CHƯƠNG 4:

TÍNH TRONG SUỐT PHÂN TÁN

Thời lượng: 6 tiết

Giảng viên: ThS. Thái Bảo Trân

TÍNH TRONG SUỐT PHÂN TÁN

- **Tính trong suốt của một hệ phân tán được hiểu như là việc che khuất đi các thành phần riêng biệt của hệ đối với người sử dụng và những người lập trình ứng dụng.**
- **Các loại trong suốt trong hệ phân tán:**

  a. Trong suốt phân đoạn (fragmentation transparency)

  b. Trong suốt về vị trí (location transparency)

  c. Trong suốt ánh xạ địa phương (local mapping transparency)

  d. Không trong suốt (no transparency)

TÍNH TRONG SUỐT PHÂN TÁN

a. Trong suốt phân đoạn *(fragmentation transparency)*:

Khi dữ liệu đã được phân đoạn thì việc truy cập vào CSDL được thực hiện bình thường như là chưa bị phân tán và không ảnh hưởng tới người sử dụng.

Ví dụ: Xét quan hệ tổng thể NCC (Id, Tên, Tuổi)

và các phân đoạn được tách ra từ nó:

NCC1 (Id, Tên, Tuổi)

NCC2 (Id, Tên, Tuổi)

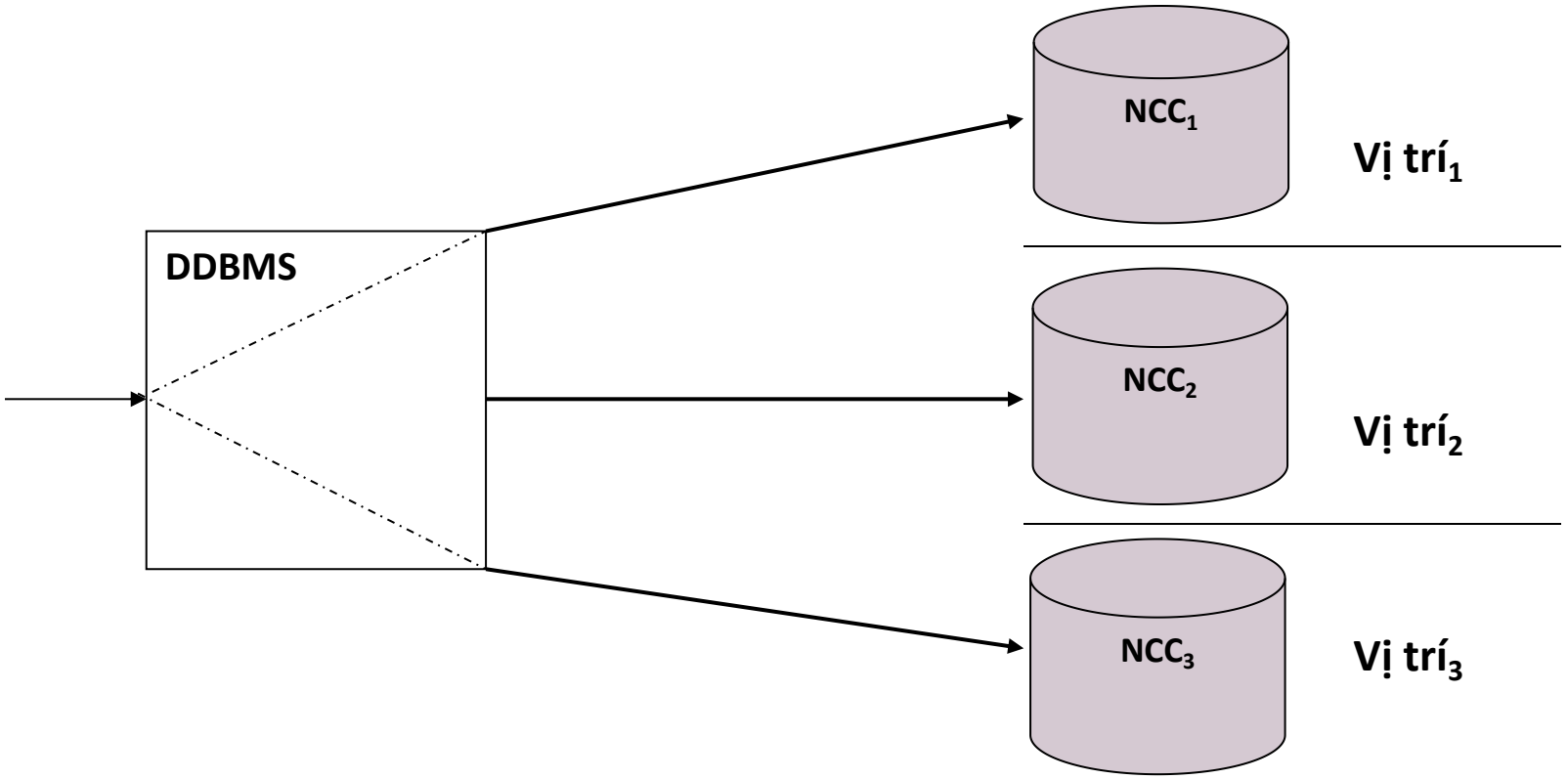
NCC3 (Id, Tên, Tuổi)

Giả sử DDBMS cung cấp tính trong suốt về phân đoạn, khi đó ta có thể thấy tính trong suốt này được thể hiện như sau:

Khi muốn tìm một người có **Id=”Id1“** thì chỉ cần tìm trên quan hệ tổng thể NCC mà không cần biết quan hệ NCC có phân tán hay ko?.

TÍNH TRONG SUỐT PHÂN TÁN

```
SELECT      *
FROM        NCC
WHERE       Id=”Id1”
```



Trong suốt phân đoạn

TÍNH TRONG SUỐT PHÂN TÁN

b.Tính trong suốt về vị trí (*location transparency*):

❖Người sử dụng không cần biết về vị trí vật lý của dữ liệu mà có quyền truy cập đến cơ sở dữ liệu tại bất cứ vị trí nào.

❖Các thao tác để lấy hoặc cập nhật một dữ liệu từ xa được tự động thực hiện bởi hệ thống tại điểm đưa ra yêu cầu.

❖Tính trong suốt về vị trí rất hữu ích, nó cho phép người sử dụng bỏ qua các bản sao dữ liệu đã tồn tại ở mỗi vị trí.

❖ Do đó có thể di chuyển một bản sao dữ liệu từ một vị trí này đến một vị trí khác và cho phép tạo các bản sao mới mà không ảnh hưởng đến các ứng dụng.

TÍNH TRONG SUỐT PHÂN TÁN

Ví dụ: Với quan hệ tổng thể R và các phân đoạn như đã nói ở trên nhưng giả sử rằng DBMS cung cấp trong suốt về vị trí nhưng không cung cấp trong suốt về phân đoạn.

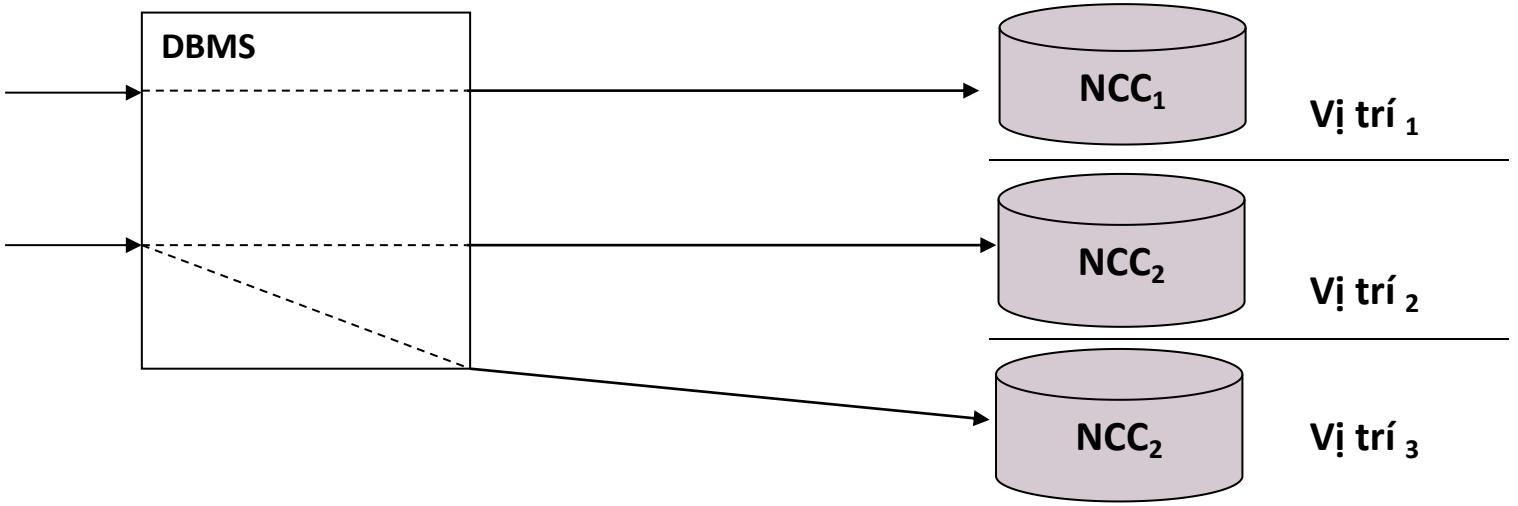
Xét câu truy vấn *tìm người có* **Id=”Id1”**.

```
SELECT      *
FROM        NCC1
WHERE       Id=”Id1”
        IF NOT #FOUND THEN
                SELECT      *
                FROM        NCC2
                WHERE       Id=”Id1”
```

TÍNH TRONG SUỐT PHÂN TÁN

❖ Đầu tiên hệ thống sẽ thực hiện tìm kiếm ở phân đoạn NCC_1 và nếu DBMS trả về biến điều khiển #FOUND thì một câu lệnh truy vấn tương tự được thực hiện trên phân đoạn NCC_2 ,...

❖ Ở đây quan hệ NCC_2 được sao làm hai bản trên hai vị trí$_2$ và vị trí$_3$, ta chỉ cần tìm thông tin trên quan hệ NCC_2 mà không cần quan tâm nó ở vị trí nào.

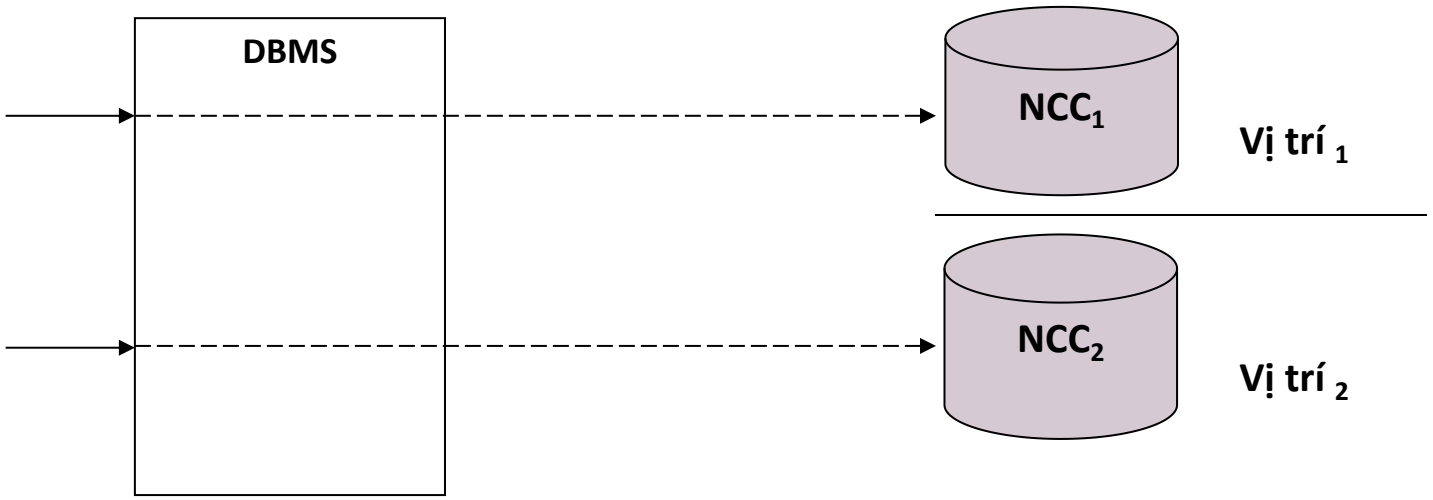


Sự trong suốt về vị trí

TÍNH TRONG SUỐT PHÂN TÁN

c. Trong suốt ánh xạ địa phương ***(local mapping transparency)*****:**

- **Là một đặc tính quan trọng trong một hệ thống DBMS không đồng nhất**
- **Ứng dụng tham chiếu đến các đối tượng có các tên độc lập từ các hệ thống cục bộ địa phương.**
- **Ứng dụng được cài đặt trên một hệ thống không đồng nhất nhưng được sử dụng như một hệ thống đồng nhất.**



Sự trong suốt ánh xạ địa phương

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:**

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

$emp_2 = \sigma_{10 < deptnum \leq 20} \Pi_{empnum, name, mgrnum, deptnum}\ emp$

$emp_3 = \sigma_{deptnum > 20} \Pi_{empnum, name, mgrnum, deptnum}\ emp$

$emp_4 = \Pi_{empnum, name, sal, tax}\ emp$

Example.DDB

- Lược đồ phân mảnh:

$dept_1 = \sigma_{deptnum \leq 10}\ dept$

$dept_2 = \sigma_{10 < deptnum \leq 20}\ dept$

$dept_3 = \sigma_{deptnum > 20}\ dept$

$supplier_1 = \sigma_{city = \text{'SF'}}\ supplier$

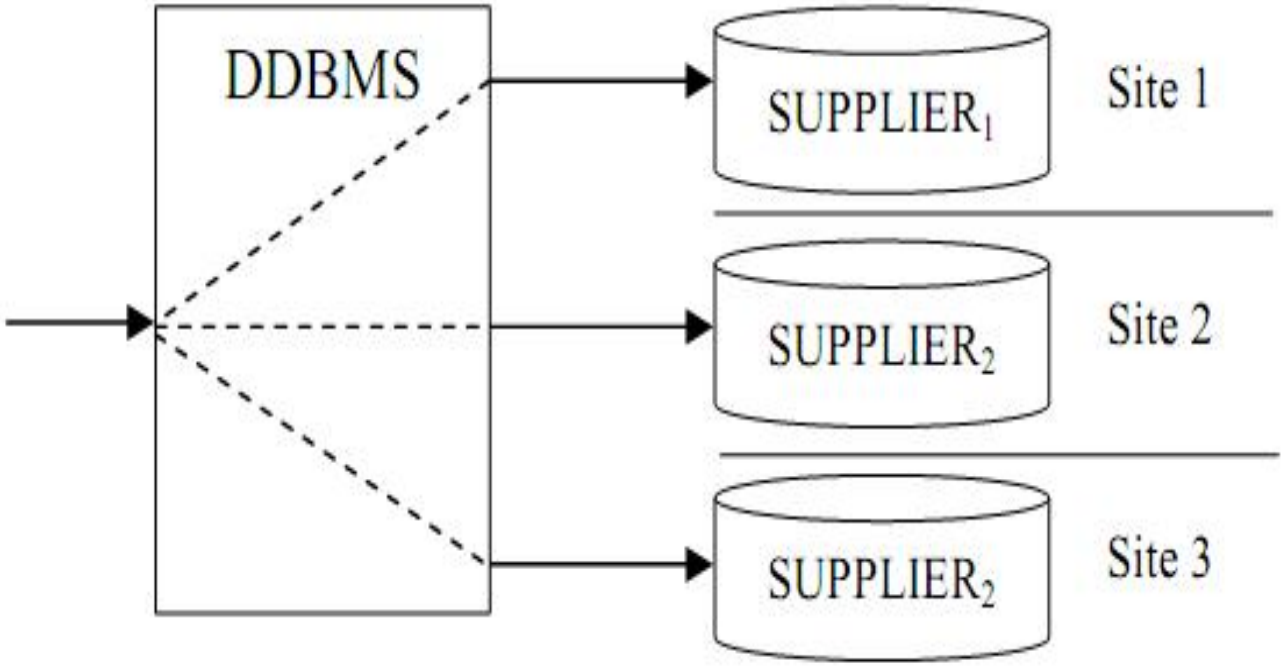
$supplier_2 = \sigma_{city = \text{'LA'}}\ supplier$

$supply_1 = supply \triangleright\!< _{snum = snum}\ supplier_1$

$supply_2 = supply \triangleright\!< _{snum = snum}\ supplier_2$

Tính trong suốt phân tán cho ứng dụng chỉ đọc

Mức 1: Sự trong suốt phân mảnh



Nhận xét: Câu truy vấn mức 1 tương tự như câu truy vấn cục bộ, không cần chỉ ra các phân mảnh cũng như các vị trí cấp phát cho các phân mảnh đó. Khi đó người sử dụng không hề có cảm giác là đang thao tác trên một câu truy vấn phân tán.

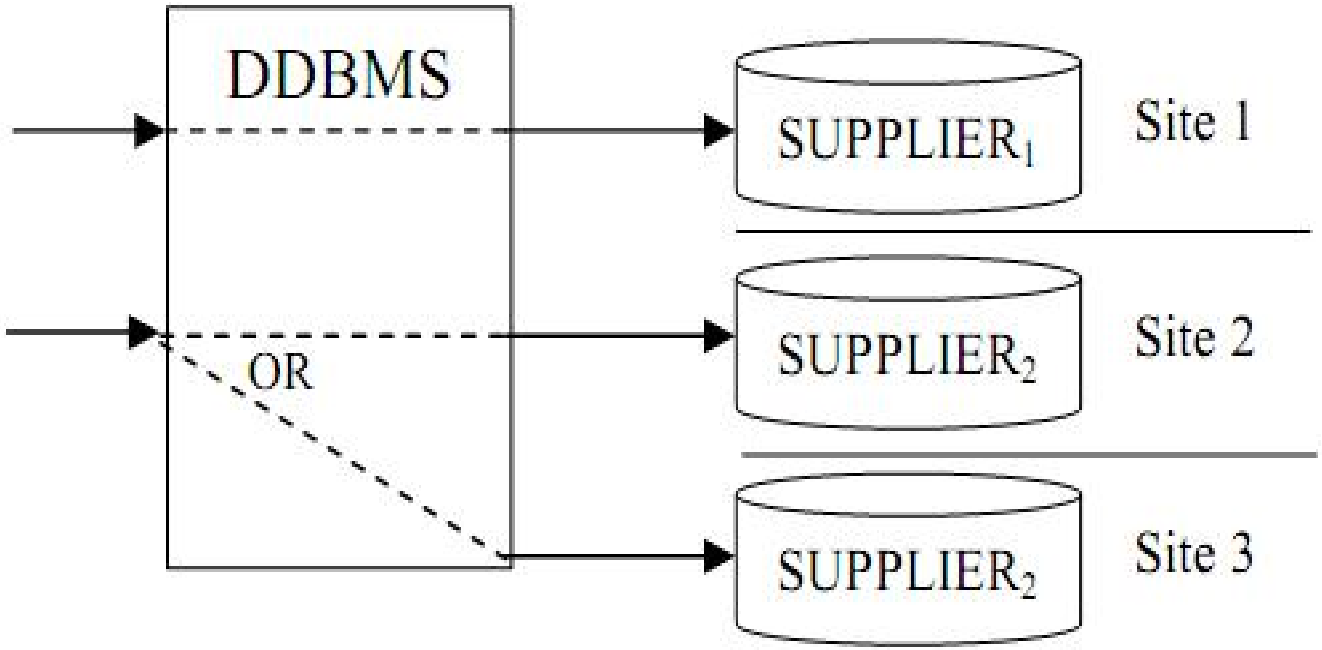
Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

- 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 cho ứng dụng chỉ đọc

Mức 2: Sự trong suốt vị trí



Nhận xét: Người sử dụng phải cung cấp các phân mảnh cụ thể cho câu truy vấn nhưng không cần chỉ ra vị trí của các phân mảnh.

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');
```

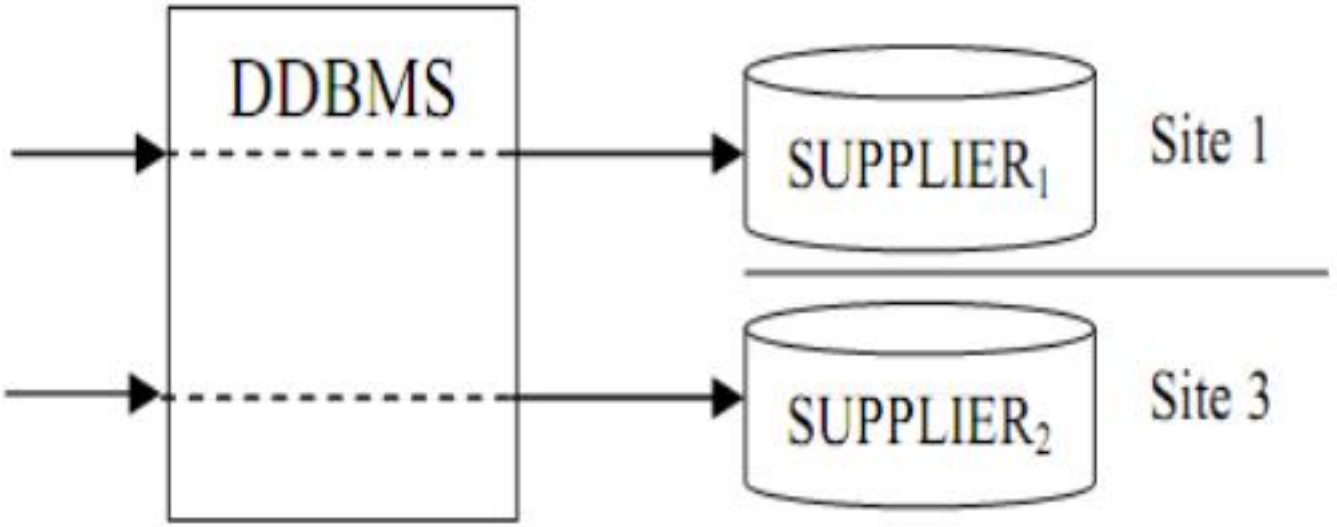
Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc

- 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 cho ứng dụng chỉ đọc

Mức 3: Sự trong suốt ánh xạ cục bộ



Nhận xét: tại mức trong suốt này người sử dụng phải cung cấp các phân mảnh và vị trí cấp phát của chúng.

Tính trong suốt phân tán cho ứng dụng chỉ đọc

- **Mức 3: Sự trong suốt ánh xạ cục bộ**

```
Read (terminal, $SNUM)
   Select NAME into $NAME
   From SUPPLIER1  AT  SITE 1
   Where SNUM = $SNUM
If not #Found then
   Select NAME into $NAME
   From SUPPLIER2  AT  SITE 3
   Where SNUM = $SNUM
Write(terminal, $NAME)
```

Tính trong suốt phân tán cho ứng dụng chỉ đọc

Mức 4: Không trong suốt

```
Read(Terminal,SSUPNUM)
Execute SSUPIMS(SSUPNUM, $FOUND, $NAME) at site 1;
if not $FOUND then
    Execute SSUPCODASYL(SSUPNUM, $FOUND, $NAME)
    at site 3;
Write(Terminal,$NAME)
```

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 cho ứng dụng chỉ đọc

- **Mức 3: Trong suốt ánh xạ cục bộ**

Giả sử các sơ đồ cấp phát các phân mảnh của quan hệ SUPPLY và SUPPLIER như sau:

SUPPLIER1 : Tại site 1

SUPPLIER2 : Tại site 2

SUPPLY1 : Tại site 3

SUPPLY2 : Tại site 4

Tính trong suốt phân tán cho ứng dụng chỉ đọc

- **Mức 3: Trong suốt ánh xạ cục bộ**

```
Read(Terminal, $PNUM)
Select SNUM into $SNUM
from SUPPLY1 at site 3
where SUPPLY1.PNUM = $PNUM
if  #FOUND then
begin
   send $SNUM from site 3 to site 1
   Select NAME into $NAME
   from SUPPLIER1 at site 1
   where SUPPLIER1.SNUM = $SNUM
end
```

Tính trong suốt phân tán cho ứng dụng chỉ đọc

- **Mức 3: Trong suốt ánh xạ cục bộ**

```
else
begin
    Select SNUM into $SNUM
    from SUPPLY2 at site 4
    where SUPPLY2.PNUM = $PNUM
    send $SNUM from site 4 to site 2
    Select NAME into $NAME
    from SUPPLIER2 at site 2
    where SUPPLIER2.SNUM = $SNUM
end;
write(Terminal, $NAME)
```

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.

NHẮC LẠI

- Lược đồ toàn cục:

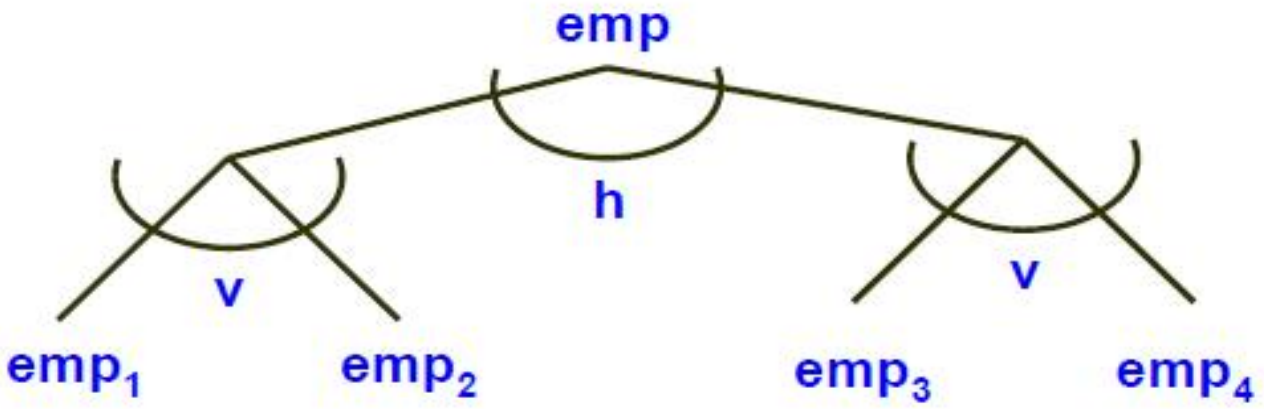
  emp (empnum, name, sal, tax, mgrnum, deptnum)

  dept (deptnum, name, area, mgrnum)

  supplier (snum, name, city)

  supply (snum, pnum, deptnum, quan)

Tính trong suốt phân tán dùng cho ứng dụng cập nhật



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

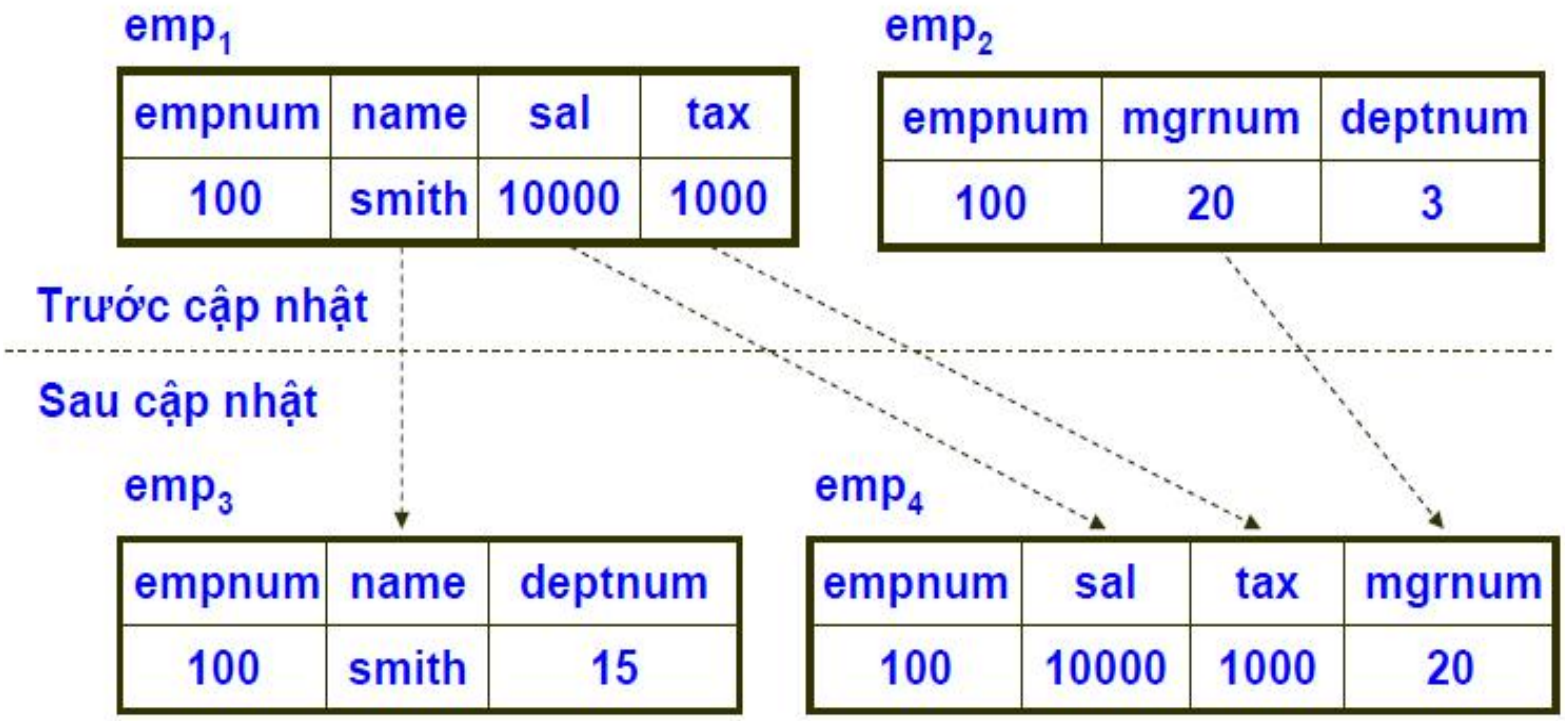
$emp_2 = \Pi_{empnum, mgrnum, deptnum}\ \sigma_{deptnum <= 10}\ emp$

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

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

Cây phân mảnh của quan hệ EMP

Tính trong suốt phân tán dùng cho ứng dụng cập nhật



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;
```

Tính trong suốt phân tán cho ứng dụng cập nhật

- **Mức 3: Trong suốt ánh xạ cục bộ**

Tại mức này ứng dụng phải giải quyết vị trí của các phân mảnh một cách tường minh. Giả sử các phân mảnh của quan hệ EMP được cấp phát như sau:

EMP1 : site 1 và 5
EMP2 : site 2 và 6
EMP3 : site 3 và 7
EMP4 : site 4 và 8

Tính trong suốt phân tán cho ứng dụng cập nhật

- **Mức 3: Trong suốt ánh xạ cục bộ**

```
Select NAME, SAL, TAX into $NAME, $SAL, $TAX
from EMP1 at site 1
where EMPNUM=100;
Select MGRNUM into $MGRNUM
from EMP2 at site 2
where EMPNUM=100;
Insert into EMP3 (EMPNUM, NAME, DEPTNUM) at site 3
Values (100, $NAME, 15);
Insert into EMP3 (EMPNUM, NAME, DEPTNUM) at site 7
Values (100, $NAME, 15);
```

Tính trong suốt phân tán cho ứng dụng cập nhật

- **Mức 3: Trong suốt ánh xạ cục bộ**

Insert into EMP4 (EMPNUM, SAL, TAX, MGRNUM) at site 4
Values (100, $SAL, $TAX, $MGRNUM);
Insert into EMP4 (EMPNUM, SAL, TAX, MGRNUM) at site 8
Values (100, $SAL, $TAX, $MGRNUM);
Delete EMP1 at site 1 where EMPNUM = 100;
Delete EMP1 at site 5 where EMPNUM = 100;
Delete EMP2 at site 2 where EMPNUM = 100;
Delete EMP2 at site 6 where EMPNUM = 100;