

Chương 3

Các mức trong suốt phân tán

Nội dung

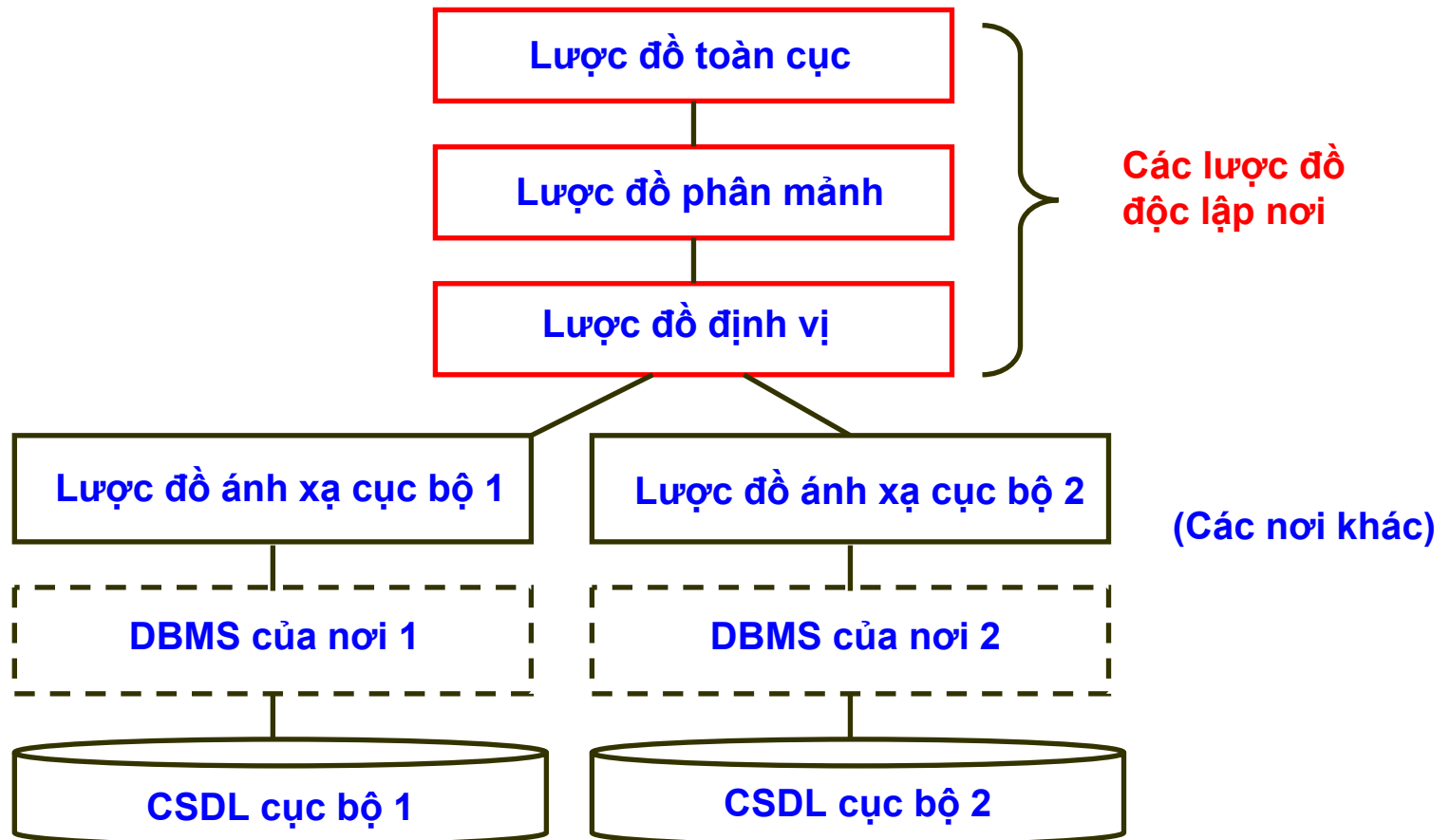
- ❖ Kiến trúc tham khảo của CSDL phân tán.
- ❖ Các loại phân mảnh dữ liệu.
- ❖ Các điều kiện đúng dẫn để phân mảnh dữ liệu.
- ❖ Phân mảnh ngang chính.
- ❖ Phân mảnh ngang dẫn xuất.
- ❖ Phân mảnh dọc.
- ❖ Phân mảnh hỗn hợp.
- ❖ Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc.



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*
- ▶ Lược đồ toàn cục (*global schema*)
- ▶ Quan hệ toàn cục (*global relation*)

❖ Trong suốt vị trí

- ▶ *location transparency*
- ▶ Lược đồ phân mảnh (*fragmentation schema*)
- ▶ Mảnh (*fragment*)

❖ Trong suốt nhân bản

- ▶ *replication transparency*
- ▶ Bản nhân (*replica*)

Kiến trúc tham khảo của CSDL phân tán

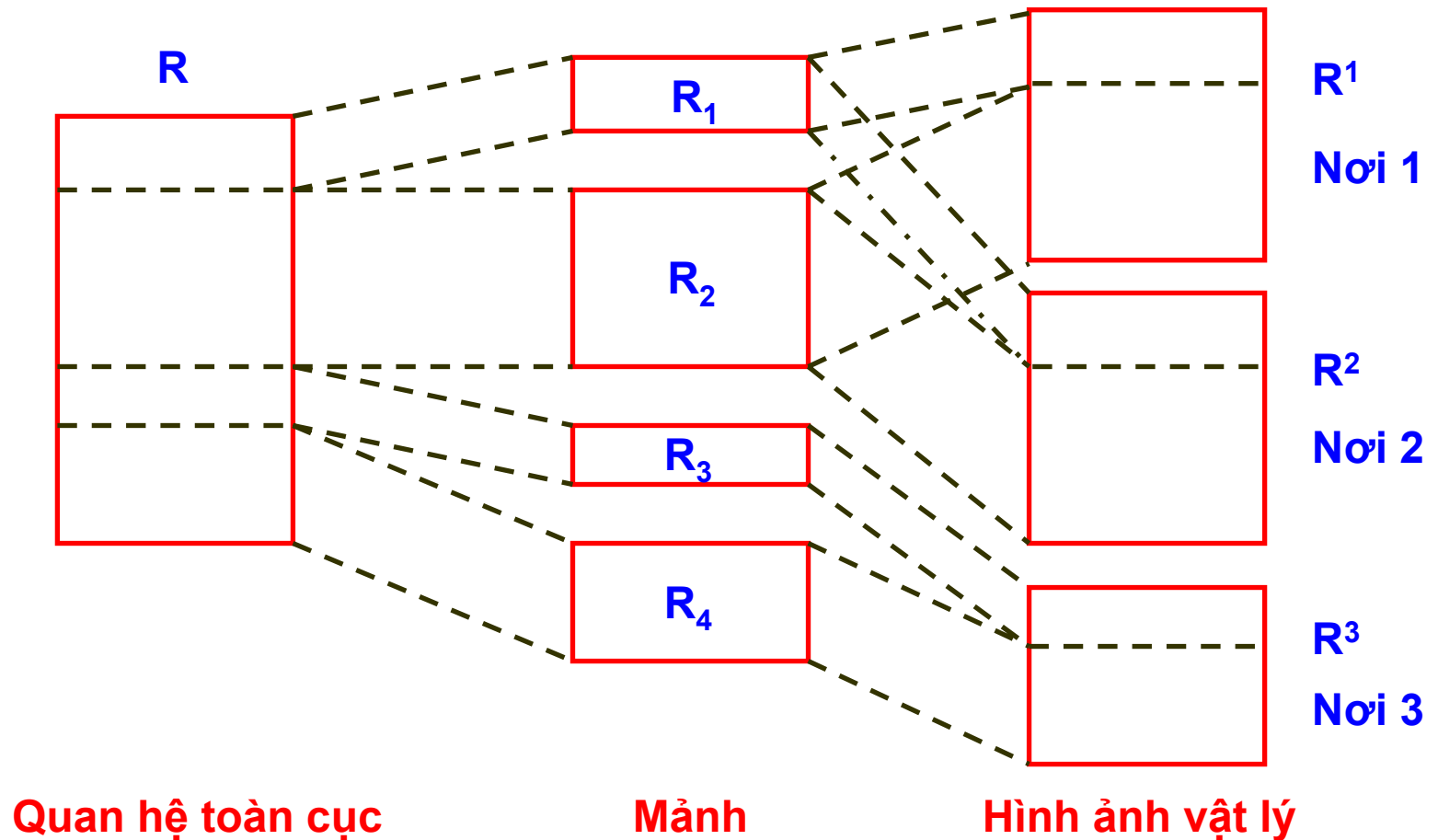
❖ Trong suốt ánh xạ cục bộ

- ▶ *local mapping transparency*
- ▶ Lược đồ định vị (*allocation schema*)
- ▶ Quan hệ cục bộ (*local relation*)

❖ 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.

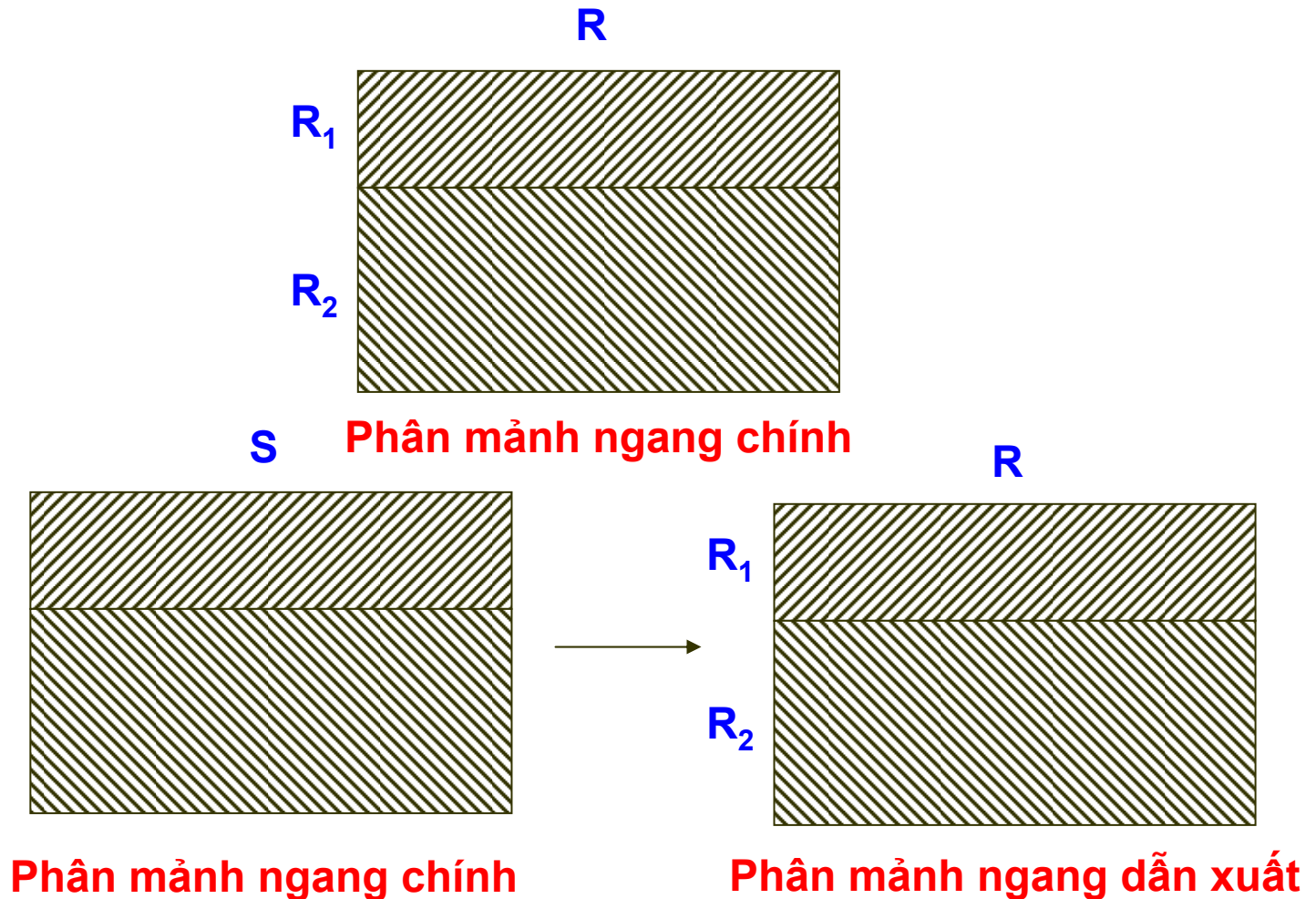


Các loại phân mảnh dữ liệu

❖ 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



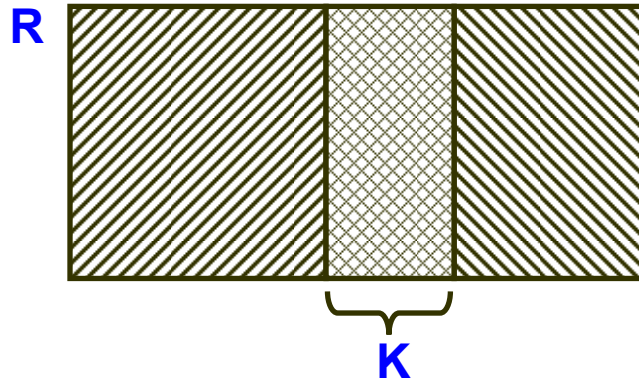
Hình 3.3. Các loại phân mảnh ngang.

Các loại phân mảnh dữ liệu

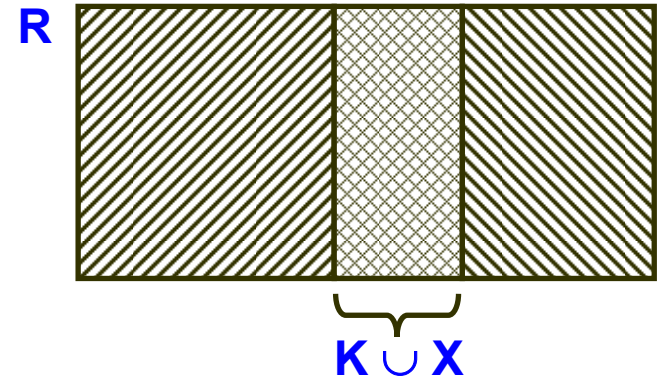
❖ Phân mảnh dọc

- ▶ *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*)

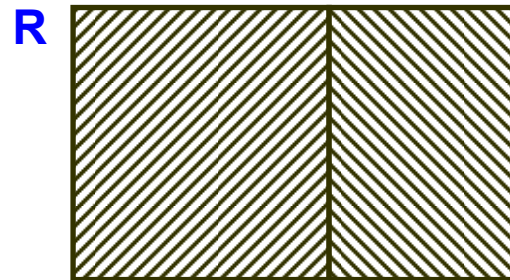
Các loại phân mảnh dữ liệu



Phân mảnh gom tụ không dư thừa



Phân mảnh gom tụ có dư thừa



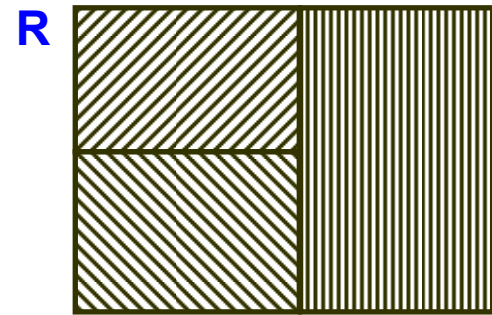
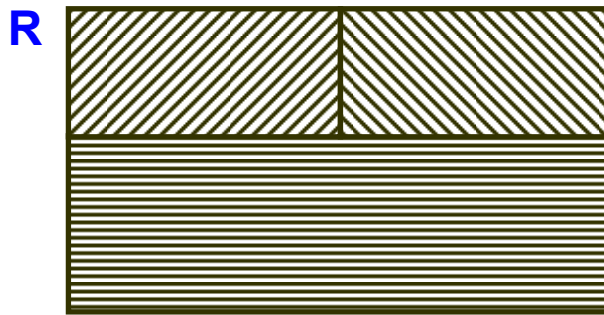
Phân mảnh dọc tách biệt

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*
 - ▶ Mỗi 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 ∇ sao cho:

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

▶ **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

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

Phân mảnh ngang chính

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

Phân mảnh ngang chính

❖ Ví dụ

▶ Quan hệ toàn cục:

supplier (snum, name, city)

▶ Các mảnh ngang:

supplier₁ = $\sigma_{\text{city} = \text{'SF'}}$ supplier

supplier₂ = $\sigma_{\text{city} = \text{'LA'}}$ supplier

▶ Các vị từ định tính:

q₁: city = 'SF'

q₂: city = 'LA'

▶ Xét các điều kiện đúng đắn



Phân mảnh ngang chính

- ❖ Điều kiện đầy đủ: tập hợp các vị từ định tính của tất cả các mảnh ngang phải đầy đủ.
- ❖ Điều kiện tái tạo: phép hợp.
- ❖ Điều kiện tách biệt: các vị từ định tính phải loại trừ nhau.

Phân mảnh ngang dẫn xuất

- ❖ **Phân mảnh ngang dẫn xuất** (*derived horizontal fragmentation*) là sự phân chia các bộ của một quan hệ toàn cục thành các tập hợp con, được gọi là các mảnh ngang, dựa vào phân mảnh ngang của một quan hệ khác (được gọi là quan hệ chủ).
- ❖ 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:

supply (snum, pnum, deptnum, quan)

▶ Các mảnh ngang dẫn xuất:

supply₁ = supply ▷<_{snum = snum} supplier₁

supply₂ = supply ▷<_{snum = snum} supplier₂

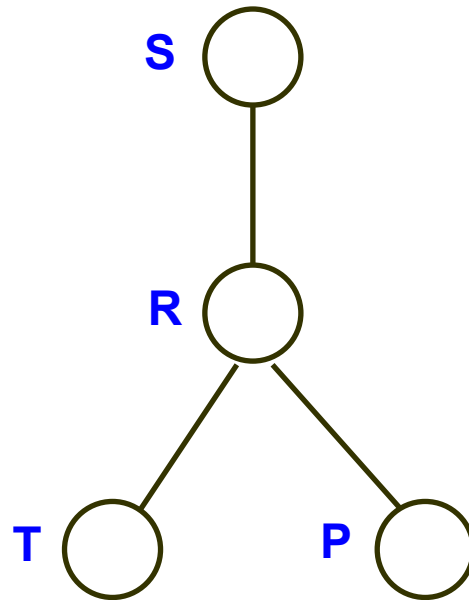
▶ Các vị từ định tính:

**q₁: supply.snum = supplier.snum AND
supplier.city = 'SF'**

**q₂: supply.snum = supplier.snum AND
supplier.city = 'LA'**

▶ Xét các điều kiện đúng đắn

Phân mảnh ngang dẫn xuất



Hình 3.6. Cây phân mảnh ngang dẫn xuất.

Phân mảnh dọc

- ❖ **Phân mảnh dọc** (*vertical fragmentation*) là sự phân chia tập thuộc tính của một quan hệ toàn cục 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 đọc

❖ Ví dụ

► Quan hệ toàn cục:

emp (empnum, name, sal, tax, mgrnum, deptnum)

► Phân mảnh đọc không dư thừa:

$\text{emp}_1 = \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$

$\text{emp}_2 = \Pi_{\text{empnum, sal, tax}} \text{emp}$

$\text{emp} = \text{emp}_1 \bowtie_{\text{empnum} = \text{empnum}} \text{emp}_2$

► Phân mảnh đọc dư thừa:

$\text{emp}_1 = \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$

$\text{emp}_2 = \Pi_{\text{empnum, name, sal, tax}} \text{emp}$

$\text{emp} = \text{emp}_1 \bowtie_{\text{empnum} = \text{empnum}} \Pi_{\text{empnum, sal, tax}} \text{emp}_2$

Phân mảnh hỗn hợp

- ❖ Một mảnh ngang được phân mảnh dọc.
- ❖ Một mảnh dọc được phân mảnh ngang.
- ❖ Ví dụ
 - ▶ Quan hệ toàn cục:
emp (empnum, name, sal, tax, mgrnum, deptnum)
 - ▶ Phân mảnh hỗn hợp:

$$\text{emp}_1 = \sigma_{\text{deptnum} \leq 10} \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$$

$$\text{emp}_2 = \sigma_{10 < \text{deptnum} \leq 20} \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$$

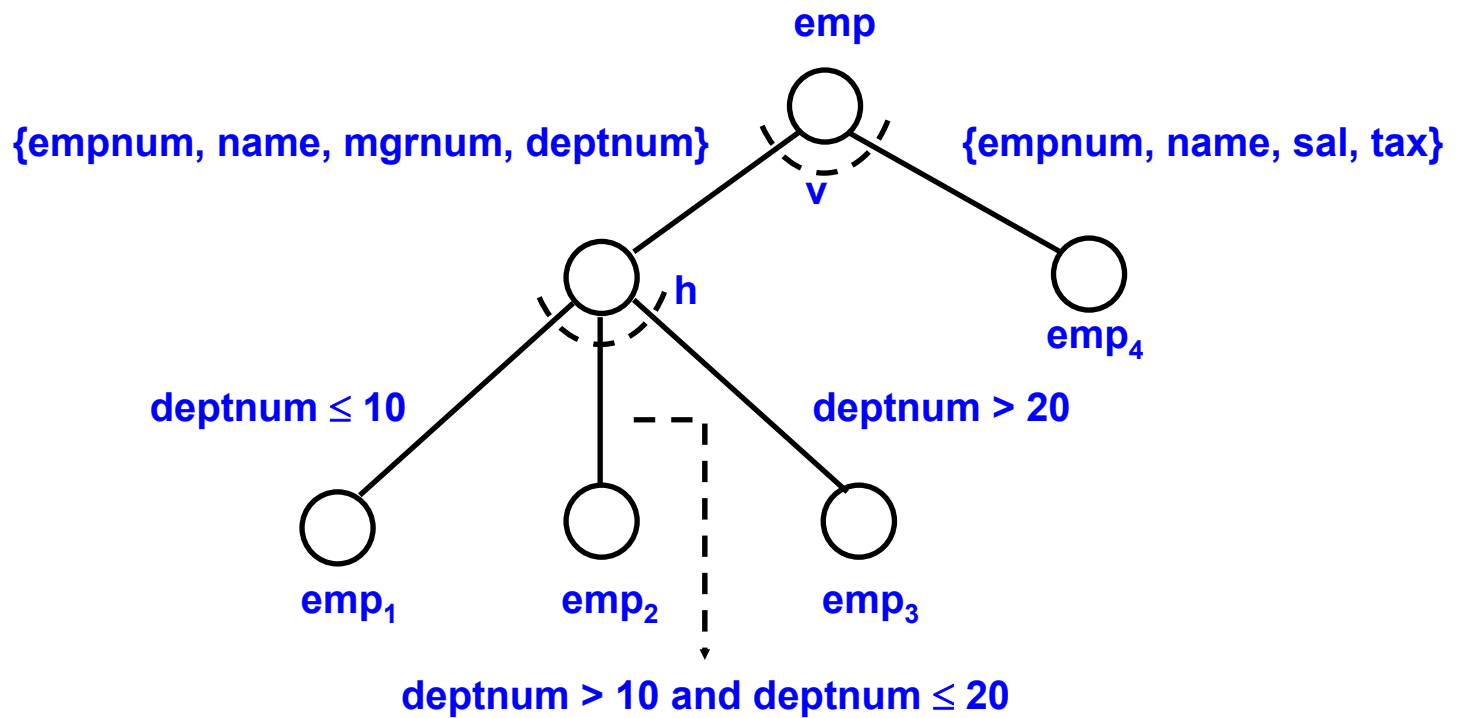
$$\text{emp}_3 = \sigma_{\text{deptnum} > 20} \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$$

$$\text{emp}_4 = \Pi_{\text{empnum, name, sal, tax}} \text{emp}$$

$$\text{emp} = (\text{emp}_1 \cup \text{emp}_2 \cup \text{emp}_3)$$

$$\triangleright \triangleleft_{\text{empnum} = \text{empnum}} \Pi_{\text{empnum, sal, tax}} \text{emp}_4$$

Phân mảnh hỗn hợp



Hình 3.7. Cây phân mảnh của quan hệ EMP.

Example.DDB

❖ Lược đồ toàn cục:

emp (empnum, name, sal, tax, mgrnum, deptnum)

dept (deptnum, name, area, mgrnum)

supplier (snum, name, city)

supply (snum, pnum, deptnum, quan)

❖ Lược đồ phân mảnh:

$\text{emp}_1 = \sigma_{\text{deptnum} \leq 10} \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$

$\text{emp}_2 = \sigma_{10 < \text{deptnum} \leq 20} \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$

$\text{emp}_3 = \sigma_{\text{deptnum} > 20} \Pi_{\text{empnum, name, mgrnum, deptnum}} \text{emp}$

$\text{emp}_4 = \Pi_{\text{empnum, name, sal, tax}} \text{emp}$

Example.DDB

❖ **Lược đồ phân mảnh:**

$\text{dept}_1 = \sigma_{\text{deptnum} \leq 10} \text{dept}$

$\text{dept}_2 = \sigma_{10 < \text{deptnum} \leq 20} \text{dept}$

$\text{dept}_3 = \sigma_{\text{deptnum} > 20} \text{dept}$

$\text{supplier}_1 = \sigma_{\text{city} = \text{'SF'}} \text{supplier}$

$\text{supplier}_2 = \sigma_{\text{city} = \text{'LA'}} \text{supplier}$

$\text{supply}_1 = \text{supply} \bowtie_{\text{snum} = \text{snum}} \text{supplier}_1$

$\text{supply}_2 = \text{supply} \bowtie_{\text{snum} = \text{snum}} \text{supplier}_2$



Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

❖ Ví dụ 1

- ▶ Cho biết tên của nhà cung cấp có mã được nhập từ thiết bị đầu cuối.

▶ Mức 1 – Trong suốt phân mảnh

```
read (terminal, $snum);  
select name into $name  
from supplier  
where snum = $snum;  
if #FOUND then write (terminal, $name)  
else write (terminal, 'Not found');
```



Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

► Mức 2 – Trong suốt vị trí

```
read (terminal, $snum);  
select name into $name  
from supplier1  
where snum = $snum;  
if not #FOUND then  
    select name into $name  
    from supplier2  
    where snum = $snum;  
if #FOUND then  
    write (terminal, $name)  
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, $snum);
```

```
read (terminal, $city);
```

```
case $city of
```

```
  'SF': select name into $name  
        from supplier1
```

```
        where snum = $snum;
```

```
  'LA': select name into $name  
        from supplier2
```

```
        where snum = $snum;
```

```
end;
```

```
if #FOUND then write (terminal, $name)
```

```
else write (terminal, 'Not found');
```



Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

❖ Ví dụ 2

- ▶ Cho biết tên của nhà cung cấp mà họ cung cấp mặt hàng có mã được nhập từ thiết bị đầu cuối.
- ▶ Giả sử một mặt hàng chỉ được cung cấp bởi một nhà cung cấp.



Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

► Mức 1 – Trong suốt phân mảnh

```
read (terminal, $pnum);  
select name into $name  
from supplier, supply  
where supplier.snum = supply.snum  
      and supply.pnum = $pnum;  
if #FOUND then write (terminal, $name)  
else write (terminal, 'Not found');
```



Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

► Mức 2 – Trong suốt vị trí

```
read (terminal, $pnum);  
select name into $name  
from supplier1, supply1  
where supplier1.snum = supply1.snum  
      and supply1.pnum = $pnum;  
if not #FOUND then  
  select name into $name  
  from supplier2, supply2  
  where supplier2.snum = supply2.snum  
        and supply2.pnum = $pnum;  
if #FOUND then write (terminal, $name)  
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*.
- ❖ Qui tắc *owner – member*.



Tính trong suốt phân tán dùng cho ứng dụng cập nhật

❖ Sửa dữ liệu trong CSDL phân tán

- ▶ Mục dữ liệu bị sửa không có trong vị từ định tính.
- ▶ 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.
- ▶ 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.

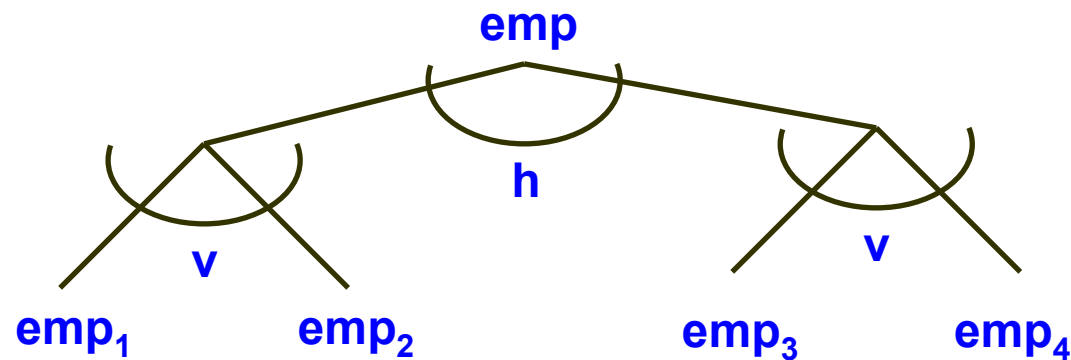


Tính trong suốt phân tán dùng cho ứng dụng cập nhật

❖ Ví dụ

- ▶ Sửa dữ liệu của nhân viên có mã 100: mã phòng 3 thành mã phòng 15.
- ▶ Các mảnh:
 - emp_1 được đặt tại nơi 1 và 5.
 - emp_2 được đặt tại nơi 2 và 6.
 - emp_3 được đặt tại nơi 3 và 7.
 - emp_4 được đặt tại nơi 4 và 8.

Tính trong suốt phân tán dùng cho ứng dụng cập nhật



$$emp_1 = \Pi_{empnum, name, sal, tax} \sigma_{deptnum \leq 10} emp$$

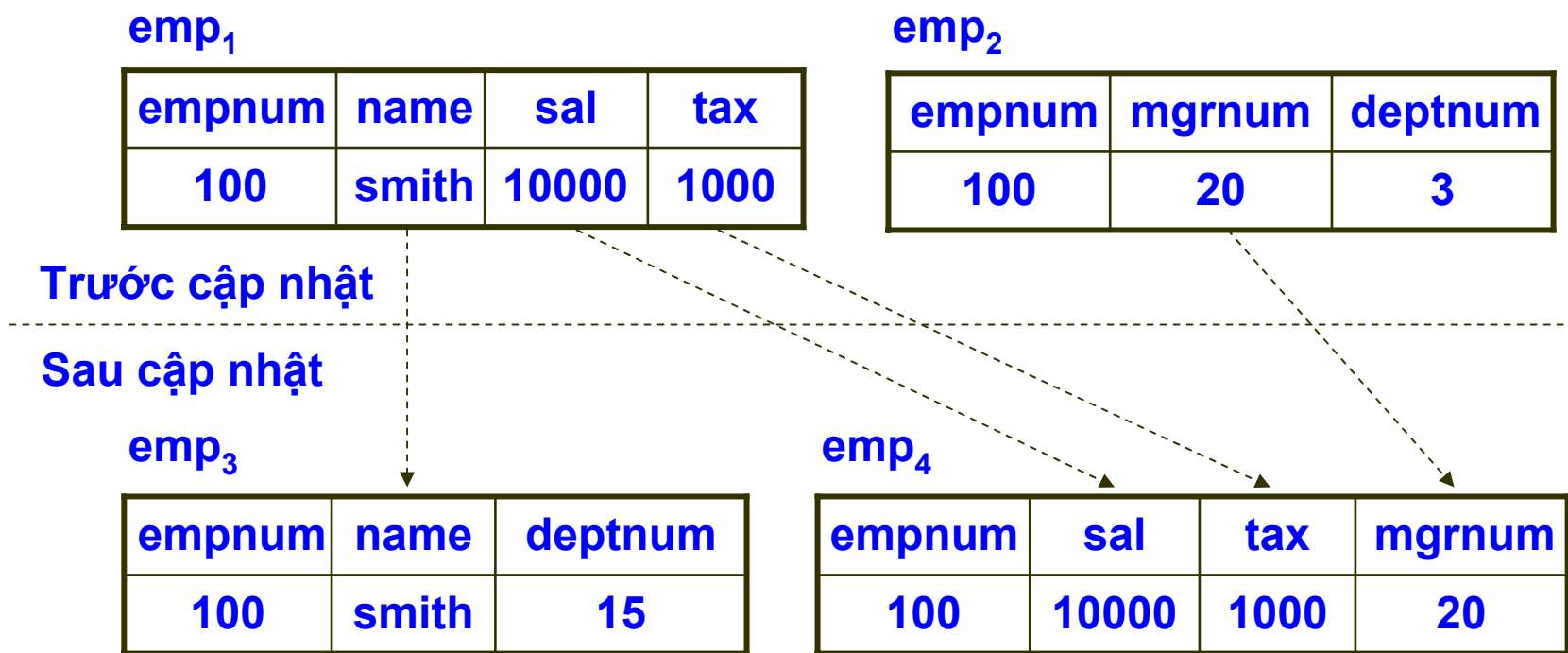
$$emp_2 = \Pi_{empnum, mgrnum, deptnum} \sigma_{deptnum \leq 10} emp$$

$$emp_3 = \Pi_{empnum, name, deptnum} \sigma_{deptnum > 10} emp$$

$$emp_4 = \Pi_{empnum, sal, tax, mgrnum} \sigma_{deptnum > 10} emp$$

Hình 3.9. Cây phân mảnh của quan hệ EMP.

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 *emp*

set *deptnum* = 15

where *empnum* = 100;



Tính trong suốt phân tán dùng cho ứng dụng cập nhật

► Mức 2 – Trong suốt vị trí

```
select name, sal, tax into $name, $sal, $tax
from emp1
where empnum = 100;
if #FOUND then
begin
    select mgrnum into $mgrnum
    from emp2
    where empnum = 100;
    insert into emp3 (empnum, name, deptnum)
    values (100, $name, 15);
```



Tính trong suốt phân tán dùng cho ứng dụng cập nhật

```
insert into emp4 (empnum, sal, tax, mgrnum)  
values (100, $sal, $tax, $mgrnum);  
delete from emp1  
where empnum = 100;  
delete from emp2  
where empnum = 100  
end;
```