\text{sinhviên.khoa} = \text{'CNTT'}$

$q_2: \text{sinhviên.mãsv} = \text{điểmthi.mãsv} \text{ AND}$   
 $\text{sinhviên.khoa} = \text{'DIEN'}$

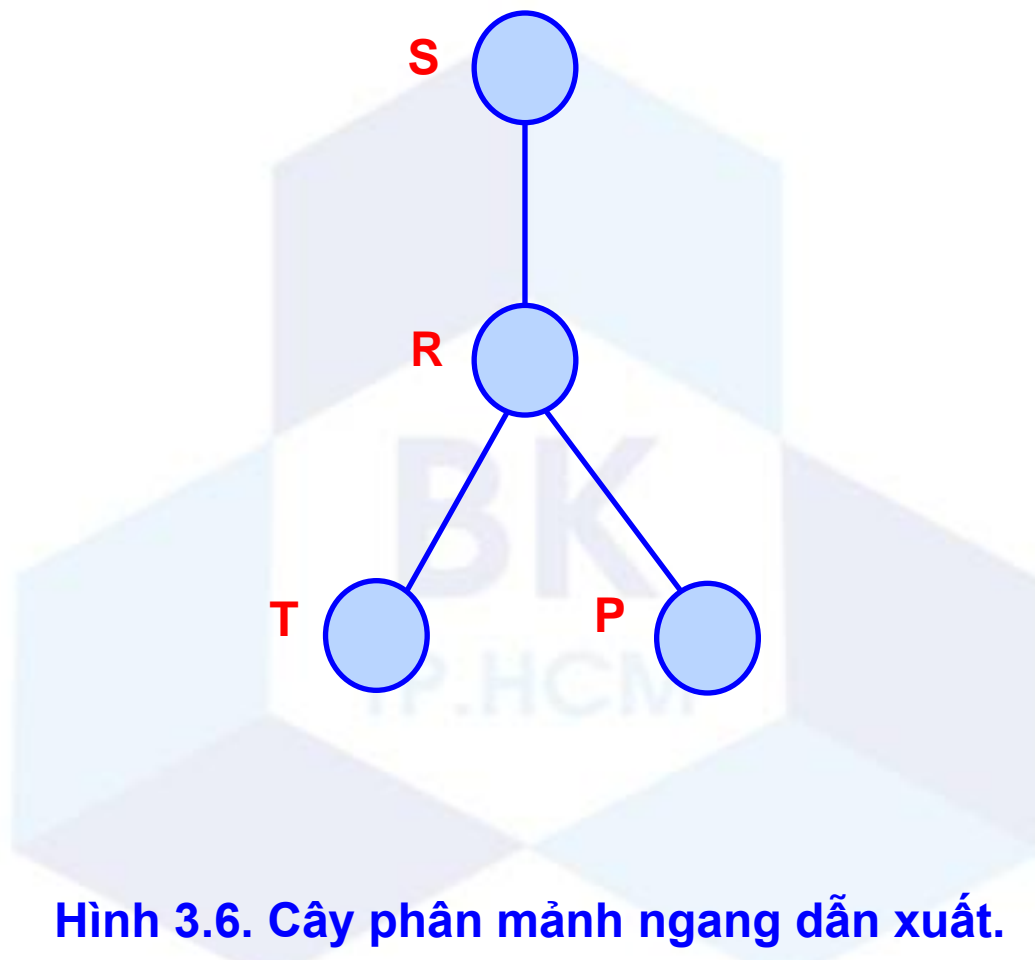
# Phân mảnh ngang dẫn xuất

## ❖ Ví dụ

### ► Xét các điều kiện đúng đắn

- **Điều kiện đầy đủ:** được thỏa mãn nếu tất cả mã sinh viên *mãsv* trong quan hệ *điểmthi* đều phải có trong quan hệ *sinhviên* (ràng buộc tham chiếu).
- **Điều kiện tái tạo:**  
$$\text{điểmthi} = \text{điểmthi}_1 \cup \text{điểmthi}_2$$
- **Điều kiện tách biệt:** được thỏa mãn nếu một bộ của quan hệ *điểmthi* không tương ứng với hai bộ của quan hệ *sinhviên* thuộc hai mảnh khác nhau, khi đó *mãsv* là khóa của *sinhviên*.

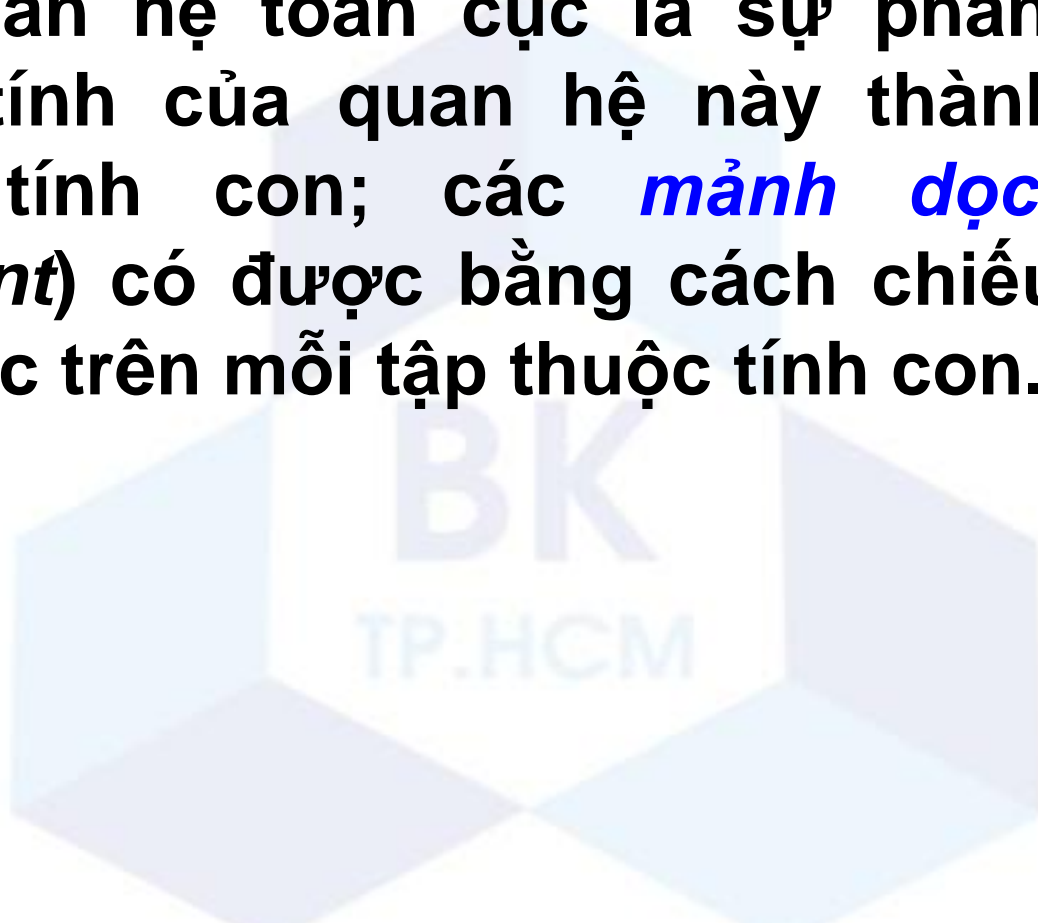
# Phân mảnh ngang dẫn xuất



Hình 3.6. Cây phân mảnh ngang dẫn xuất.

# Phân mảnh dọc

❖ **Phân mảnh dọc** (*vertical fragmentation*) của một quan hệ toàn cục là sự phân chia tập thuộc tính của quan hệ này thành các tập thuộc tính con; các **mảnh dọc** (*vertical fragment*) có được bằng cách chiếu quan hệ toàn cục trên mỗi tập thuộc tính con.





# Phân mảnh dọc

## ❖ Ví dụ

### ► Quan hệ toàn cục:

giảngviên (mãgv, họ tên, phái, mãsép, mãbm)

### ► Phân mảnh dọc không dư thừa:

$gv_1 = \Pi_{mãgv, họ tên, lương} \text{giảngviên}$

$gv_2 = \Pi_{mãgv, mãsép, mãpb} \text{giảngviên}$

$\text{giảngviên} = gv_1 \bowtie gv_2$  mãgv = mãgv

### ► Phân mảnh dọc dư thừa:

$gv_1 = \Pi_{mãgv, họ tên, lương} \text{giảngviên}$

$gv_2 = \Pi_{mãgv, họ tên, mãsép, mãbm} \text{giảngviên}$

$\text{giảngviên} = gv_1 \bowtie \Pi_{mãgv, mãsép, mãbm} gv_2$

# Phân mảnh hỗn hợp

- ❖ Một mảnh ngang được phân mảnh dọc.
- ❖ Một mảnh dọc được phân mảnh ngang.

- ❖ Ví dụ

- ▶ Quan hệ toàn cục:

giảngviên (mãgv, họ tên, lương, māsép, mấbm)

- ▶ Phân mảnh hỗn hợp:

$gv_1 = \sigma_{mấbm \leq 10} \Pi_{mãgv, \text{họ tên}, māsép, mấbm} \text{giảngviên}$

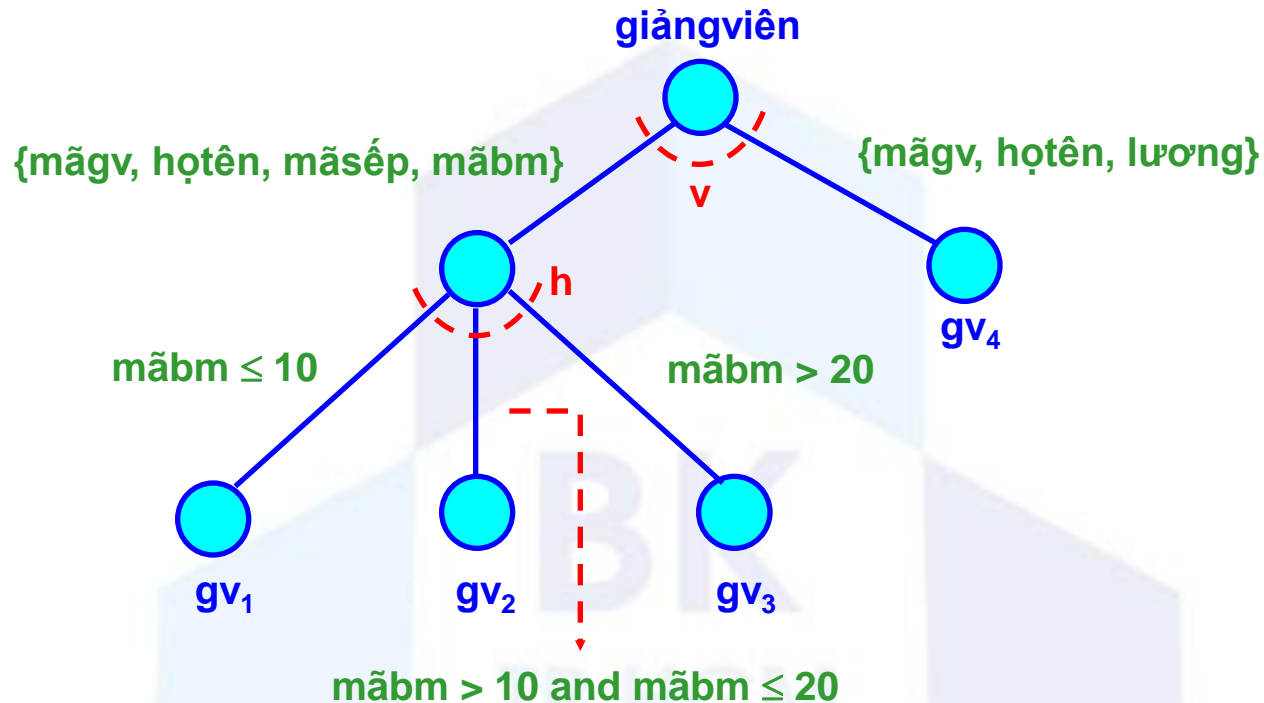
$gv_2 = \sigma_{10 < mấbm \leq 20} \Pi_{mãgv, \text{họ tên}, māsép, mấbm} \text{giảngviên}$

$gv_3 = \sigma_{mấbm > 20} \Pi_{mãgv, \text{họ tên}, māsép, mấbm} \text{giảngviên}$

$gv_4 = \Pi_{mãgv, \text{họ tên}, lương} \text{nhanviên}$

$\text{giảngviên} = (gv_1 \cup gv_2 \cup gv_3) \triangleright \triangleleft_{mãgv = mãgv} \Pi_{mãgv, \text{phái}} gv_4$

# Phân mảnh hỗn hợp



Hình 3.7. Cây phân mảnh của quan hệ *giảngviên*.

# Phân mảnh hỗn hợp

## ❖ Cây phân mảnh (*fragmentation tree*)

- ▶ **Nút gốc** tương ứng với một quan hệ toàn cục.
- ▶ Các **nút lá** tương ứng với các mảnh cuối cùng.
- ▶ Các **nút trung** gian tương ứng với các kết quả trung gian của các biểu thức định nghĩa mảnh.
- ▶ Tập hợp các nút con của một nút biểu diễn sự phân mảnh của nút này (phân mảnh ngang hoặc phân mảnh dọc).

# Example.DDB

## ❖ Lược đồ toàn cục

giảngviên (mãgv, họ tên, lương, mấsép, mấbm)

bộ môn (mấbm, tênbm, dấynhà, mấsép)

sinhviên (mấsv, họ tên, khoa)

điểmtthi (mấsv, mấmh, mấbm, điểm)

## ❖ Lược đồ phân mảnh

giảngviên<sub>1</sub> =  $\sigma_{mấbm \leq 10} \Pi_{mấnv, họ\ tên, mấsép, mấbm}$  giảngviên

giảngviên<sub>2</sub> =  $\sigma_{10 < mấbm \leq 20} \Pi_{mấnv, họ\ tên, mấsép, mấbm}$  giảngviên

giảngviên<sub>3</sub> =  $\sigma_{mấbm > 20} \Pi_{mấnv, họ\ tên, mấsép, mấbm}$  giảngviên

giảngviên<sub>4</sub> =  $\Pi_{mấnv, họ\ tên, lương}$  giảngviên

# Example.DDB

## ❖ Lược đồ phân mảnh

$\text{bộ môn}_1 = \sigma_{\text{mãbm} \leq 10} \text{bộ môn}$

$\text{bộ môn}_2 = \sigma_{10 < \text{mãbm} \leq 20} \text{bộ môn}$

$\text{bộ môn}_3 = \sigma_{\text{mãbm} > 20} \text{bộ môn}$

$\text{sinhviên}_1 = \sigma_{\text{khoa} = \text{'CNTT'}} \text{sinhviên}$

$\text{sinhviên}_2 = \sigma_{\text{khoa} = \text{'DIEN'}} \text{sinhviên}$

$\text{điểmthi}_1 = \text{điểmthi} \triangleright \text{<}_{\text{mãsv} = \text{mãsv}} \text{sinhviên}_1$

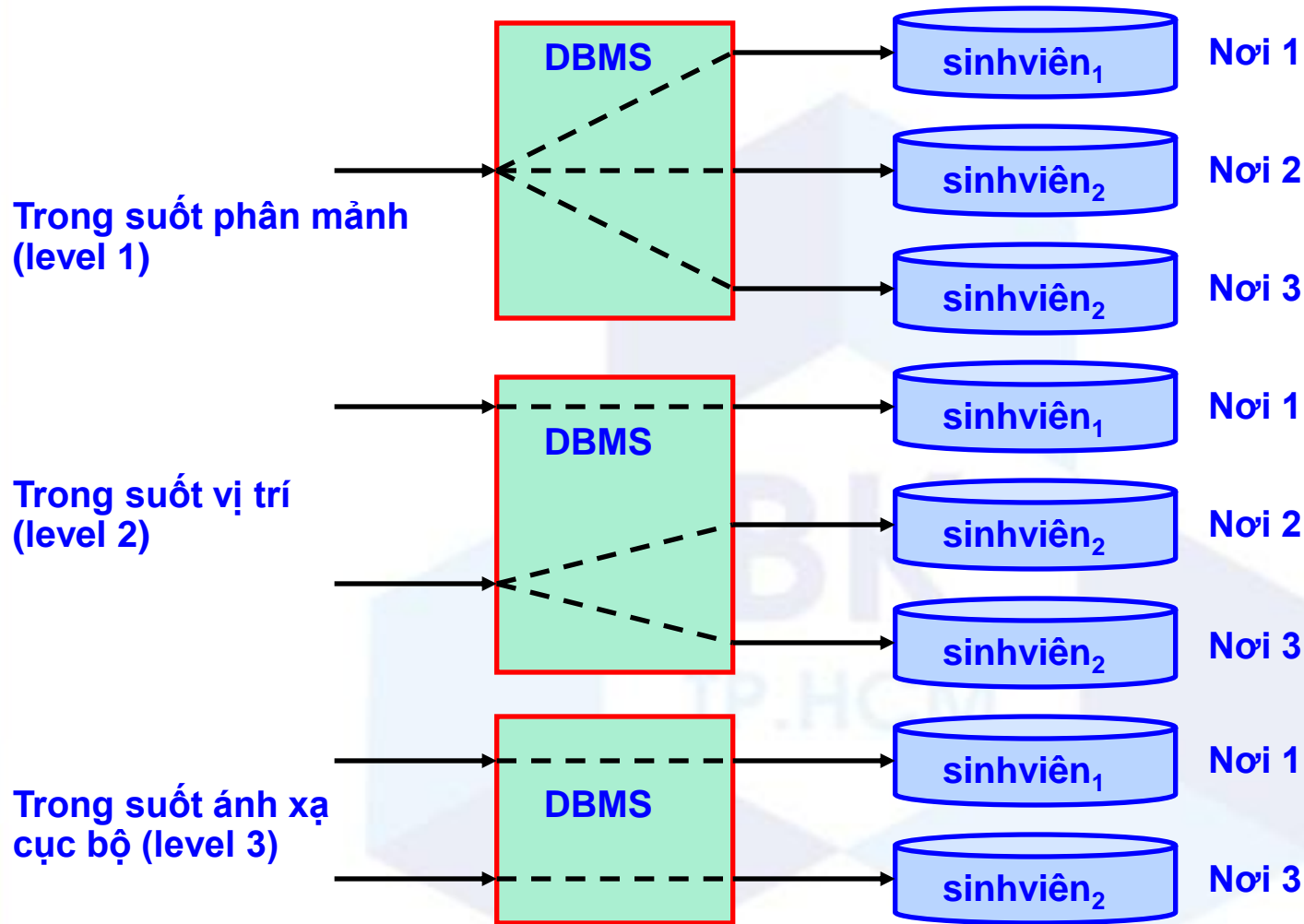
$\text{điểmthi}_2 = \text{điểmthi} \triangleright \text{<}_{\text{mãsv} = \text{mãsv}} \text{sinhviên}_2$

# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

## ❖ Ví dụ 1

- ▶ Cho biết tên của sinh viên có mã được nhập từ thiết bị đầu cuối.
- ▶ **Mức 1 – Trong suốt phân mảnh**  
read (terminal, \$mãsv);  
select *họ tên* into \$họ tên  
from *sinhvien*  
where *mãsv* = \$mãsv;  
if #FOUND then write (terminal, \$họ tên)  
else write (terminal, 'Not found');

# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc



Hình 3.8. Ứng dụng chỉ đọc ở các mức trong suốt khác nhau.



# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

## ► Mức 2 – Trong suốt vị trí

```
read (terminal, $mãsv);  
select họ tên into $họ tên  
from sinhviên1  
where mãsv = $mãsv;  
if not #FOUND then  
    select họ tên into $họ tên  
    from sinhviên2  
    where mãsv = $mãsv;  
if #FOUND then  
    write (terminal, $họ tên)  
else write (terminal, 'Not found');
```

# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

## ► Mức 3 – Trong suốt ánh xạ cục bộ

```
read (terminal, $mãsv);  
select họ tên into $họ tên  
from sinhviên1 at site 1  
where mãsv = $mãsv;  
if not #FOUND then  
    select họ tên into $họ tên  
    from sinhviên2 at site 3  
    where mãsv = $mãsv;  
if #FOUND then  
    write (terminal, $họ tên)  
else write (terminal, 'Not found');
```

# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

- ▶ Trường hợp dữ liệu nhập có liên quan đến vị từ định tính của mảnh

```
read (terminal, $mãsv);  
read (terminal, $khoa);  
case $khoa of  
  'CNTT': select họ tên into $họ tên  
           from sinhviên1  
           where mãsv = $mãsv;  
  'DIEN': select họ tên into $họ tên  
           from sinhviên2  
           where mãsv = $mãsv;  
end;  
if #FOUND then write (terminal, $họ tên)  
else write (terminal, 'Not found');
```

# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

## ❖ Ví dụ 2

- ▶ Cho biết tên của bộ môn phụ trách môn học có mã môn học được nhập từ thiết bị đầu cuối.
- ▶ Một môn học chỉ thuộc một bộ môn phụ trách.



# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

## ► Mức 1 – Trong suốt phân mảnh

```
read (terminal, $mãmh);  
select tênbm into $tênbm  
from điểmthi, bộmôn  
where điểmthi.mãbm = bộmôn.mãbm  
and điểmthi.mãmh = $mãmh;  
if #FOUND then write (terminal, $tênbm)  
else write (terminal, 'Not found');
```

# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

## ► Mức 2 – Trong suốt vị trí

```
read (terminal, $mãmh);  
select tênbm into $tênbm  
from điểmthi1, bộ môn1  
where điểmthi1.mãbm = bộ môn1.mãbm  
and điểmthi1.mãmh = $mãmh;  
if not #FOUND then  
select tênbm into $tênbm  
from điểmthi2, bộ môn2  
where điểmthi2.mãbm = bộ môn2.mãbm  
and điểmthi2.mãmh = $mãmh;  
if #FOUND then write (terminal, $tênbm)  
else write (terminal, 'Not found');
```

# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

## ► Mức 3 – Trong suốt ánh xạ cục bộ

$bộ môn_1$  đặt tại nơi 1.

$bộ môn_2$  đặt tại nơi 2.

$điểm thi_1$  đặt tại nơi 3.

$điểm thi_2$  đặt tại nơi 4.



# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

```
read (terminal, $mãmh);  
select mãbm into $mãbm  
from điểmthi1 at site 3  
where mãmh = $mãmh;  
if #FOUND then  
  begin  
    send $mãbm from site 3 to site 1;  
    select tênbm into $tênbm  
    from bộ môn1 at site 1  
    where mãbm = $mãbm;  
  end
```



# Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

```
else
  begin
    select mãbm into $mãbm
    from điểmthi2 at site 4
    where mãmh = $mãmh;
    if #FOUND then
      begin
        send $mãbm from site 4 to site 2;
        select tênbm into $tênbm
        from bộmôn2 at site 2
        where mãbm = $mãbm;
      end
    end;
    if #FOUND then write (terminal, $tênbm)
    else write (terminal, 'Not found');
```

# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

- ❖ Cập nhật dữ liệu (thêm, sửa, xóa) phải bảo đảm các ràng buộc toàn vẹn về khóa chính, khóa ngoại, phụ thuộc hàm, ràng buộc nghiệp vụ ...
- ❖ **Qui tắc *read-one write-all***
  - ▶ Việc lấy dữ liệu có thể được thực hiện từ bất kỳ một bản sao nào.
  - ▶ Việc cập nhật một mục dữ liệu phải được thực hiện trên tất cả các bản sao của nó.

# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

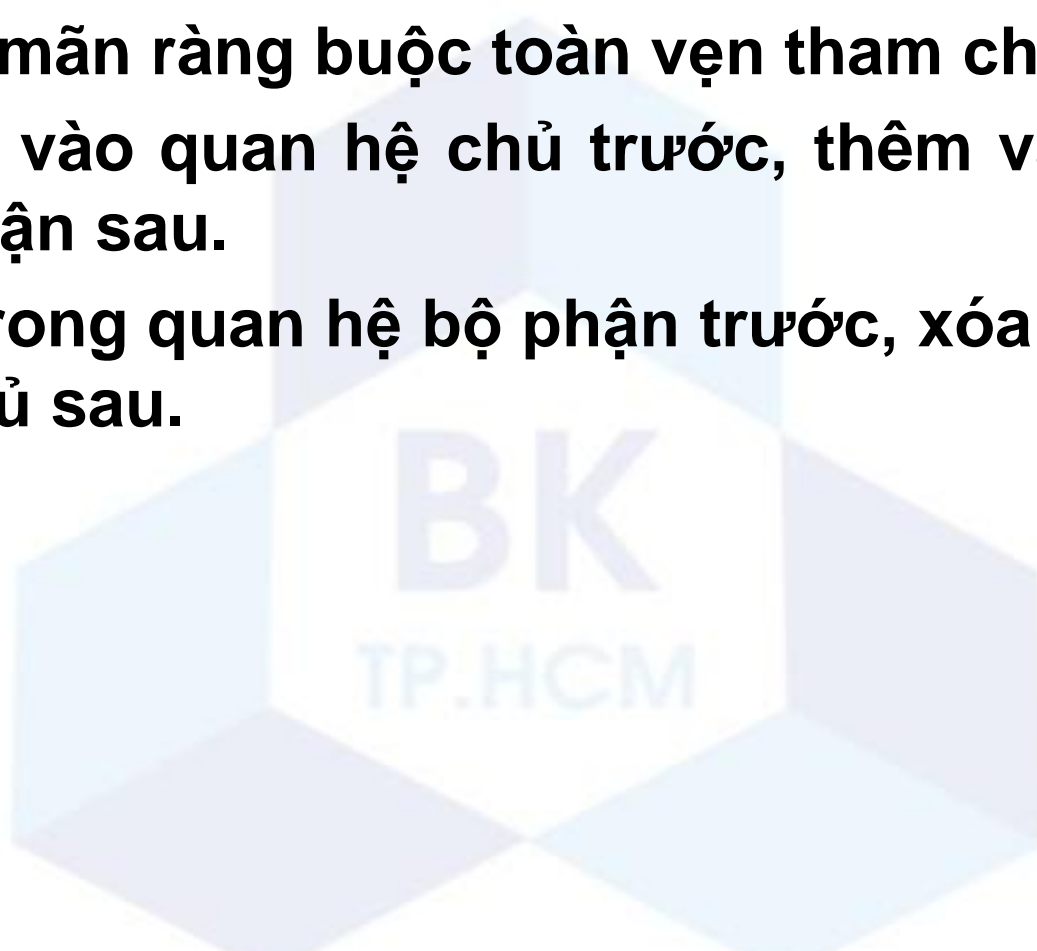
## ❖ Các trường hợp sửa dữ liệu trong CSDL phân tán

- ▶ Nếu mục dữ liệu bị sửa *không có trong vị từ định tính*: sửa dữ liệu giống như trong CSDL tập trung.
- ▶ Nếu mục dữ liệu bị sửa có trong vị từ định tính và *giá trị của vị từ định tính không bị thay đổi* khi thay thế dữ liệu cũ và dữ liệu mới: sửa dữ liệu giống như trong CSDL tập trung.
- ▶ Nếu mục dữ liệu bị sửa có trong vị từ định tính và *giá trị của vị từ định tính bị thay đổi* khi thay thế dữ liệu cũ và dữ liệu mới: di chuyển dữ liệu giữa các mảnh.

# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

## ❖ Quy tắc *owner - member*

- ▶ Thỏa mãn ràng buộc toàn vẹn tham chiếu.
- ▶ Thêm vào quan hệ chủ trước, thêm vào quan hệ bộ phận sau.
- ▶ Xóa trong quan hệ bộ phận trước, xóa trong quan hệ chủ sau.

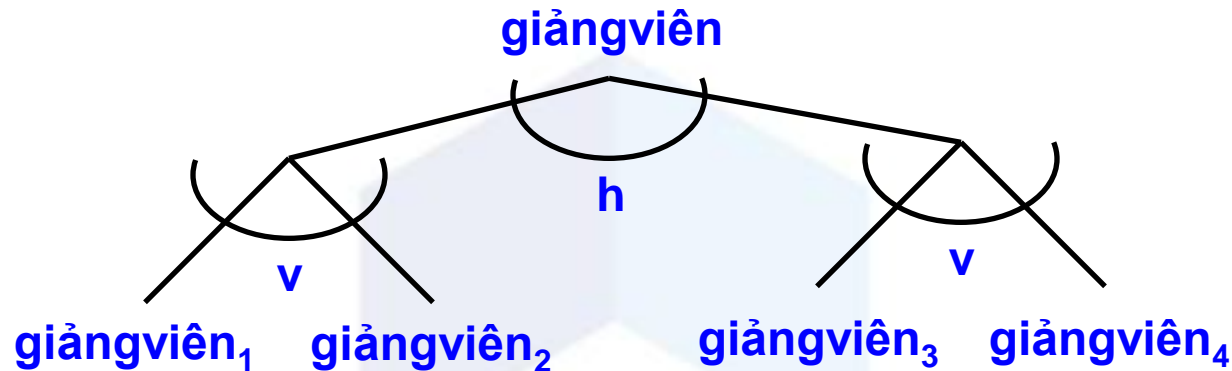


# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

## ❖ Ví dụ

- ▶ Sửa dữ liệu của giảng viên có mã 100: mã bộ môn 3 thành mã bộ môn 15.
- ▶ Các mảnh:
  - giảngviên<sub>1</sub> được đặt tại nơi 1 và 5.
  - giảngviên<sub>2</sub> được đặt tại nơi 2 và 6.
  - giảngviên<sub>3</sub> được đặt tại nơi 3 và 7.
  - giảngviên<sub>4</sub> được đặt tại nơi 4 và 8.

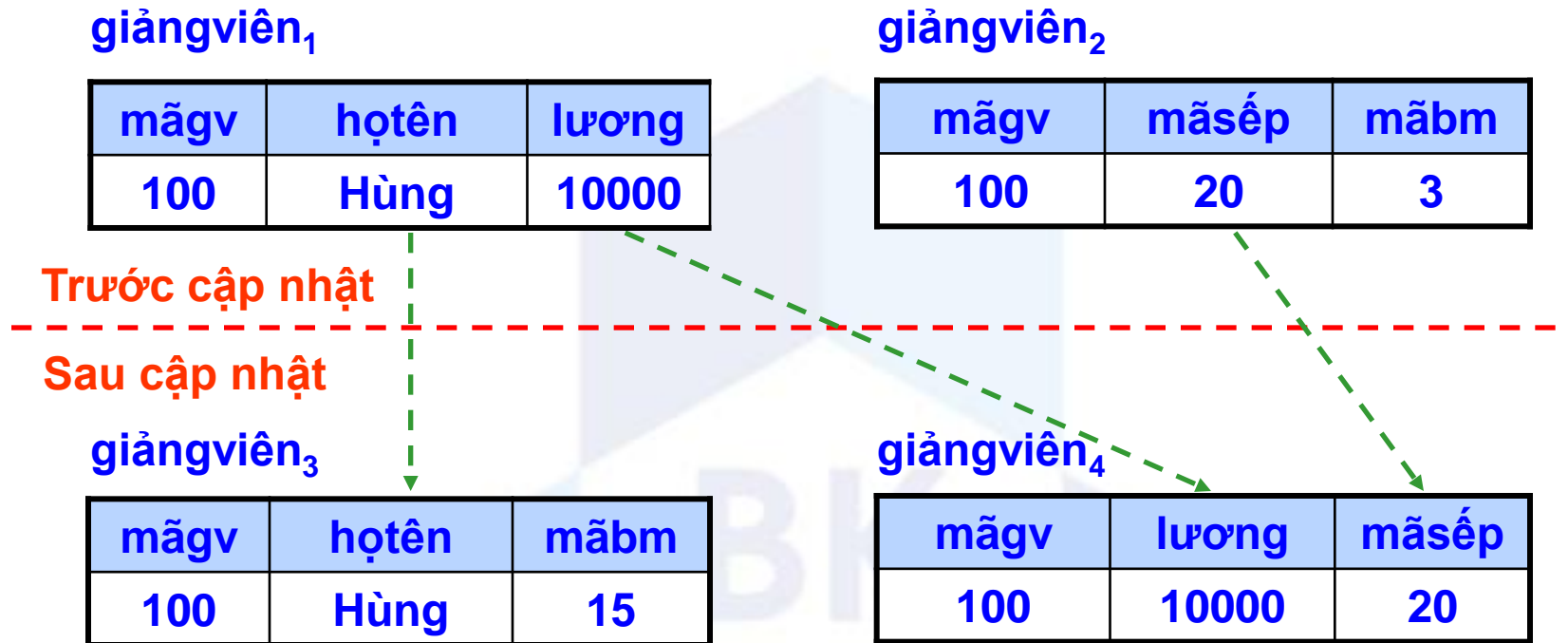
# Tính trong suốt phân tán dùng cho ứng dụng cập nhật



$\text{giảngviên}_1 = \Pi_{\text{mãgv, họ tên, lương}} \sigma_{\text{mãbm} \leq 10} \text{giảngviên}$   
 $\text{giảngviên}_2 = \Pi_{\text{mãgv, mãsép, mãbm}} \sigma_{\text{mãbm} \leq 10} \text{giảngviên}$   
 $\text{giảngviên}_3 = \Pi_{\text{mãgv, mãsép, mãbm}} \sigma_{\text{mãbm} > 10} \text{giảngviên}$   
 $\text{giảngviên}_4 = \Pi_{\text{mãgv, mãsép, mãbm}} \sigma_{\text{mãbm} > 10} \text{giảngviên}$

Hình 3.9. Cây phân mảnh của quan hệ *giảngviên*.

# Tính trong suốt phân tán dùng cho ứng dụng cập nhật



Hình 3.10. Ứng dụng cập nhật.

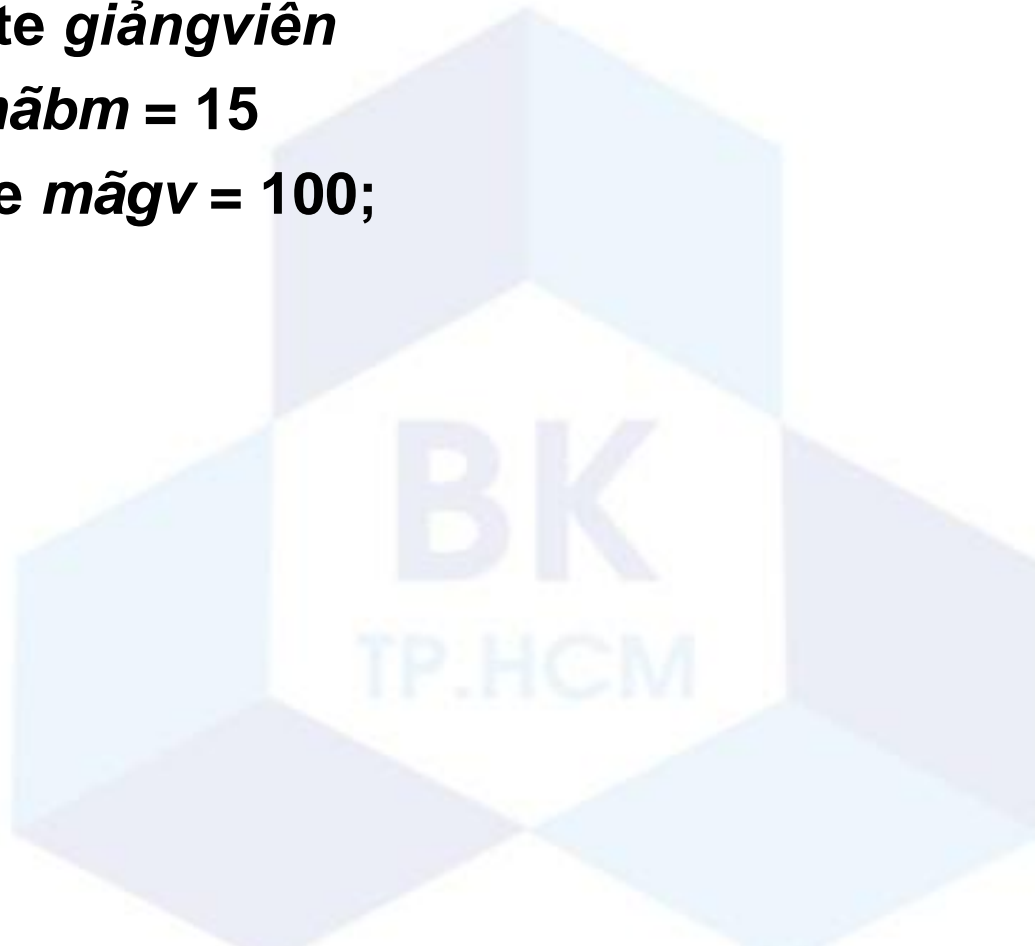
# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

## ► Mức 1 – Trong suốt phân mảnh

update *giảngviên*

set *mãbm* = 15

where *mãgv* = 100;





# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

## ► Mức 2 – Trong suốt vị trí

```
select họ tên, lương into $họ tên, $lương  
from giảngviên1  
where mãgv = 100;  
if #FOUND then  
begin  
    select mãsếp into $mãsếp  
    from giảngviên2  
    where mãgv = 100;  
    insert into giảngviên3 (mãgv, họ tên, mãbm)  
    values (100, $họ tên, 15);
```

# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

```
insert into giảngviên4 (mãgv, lương, mãsếp)  
values (100, $lương, $mãsếp);  
delete from giảngviên1  
where mãgv = 100;  
delete from giảngviên2  
where mãgv = 100  
end;
```



# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

## ► Mức 3 – Trong suốt ánh xạ cục bộ

**select *họ tên, lương* into \$*họ tên*, \$*lương***

**from giảngviên1 at site 1**

**where *mãgv* = 100;**

**if #FOUND then**

**begin**

**select *mãsếp* into \$*mãsếp***

**from giảngviên2 at site 2**

**where *mãgv* = 100;**

**insert into giảngviên3 (*mãgv, họ tên, mãbm*) at site 3  
values (100, \$*họ tên*, 15);**

# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

insert into giảngviên3 (*mãgv*, *họ tên*, *mãbm*) at site 7  
values (100, \$*họ tên*, 15);

insert into giảngviên4 (*mãgv*, *lương*, *mãsếp*) at site 4  
values (100, \$*lương*, \$*mãsếp*);

insert into giảngviên4 (*mãgv*, *lương*, *mãsếp*) at site 8  
values (100, \$ *lương*, \$ *mãsếp*);

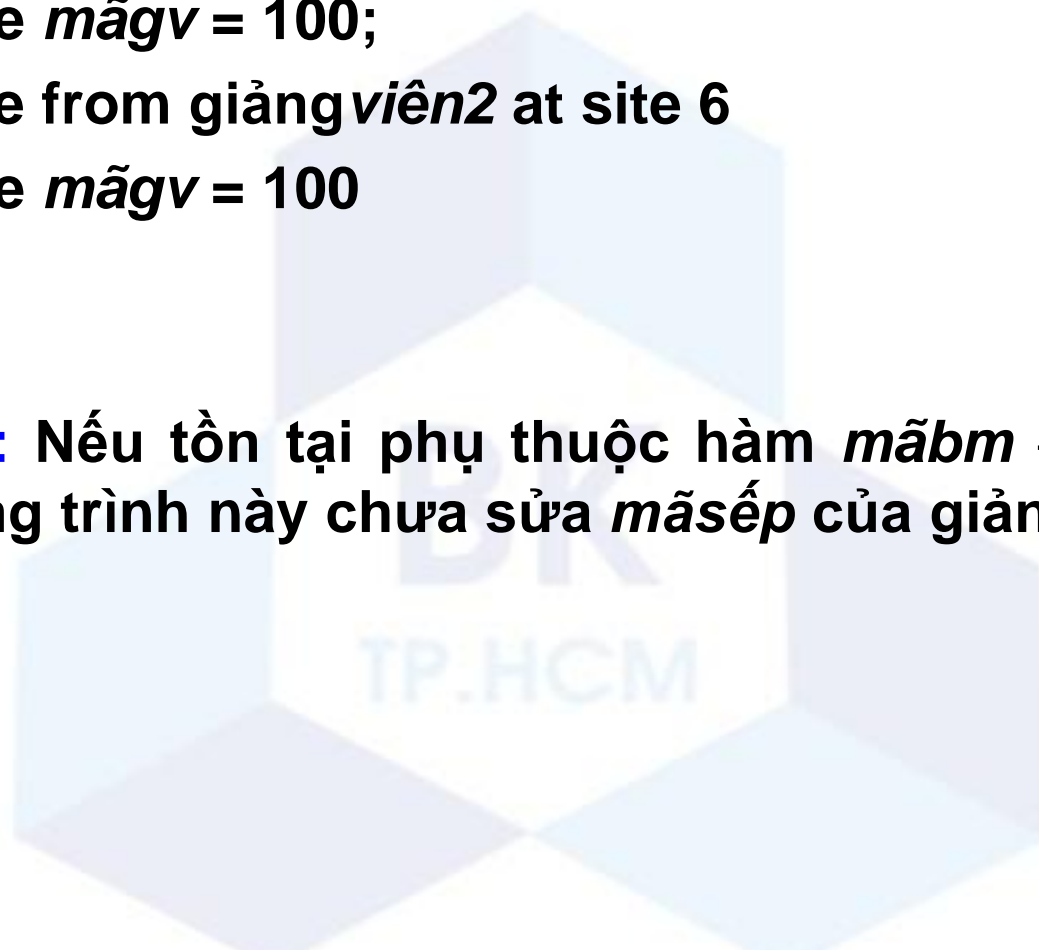
delete from giảngviên1 at site 1  
where *mãgv* = 100;

delete from giảngviên1 at site 5  
where *mãgv* = 100;

# Tính trong suốt phân tán dùng cho ứng dụng cập nhật

```
delete from giảngviên2 at site 2  
where  $mãgv = 100$ ;  
delete from giảngviên2 at site 6  
where  $mãgv = 100$   
end;
```

- **Lưu ý:** Nếu tồn tại phụ thuộc hàm  $mãbm \rightarrow mãsép$  thì chương trình này chưa sửa  $mãsép$  của giảng viên này.



# Các tác vụ cơ bản truy xuất CSDL phân tán

## ❖ Ví dụ

- ▶ Cho biết tất cả mã môn học của các sinh viên cho trước.



# Các tác vụ cơ bản truy xuất CSDL phân tán

## ► Truy xuất CSDL ứng với mỗi mã sinh viên

repeat

read (terminal, \$mãsv);

delete from \$mãmh\_rel;

insert into \$mãmh\_rel (\$mãmh)

select *mãmh*

from *điểmthi*

where *mãsv* = \$mãsv;

repeat

read (\$mãmh\_rel, \$mãmh);

write (terminal, \$mãsv, \$mãmh)

until END-OF-\$mãmh\_rel

until END-OF-TERMINAL-INPUT;

# Các tác vụ cơ bản truy xuất CSDL phân tán

- ▶ Truy xuất CSDL sau khi nhập vào tất cả mã sinh viên

```
repeat
  read (terminal, $mãsv);
  insert into $mãsv_rel ($mãsv)
  values ($mãsv)
until END-OF-TERMINAL-INPUT;
insert into $mãmh_rel ($mãsv, $mãmh)
select $mãsv, mãmh
from điểmthi, $mãsv_rel
where điểmthi. mãsv = $mãsv_rel.$mãsv;
repeat
  read ($mãmh_rel, $mãsv, $mãmh);
  write (terminal, $mãsv, $mãmh)
until END-OF-$mãmh_rel;
```



# Các tác vụ cơ bản truy xuất CSDL phân tán

- ▶ Truy xuất CSDL trước khi nhập vào tất cả mã sinh viên

```
insert into $temp_rel ($temp_mãmh, $temp_mãsv)
select mãmh, mãsv
from điểmithi;
repeat
    read (terminal, $mãsv);
    delete from $temp2_rel;
    insert into $temp2_rel ($temp2_mãmh)
    select $temp_mãmh
    from $temp_rel
    where $temp_mãsv = $mãsv;
```

# Các tác vụ cơ bản truy xuất CSDL phân tán

```
repeat
  read ($temp2_rel, $temp2_mãmh);
  write (terminal, $mãsv, $temp2_mãmh)
until END-OF-$temp2_rel
until END-OF-TERMINAL-INPUT;
```

