

Chương 1: Giới thiệu về cơ sở dữ liệu phân tán

Thời lượng: 6 tiết

Nội dung

2. Định nghĩa CSDL phân tán.
3. Các đặc điểm của CSDL phân tán so với CSDL tập trung.
4. Tại sao sử dụng CSDL phân tán.
5. Hệ quản trị CSDL phân tán.
6. Triển vọng của các hệ cơ sở dữ liệu phân tán.
7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.
8. Một số vấn đề căn bản khi nghiên cứu CSDL phân tán

1. Xử lý phân tán

- ▶ Xử lý phân tán hay còn gọi là hệ thống tính toán phân tán đó là một số **bộ phận xử lý** tự vận hành được liên kết bởi một mạng máy tính và thực hiện các nhiệm vụ mà chúng được phân công.
- ▶ Các bộ phận xử lý là các thiết bị tính toán có thể chạy được một chương trình trên chính nó.

1. Xử lý phân tán (tt)

► Những gì được phân tán:

- Các thiết bị xử lý.
- Chức năng: Nhiều chức năng của hệ thống máy tính có thể được chuyển giao cho các thành phần phần cứng và phần mềm.
- Dữ liệu: Dữ liệu được dùng bởi 1 số ứng dụng có thể được phân tán cho 1 số vị trí xử lý.
- Quyền điều khiển: Quyền điều khiển việc thực hiện 1 số nhiệm vụ cũng có thể được phân tán.

1. Xử lý phân tán (tt)

▶ Phân loại các hệ thống xử lý phân tán:

- Mức độ kết nối
- Sự liên quan giữa các thành phần
- Cấu trúc tương giao
- Sự đồng bộ hóa giữa các thành phần.

▶ Tại sao chúng ta lại thực hiện phân tán ?

- Nhằm thích ứng tốt hơn với việc phân bố rộng rãi của các công ty, xí nghiệp.
- Quan trọng hơn, nhiều ứng dụng hiện tại của công nghệ máy tính cũng được phân tán.

1. Xử lý phân tán (tt)

► Ưu điểm cơ bản việc xử lý phân tán:

- Tận dụng được sức mạnh tính toán bằng cách sử dụng nhiều bộ phận xử lý một cách tối ưu đòi hỏi phải nghiên cứu các hệ thống phân tán và hệ thống xử lý song song.
- Giải quyết bài toán theo từng nhóm hoạt động khá độc lập. Do đó, có thể kiểm soát được chi phí phát triển phần mềm.

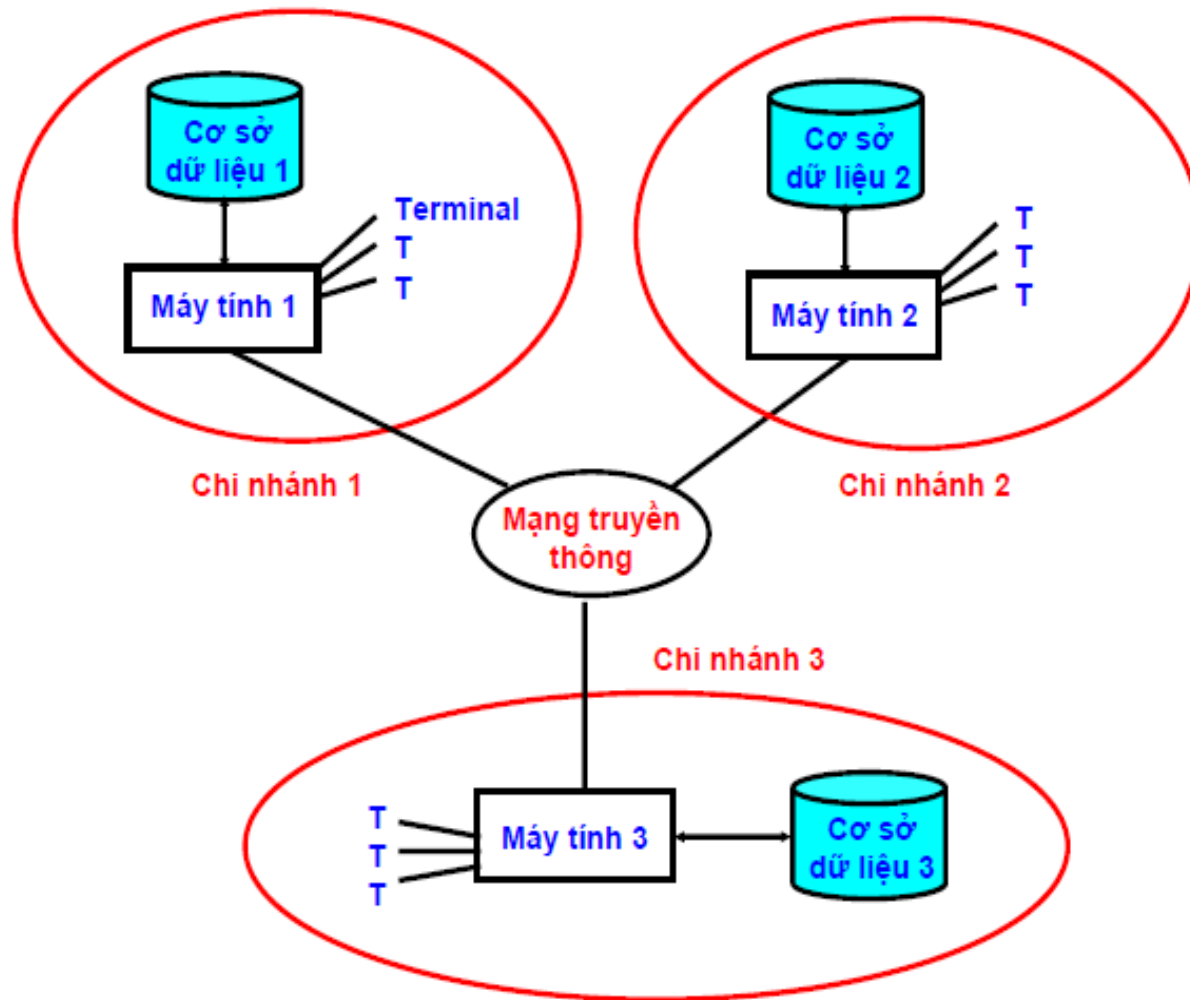
2. Định nghĩa cơ sở dữ liệu phân tán

► Định nghĩa 1:

Cơ sở dữ liệu phân tán (distributed database)

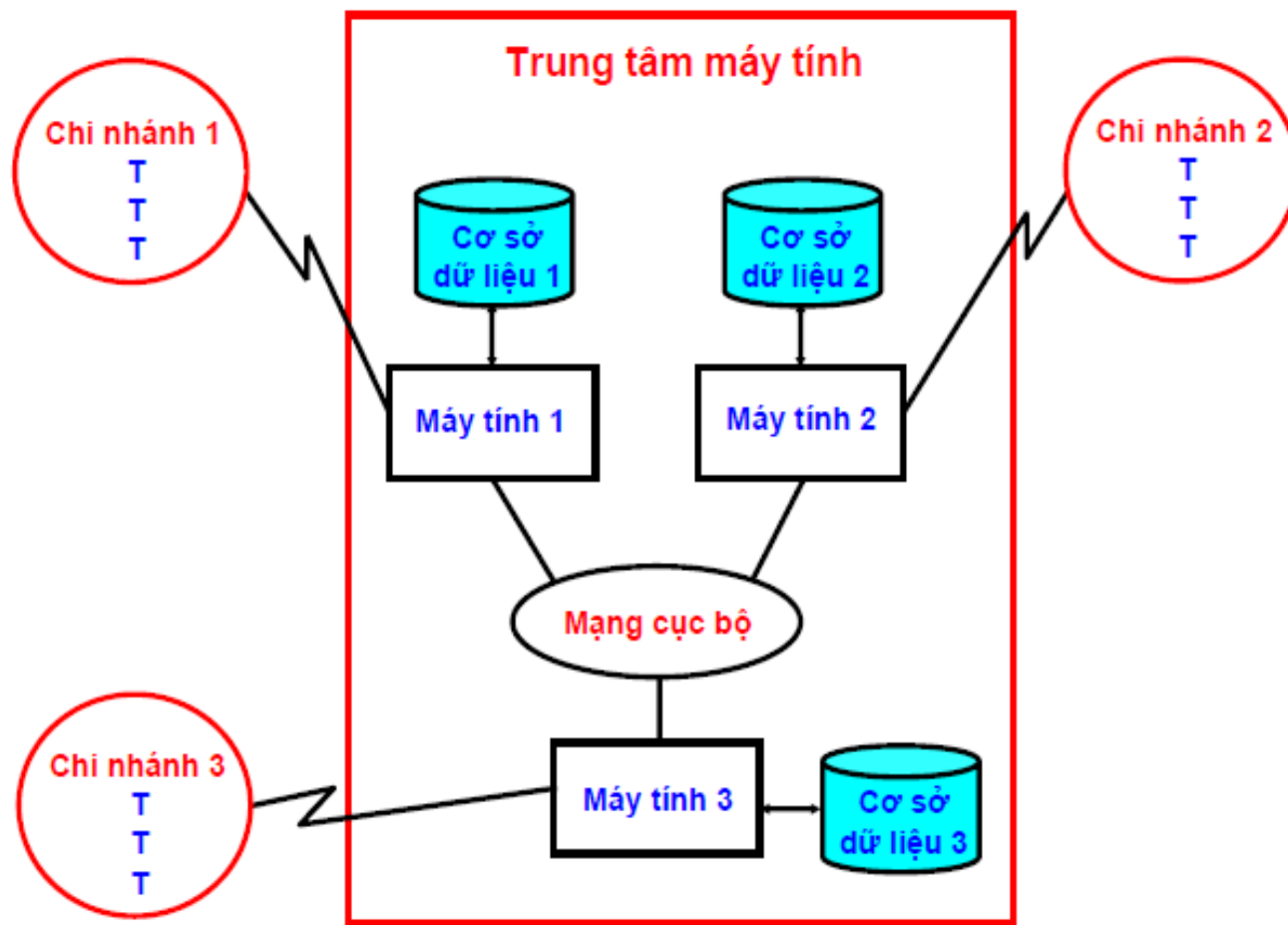
- Là sự tập hợp dữ liệu mà về mặt luận lý chúng thuộc cùng một hệ thống nhưng được đặt ở nhiều nơi (site) của một mạng máy tính.
- **Sự phân tán dữ liệu (data distribution):** dữ liệu phải được phân tán ở nhiều nơi.
- **Sự tương quan luận lý (logical correlation):** dữ liệu của các nơi được sử dụng chung để cùng giải quyết một vấn đề.

2. Định nghĩa cơ sở dữ liệu phân tán



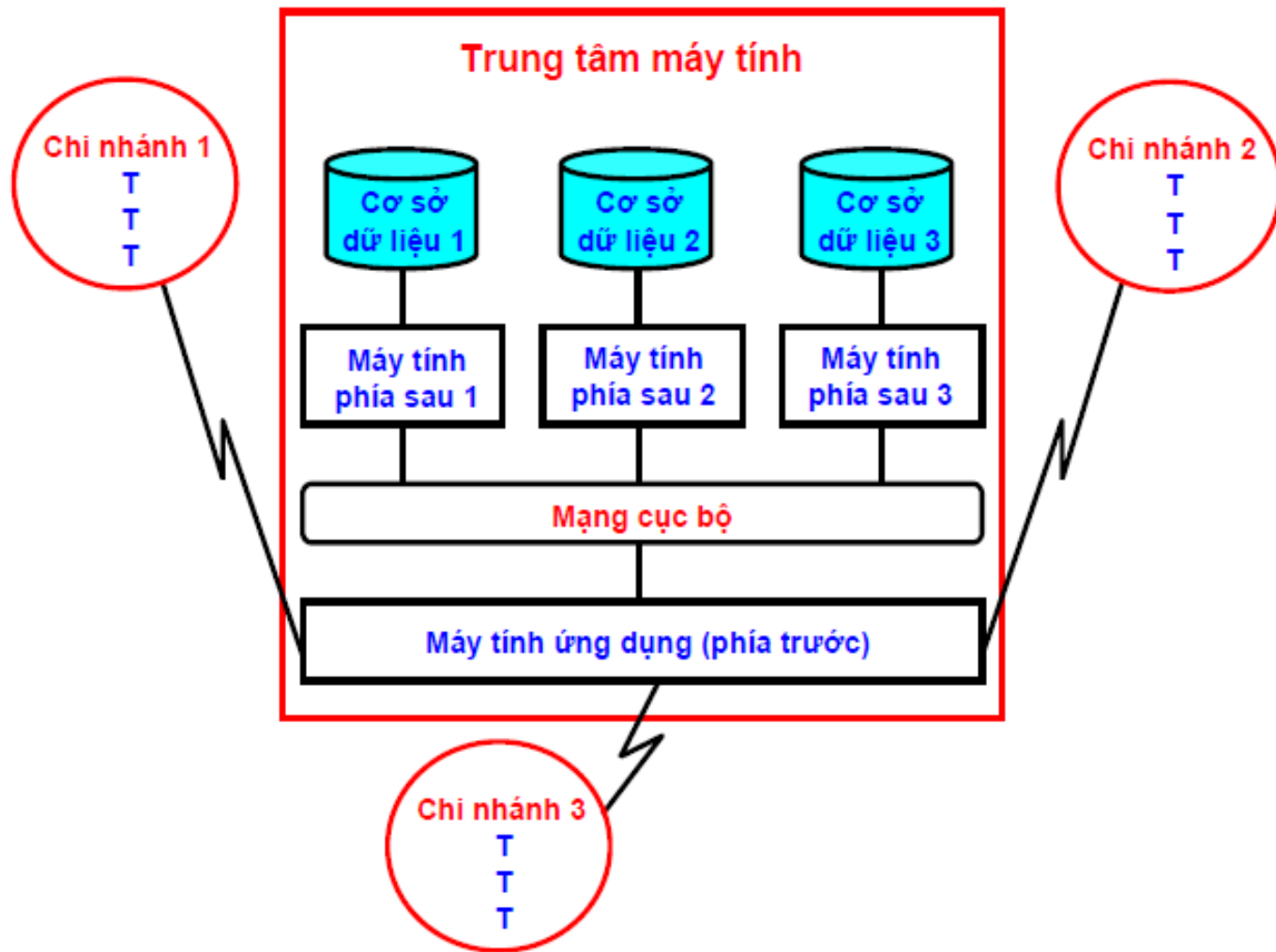
Hình 1.1. Cơ sở dữ liệu phân tán trên một mạng phân tán địa lý.

2. Định nghĩa cơ sở dữ liệu phân tán



Hình 1.2. Cơ sở dữ liệu phân tán trên một mạng cục bộ.

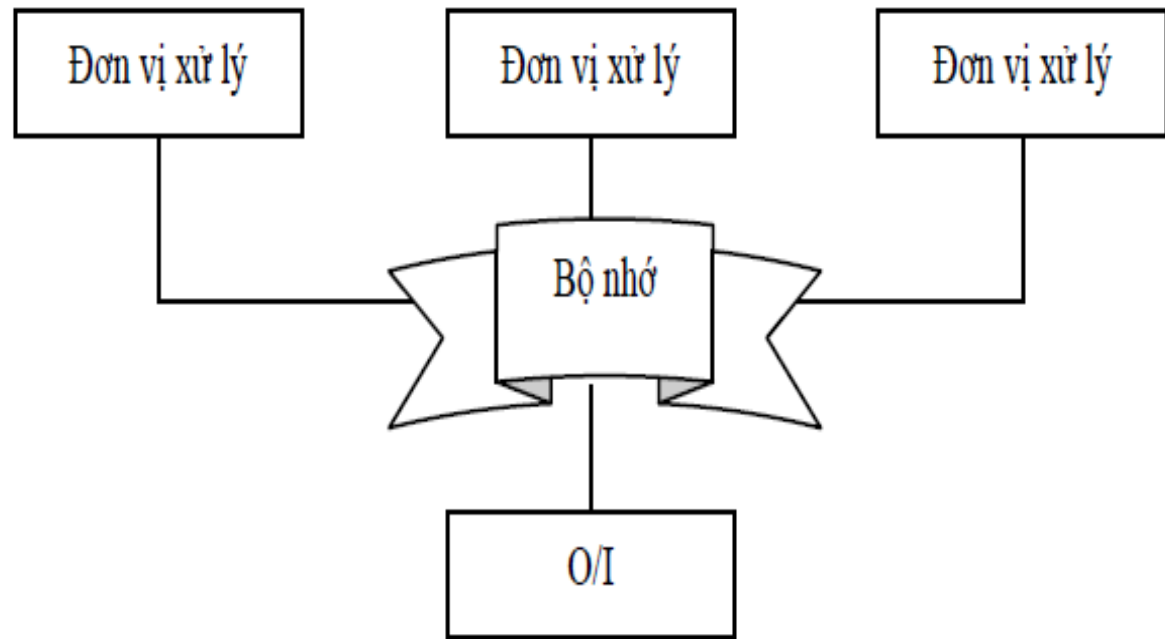
2. Định nghĩa cơ sở dữ liệu phân tán



Hình 1.3. Hệ thống đa xử lý (*multiprocessor system*).

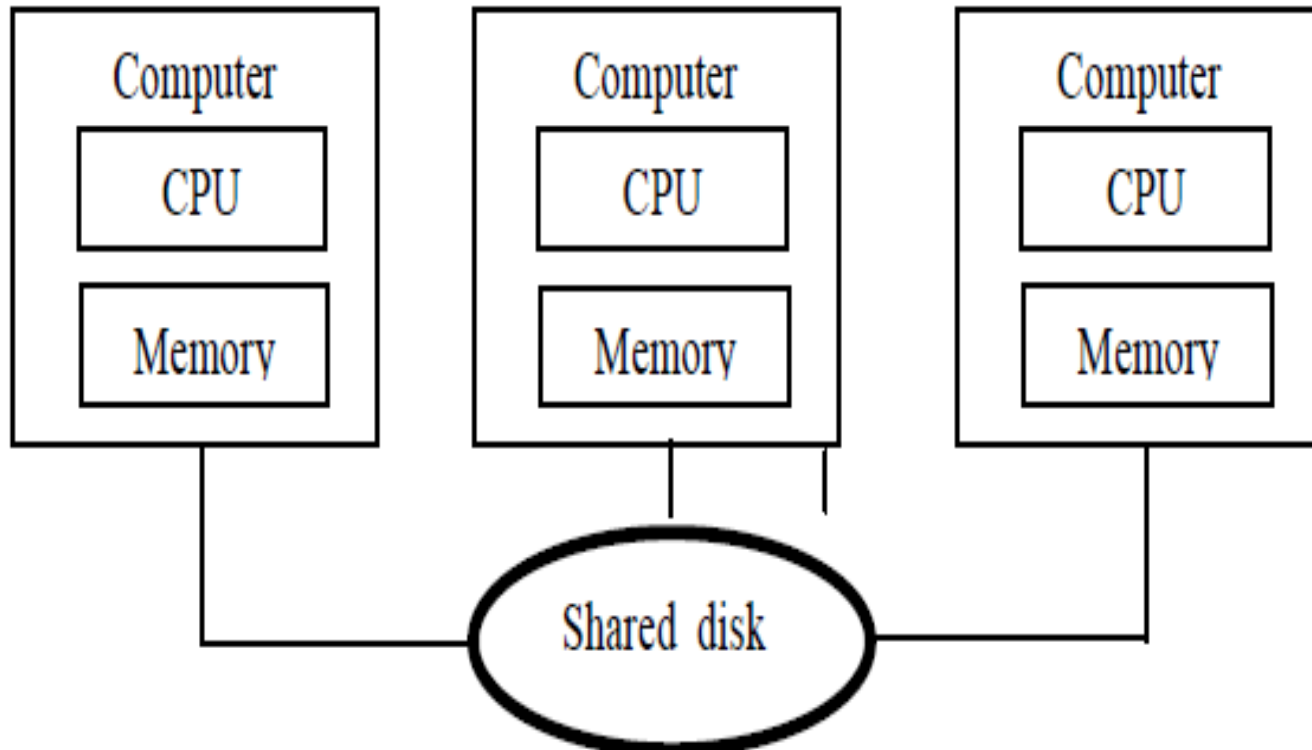
Hệ thống đa xử lý là hệ thống có nhiều đơn vị xử lý cùng dùng chung một dạng bộ nhớ.

Ví dụ 1:



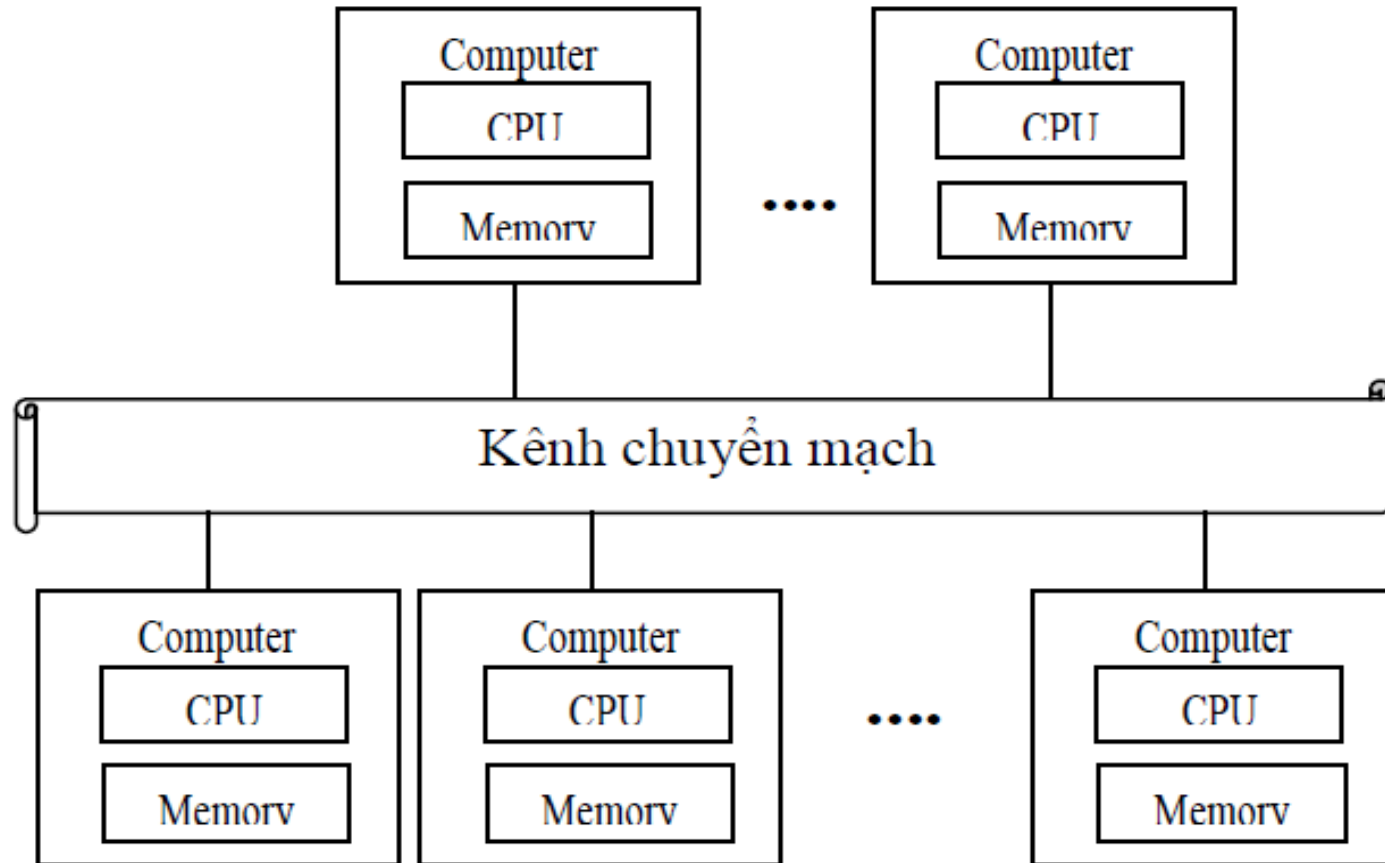
Hệ đa bộ xử lý có bộ nhớ chung

Ví dụ 2:



Hệ đa bộ xử lý có shared disk

Ví dụ 3:



Hệ đa bộ xử lý sở hữu cá nhân

2. Định nghĩa cơ sở dữ liệu phân tán

▶ Định nghĩa 2: Cơ sở dữ liệu phân tán

- Là sự tập hợp dữ liệu được phân tán trên các máy tính khác nhau của một mạng máy tính.
- Mỗi nơi của mạng máy tính có khả năng xử lý tự trị và có thể thực hiện các ứng dụng cục bộ.
- Mỗi nơi cũng tham gia thực hiện ít nhất một ứng dụng toàn cục, mà nơi này yêu cầu truy xuất dữ liệu ở nhiều nơi bằng cách dùng hệ thống truyền thông con.

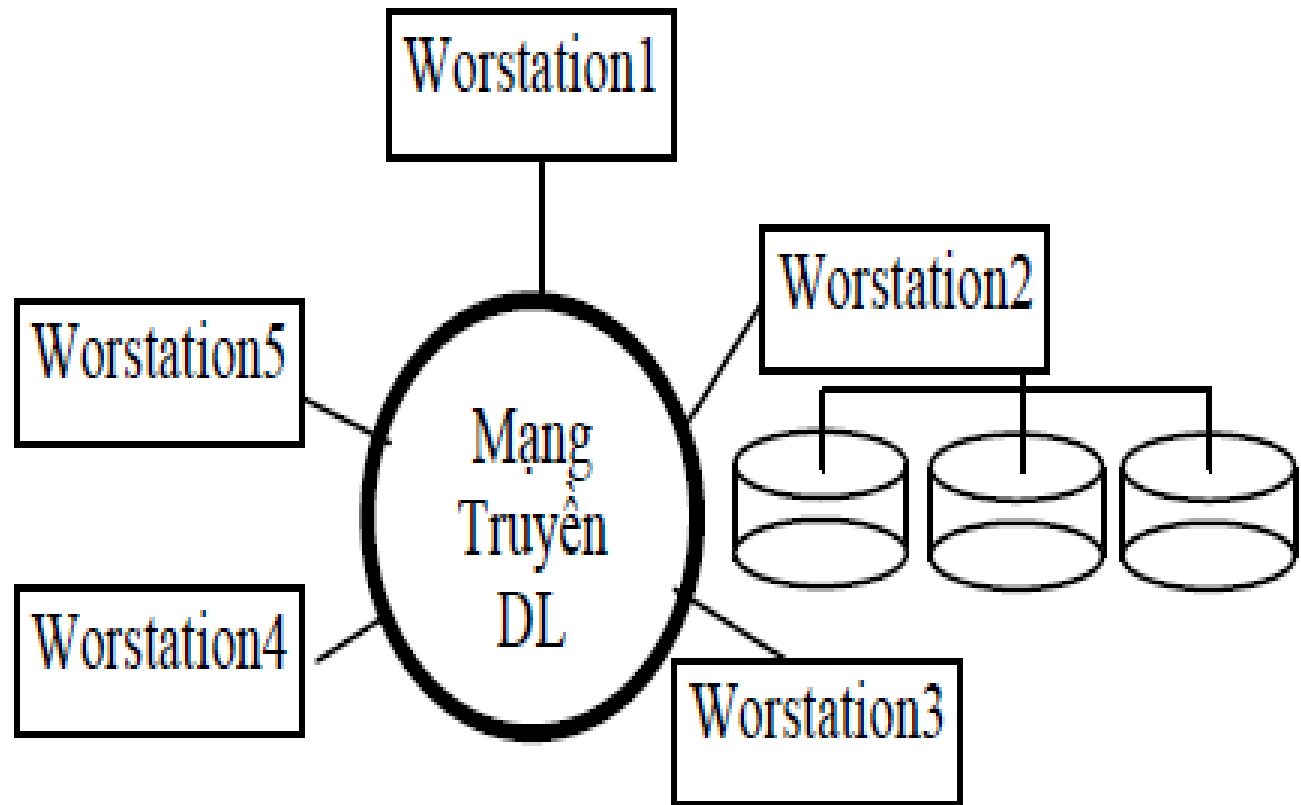
2. Định nghĩa cơ sở dữ liệu phân tán

- ▶ **Sự phân tán dữ liệu (data distribution):** dữ liệu phải được phân tán ở nhiều nơi.
- ▶ **Ứng dụng cục bộ (local application):** ứng dụng được chạy hoàn thành tại một nơi và chỉ sử dụng dữ liệu cục bộ của nơi này.
- ▶ **Ứng dụng toàn cục (hoặc ứng dụng phân tán) (global application / distributed application):** ứng dụng được chạy hoàn thành và sử dụng dữ liệu của ít nhất hai nơi.

Nhận xét

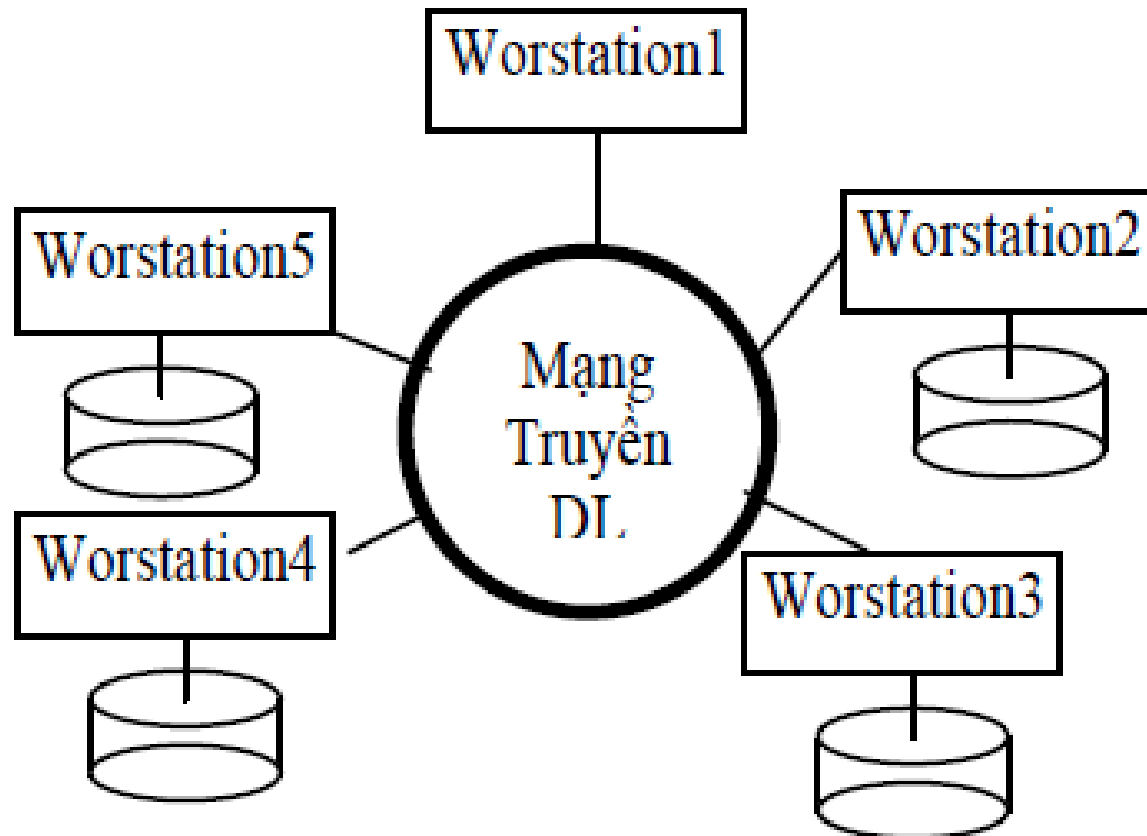
- ▶ Nếu CSDL nằm tại một nút mạng thì nó không phải là DDBS, vì vấn đề quản trị CSDL không khác với quản trị CSDL trong môi trường tập trung kiểu client/server của mạng. (Xem ví dụ 4)
- ▶ Nếu cơ sở dữ liệu được phân tán trên nhiều nút mạng, khi đó CSDL sẽ là cơ sở dữ liệu phân tán. (Xem ví dụ 5)

Ví Dụ 4:



CSDL tập trung, không phải DDBS

Ví Dụ 5:



CSDL được phân tán trên mạng, DDBS

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

- Các vấn đề của hệ CSDL phân tán phức tạp hơn so với các tập trung.
- Các cấu trúc vật lý phức tạp và truy xuất hiệu quả
 - Cấu trúc vật lý phức tạp để truy xuất hiệu quả.
 - Tối ưu hóa (optimization)
 - *Tối ưu hóa toàn cục (global optimization)*
 - *Tối ưu hóa cục bộ (local optimization)*

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

- Các hệ phân tán đòi hỏi phải có thêm các thiết bị mới (thiết bị truyền thông chẳng hạn) và như thế làm tăng chi phí phần cứng.
- Thành phần chi phí quan trọng nhất là chi phí về nhân lực. Khi các thiết bị máy tính được xây dựng ở nhiều vị trí khác nhau, chúng đòi hỏi phải có con người điều hành và quản lý.

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

- Điểm này đã được nêu ra trước đây như một ưu điểm của các hệ CSDL phân tán. Không may là sự phân tán lại gây ra các vấn đề đồng bộ hóa dữ liệu làm tăng tính phức tạp.
 - Việc điều khiển phân tán có thể trở thành một gánh nặng nếu không có những chiến lược phù hợp để giải quyết chúng.
- ▶ **Điều khiển tập trung**
- Điều khiển tập trung (*centralized control*)
 - Người quản trị CSDL cục bộ (*local DBA*)
 - Người quản trị CSDL toàn cục (*global DBA*)
 - Tính tự trị vị trí (*site autonomy*)

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

- Một trong những lợi ích chính của các CSDL tập trung là chúng bảo đảm kiểm soát được các truy xuất dữ liệu.
- Vấn đề an ninh trong các hệ CSDL phân tán rõ ràng là phức tạp hơn so với các hệ tập trung.
 - *Thực hiện truy xuất dữ liệu có thẩm quyền.*
 - *Bảo mật CSDL cục bộ.*
 - *Bảo mật mạng truyền thông.*

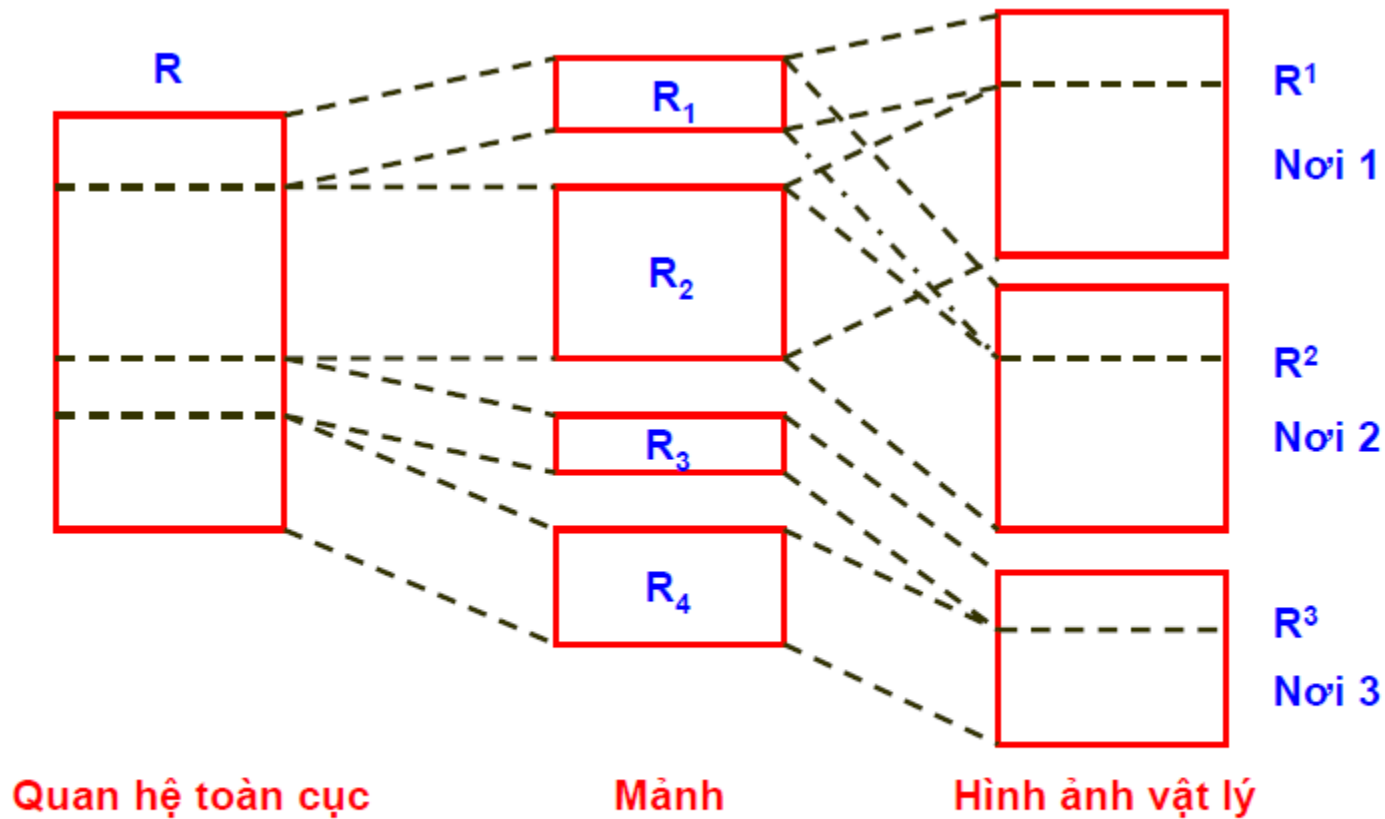
3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

- Độc lập dữ liệu (*data independence*)
- Tính trong suốt dữ liệu (*data transparency*)
- Trong suốt phân mảnh (*fragmentation transparency*):
 - Không nhìn thấy các mảnh.
 - Nhìn thấy các quan hệ toàn cục (*global relation*).
 - 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*).
 - Lược đồ phân mảnh (*fragmentation schema*).

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

- 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.*
- Trong suốt phân tán (distribution transparency) gồm bốn tính trong suốt trên.

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung



Hình 1.4. Các mảnh và các hình ảnh vật lý của một quan hệ toàn cục.

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

▶ Giảm dư thừa dữ liệu

- Dư thừa dữ liệu (*data redundancy*)
- Nhược điểm của dư thừa dữ liệu
 - Không nhất quán dữ liệu (*data inconsistency*).
 - Tốn nhiều vùng nhớ lưu trữ.
- Ưu điểm của dư thừa dữ liệu
 - Tính cục bộ (*locality*) của ứng dụng cao.
 - Tính sẵn sàng của dữ liệu (*data availability*) cao.
- Nhân bản dữ liệu (*data replication*): dữ liệu được lưu trữ thành nhiều bản.
 - Ứng dụng chỉ đọc (*read-only application*)
 - Ứng dụng cập nhật (*update application*)

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

▶ Tính toàn vẹn (Integrity)

- Giao tác (transaction)

- *Giao tác là một đơn vị thực hiện nguyên tử.*
- *Một chuỗi các tác vụ mà tất cả các tác vụ này đều được thực hiện hoặc đều không được thực hiện.*

- Giao tác toàn cục (global transaction)

- *Giao tác toàn cục là một ứng dụng toàn cục.*

- Tính nguyên tử (atomicity)

- *Sự hư hỏng*
- *Tính đồng thời*

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

► Độc lập dữ liệu

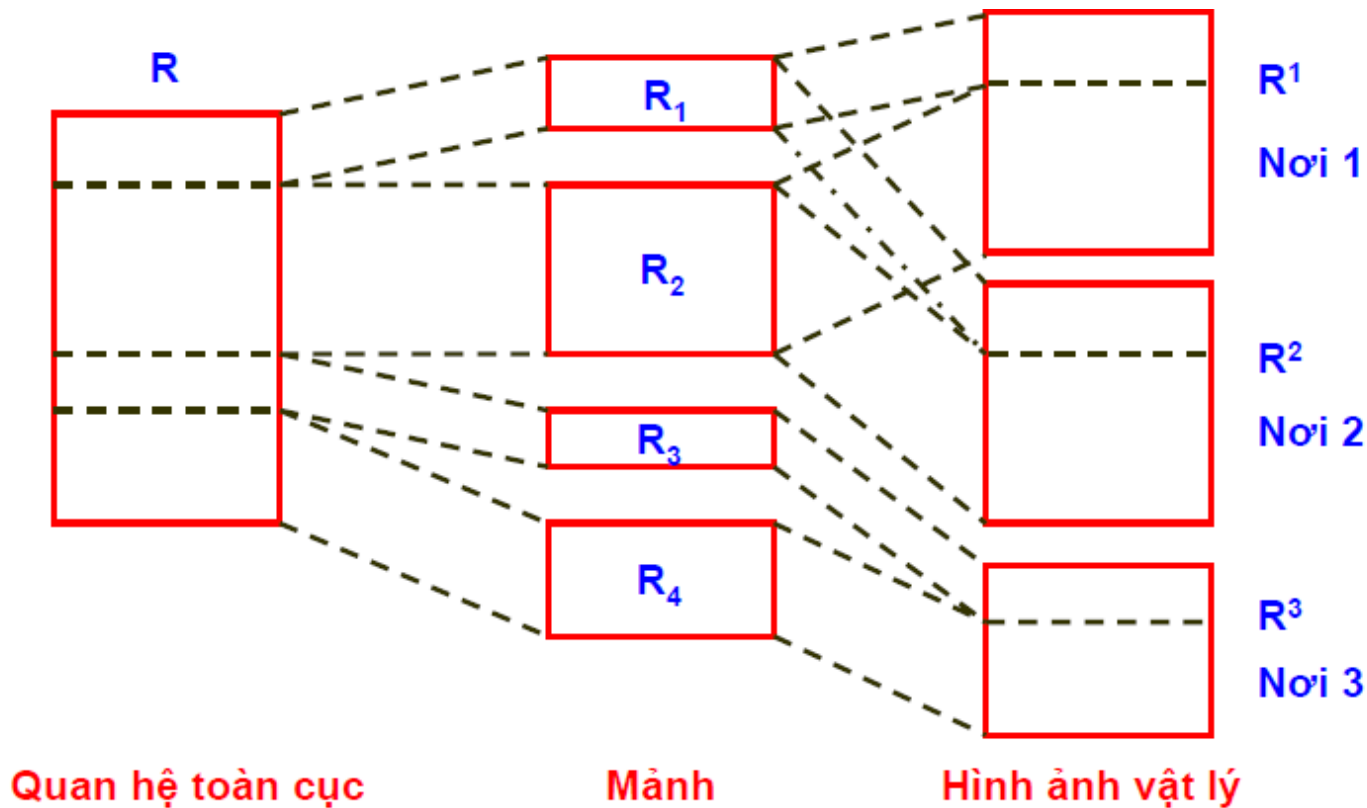
- Độc lập dữ liệu (*data independence*)
- Tính trong suốt dữ liệu (*data transparency*)
- Trong suốt phân mảnh (*fragmentation transparency*)
 - *Không nhìn thấy các mảnh.*
 - *Nhìn thấy các quan hệ toàn cục (global relation).*
 - *Lược đồ toàn cục (global schema).*
- Trong suốt vị trí (*location transparency*)
 - *Không nhìn thấy các quan hệ cục bộ.*
 - *Nhìn thấy các mảnh (fragment).*
 - *Lược đồ phân mảnh (fragmentation schema).*

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

▶ Độc lập dữ liệu

- 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.*
- Trong suốt phân tán (distribution transparency) gồm bốn tính trong suốt trên.

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung



Hình 1.4. Các mảnh và các hình ảnh vật lý của một quan hệ toàn cục.

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

▶ Giảm dư thừa dữ liệu

- Dư thừa dữ liệu (*data redundancy*)
- Nhược điểm của dư thừa dữ liệu
 - Không nhất quán dữ liệu (*data inconsistency*).
 - Tốn nhiều vùng nhớ lưu trữ.
- Ưu điểm của dư thừa dữ liệu
 - Tính cục bộ (*locality*) của ứng dụng cao.
 - Tính sẵn sàng của dữ liệu (*data availability*) cao.
- Nhân bản dữ liệu (*data replication*): *dữ liệu được lưu trữ thành nhiều bản.*
 - Ứng dụng chỉ đọc (*read-only application*)
 - Ứng dụng cập nhật (*update application*)

3. Các đặc điểm của CSDL phân tán so với CSDL tập trung

► Tính toàn vẹn

- Integrity
- Giao tác (transaction)
 - *Giao tác là một đơn vị thực hiện nguyên tố.*
 - *Một chuỗi các tác vụ mà tất cả các tác vụ này đều được thực hiện hoặc đều không được thực hiện.*
- Giao tác toàn cục (global transaction)
 - *Giao tác toàn cục là một ứng dụng toàn cục.*
- Tính nguyên tố (atomicity)
 - *Sự hư hỏng*
 - *Tính đồng thời*

4. Tại sao sử dụng cơ sở dữ liệu phân tán

- ▶ Các lý do về tổ chức và về kinh tế
- ▶ Nhiều tổ chức không được tập trung hóa.
 - Các CSDL hiện tại cần kết nối với nhau
- ▶ Nhiều CSDL đã tồn tại trong một công ty và cần phải thực hiện nhiều ứng dụng toàn cục hơn.
 - Sự lớn mạnh gia tăng
- ▶ Có thêm các đơn vị tổ chức tương đối độc lập.
 - Giảm chi phí truyền thông
- ▶ Nhiều ứng dụng cục bộ làm giảm chi phí truyền thông so với CSDL tập trung.

4. Tại sao sử dụng cơ sở dữ liệu phân tán

- ▶ **Các nghiên cứu về hiệu suất**
 - *Hiệu suất được nâng cao bằng một cơ chế song song hóa.*
 - *Phân mảnh dữ liệu theo ứng dụng, làm cực đại hóa tính cục bộ của ứng dụng.*
- ▶ **Độ tin cậy và tính sẵn sàng**
 - *Vì dư thừa dữ liệu, tính sẵn sàng của dữ liệu (data availability) cao.*
 - *Cần phải bảo đảm độ tin cậy của dữ liệu (data reliability).*

5. Hệ quản trị CSDL phân tán (DDBMS)

- ▶ **Hệ CSDL phân tán – DDB System (DDBS):** là một tập hợp dữ liệu có liên hệ logic và được phân bố trên các nút của một mạng máy tính.
- ▶ **Hệ quản trị CSDL phân tán - DDBS Management System (DDBMS):** một hệ thống phần mềm cho phép quản lý các DDBS và làm cho việc phân tán trở nên vô hình đối với người sử dụng.

5. Hệ quản trị CSDL phân tán (DDBMS)

▶ Các thành phần của DDBMS

■ Truyền thông dữ liệu

- *DC–Data Communication*
- *Nhận yêu cầu truy xuất dữ liệu của ứng dụng chạy tại thiết bị đầu cuối.*
- *Trả kết quả về cho ứng dụng.*

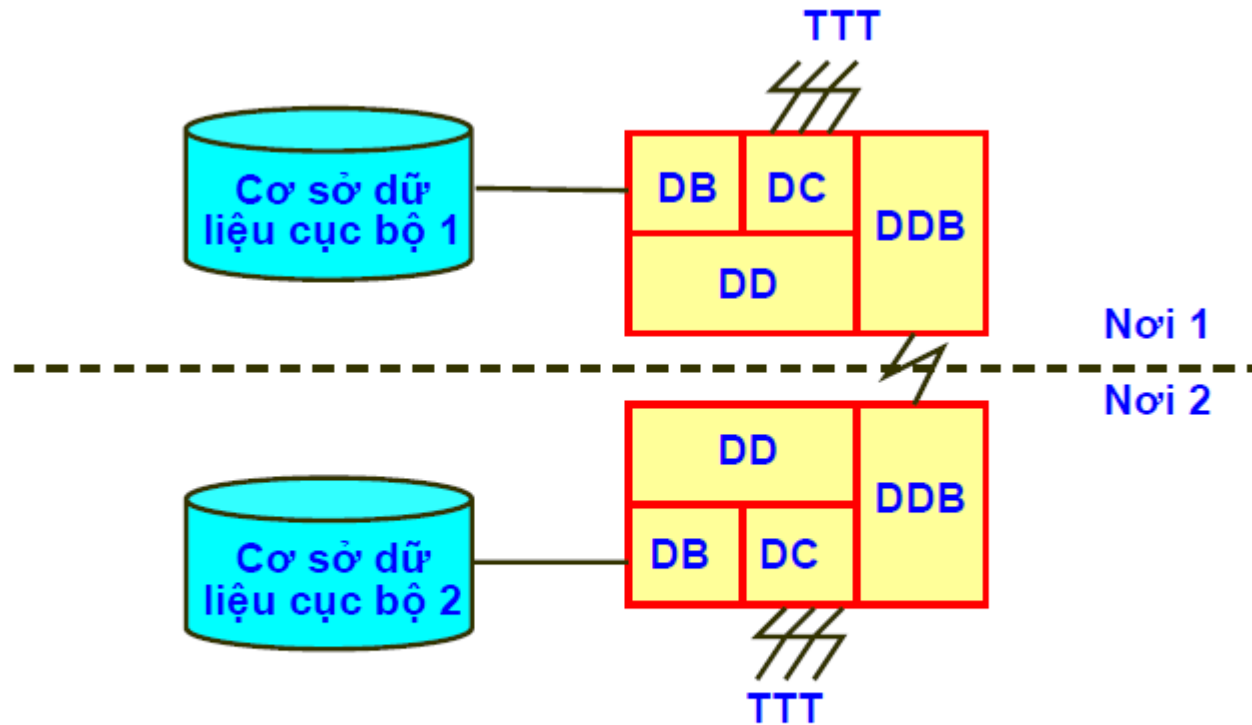
■ Quản trị CSDL

- *DB –DataBase management*
- *Quản lý CSDL.*
- *Thực hiện các yêu cầu của ứng dụng: xử lý dữ liệu (data processing).*

5. Hệ quản trị CSDL phân tán (DDBMS)

- ▶ **Các thành phần của DDBMS (tt)**
 - **Từ điển dữ liệu**
 - *DD –Data Dictionary*
 - *Lưu trữ thông tin về các đối tượng dữ liệu trong CSDL.*
 - *Lưu trữ thông tin về sự phân tán dữ liệu tại các nơi.*
 - **CSDL phân tán**
 - *DDB –Distributed DataBase*
 - *Liên lạc giữa các nơi: gửi yêu cầu và nhận kết quả.*

5. Hệ quản trị CSDL phân tán (DDBMS)



Hình 1.6. Các thành phần của DDBMS thương mại.

5. Hệ quản trị CSDL phân tán (DDBMS)

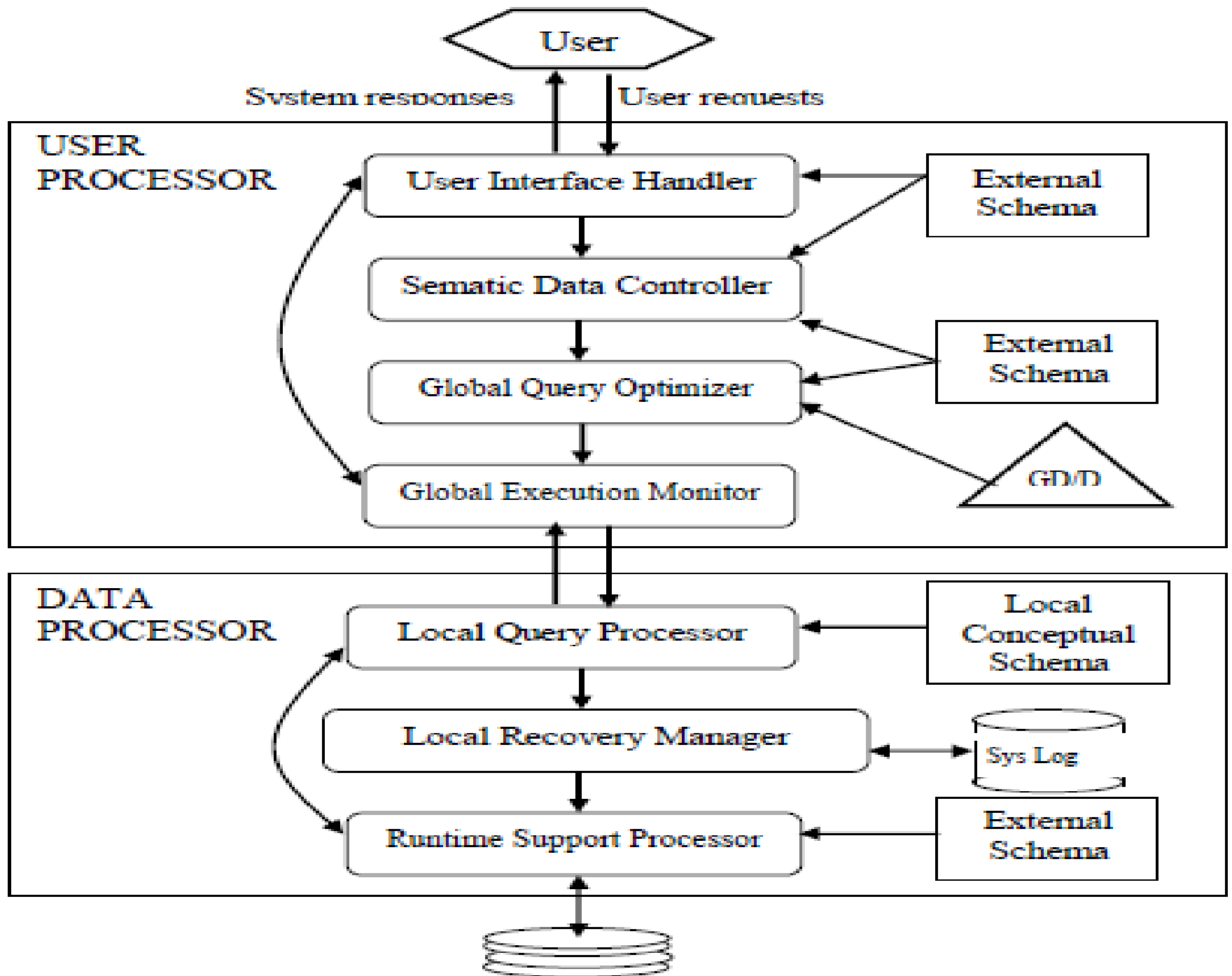
Các thành phần của một DBMS phân tán

❖ Bộ phận tiếp nhận người dùng (user processor)

- ▶ Bộ phận giao tiếp (User Interface Handler)
- ▶ Bộ phận kiểm soát ngữ nghĩa. (Semanticdata controller)
- ▶ Bộ phận phân rã và tối ưu vấn tin toàn cục (global query optimier and Decomposer).
- ▶ Bộ phận theo dõi việc thực hiện phân tán (Distributed Execution monitor)

❖ Bộ xử lý (database processor)

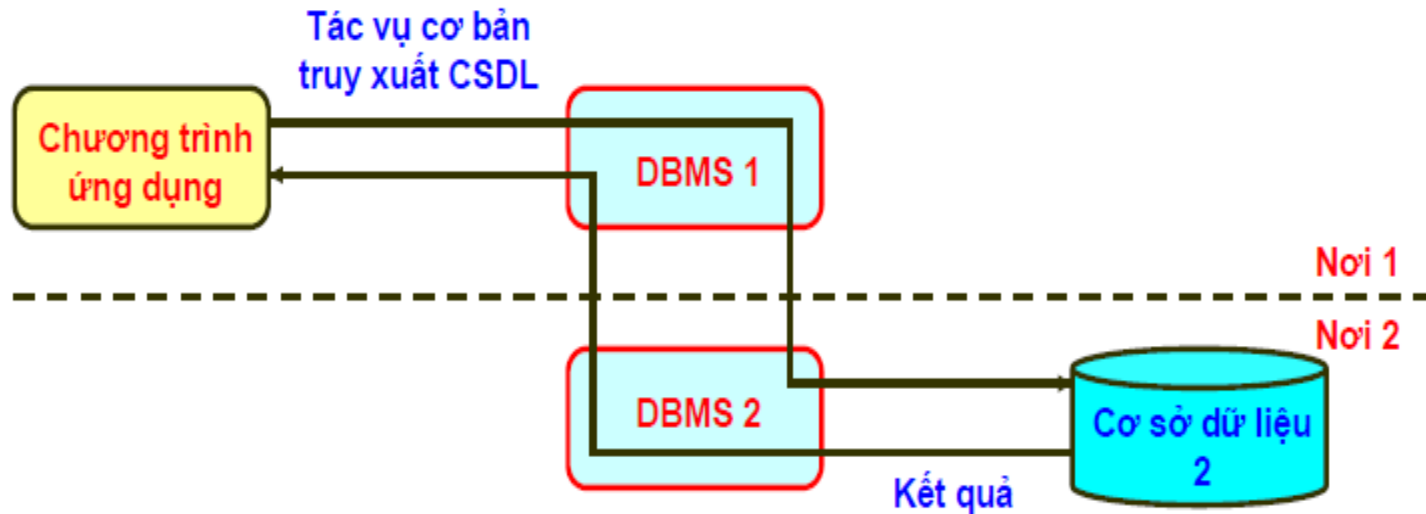
- ▶ Bộ phận tối ưu hóa vấn tin cục bộ (local query optimijer)
- ▶ Bộ phận khôi phục cục bộ (Local recovery manager)
- ▶ Bộ phận hỗ trợ xử lý thực thi (Runtime. Support processor).



5. Hệ quản trị CSDL phân tán (DDBMS)

- ▶ **Các chức năng tiêu biểu của DDBMS**
 - Truy xuất CSDL từ xa.
 - Hỗ trợ một số mức trong suốt phân tán.
 - Hỗ trợ cho việc quản trị CSDL phân tán.
 - Hỗ trợ cho việc điều khiển tương tranh và phục hồi các giao tác phân tán.

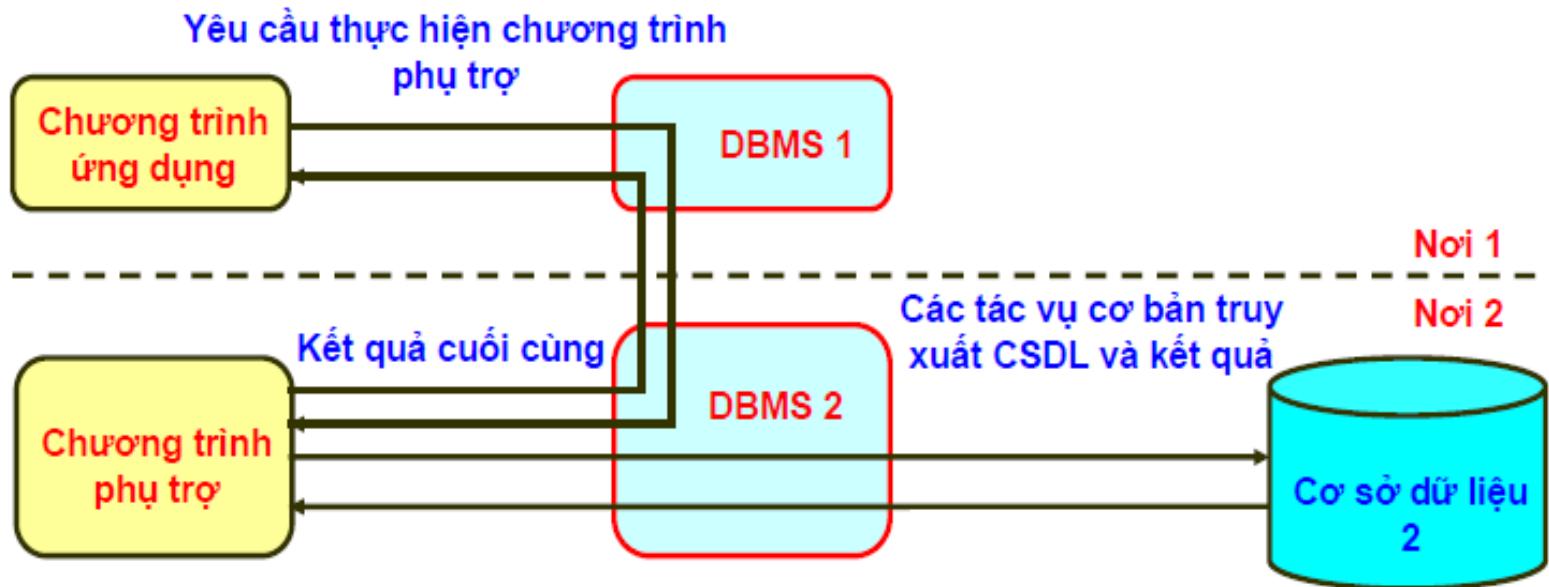
5. Hệ quản trị CSDL phân tán (DDBMS)



a. Truy xuất từ xa thông qua các tác vụ cơ bản của DBMS

Hình 1.7. Các loại truy xuất cơ sở dữ liệu phân tán.

5. Hệ quản trị CSDL phân tán (DDBMS)



b. Truy xuất từ xa thông qua chương trình phụ trợ

Hình 1.7. Các loại truy xuất cơ sở dữ liệu phân tán.

5. Hệ quản trị CSDL phân tán (DDBMS)

- ▶ **Tính đồng nhất và tính không đồng nhất**
 - Homogeneity, heterogeneity
 - Phần cứng (hardware)
 - Hệ điều hành (operating system)
 - Các DBMS cục bộ
- ▶ **DDBMS đồng nhất**
 - *Các DBMS cục bộ giống nhau.*
- ▶ **DDBMS không đồng nhất**
 - Có ít nhất hai DBMS cục bộ khác nhau.
 - Chuyển đổi các mô hình dữ liệu khác nhau.

6. Triển vọng của các hệ cơ sở dữ liệu phân tán

- ▶ Từ lý do xã hội của việc phi tập trung đến tính hiệu quả kinh tế của CSDL phân tán người ta phân DDBS thành các nhóm triển vọng sau:
 - ❑ *Quản lý dữ liệu phân tán và nhân bản vô hình.*
 - ❑ *Yêu cầu độ tin cậy qua các giao dịch phân tán*
 - ❑ *Nâng cao hiệu năng*

6. Triển vọng của các hệ cơ sở dữ liệu phân tán

❑ Quản lý dữ liệu phân tán và nhân bản vô hình.

- Một hệ thống vô hình là cách cho ẩn đi thao tác cài đặt, cài mà người sử dụng không cần quan tâm đến.
- **Ví dụ:** Một công ty Điện - Toán, có các văn phòng ở Boston , Edmonton, Paris và San Francisco, có một số dự án (project) được thực hiện tại các địa điểm đó, và muốn dùng CSDL để quản lý nhân công (Employee), quản lý dự án và các dữ liệu liên quan khác. Giả sử CSDL là CSDL quan hệ - Relational database (RDB) có thể lưu các bảng sau:

6. Triển vọng của các hệ cơ sở dữ liệu phân tán

EMP(ENO, ENAME, TITLE)

PROJ(PN_o, PNAME, BUDGET)

PAY(TITLE, SAL)

ASG (ENO, PNO, DUR, RESP)

- ▶ Nếu dữ liệu được lưu ở hệ CSDL tập trung và nếu muốn có danh sách tên và lương của các nhân viên đã làm dự án nào đó trên 12 tháng thì câu lệnh truy vấn SQL sẽ là:

SELECT ENAME, SAL

FROM EMP, AGG , PAY

WHERE ASG.DUR > 12 AND EMP.ENO = ASG.ENO

AND PAY.TITLE = EMP.TITLE

▶ **Phân mảnh CSDL (Fragmentation)**

- Vì công ty được phân tán các vị trí khác nhau (Boston, Edmonton, Paris và San Francisco), nên công ty muốn để dữ liệu về các nhân viên các dự án của vị trí nào được lưu ở vị trí đó.
- Do đó cần phải phân hoạch các quan hệ EMP và PROJ và lưu chúng tại các vị trí như đã yêu cầu. Quá trình làm này được gọi là **phân mảnh (fragmentation)**

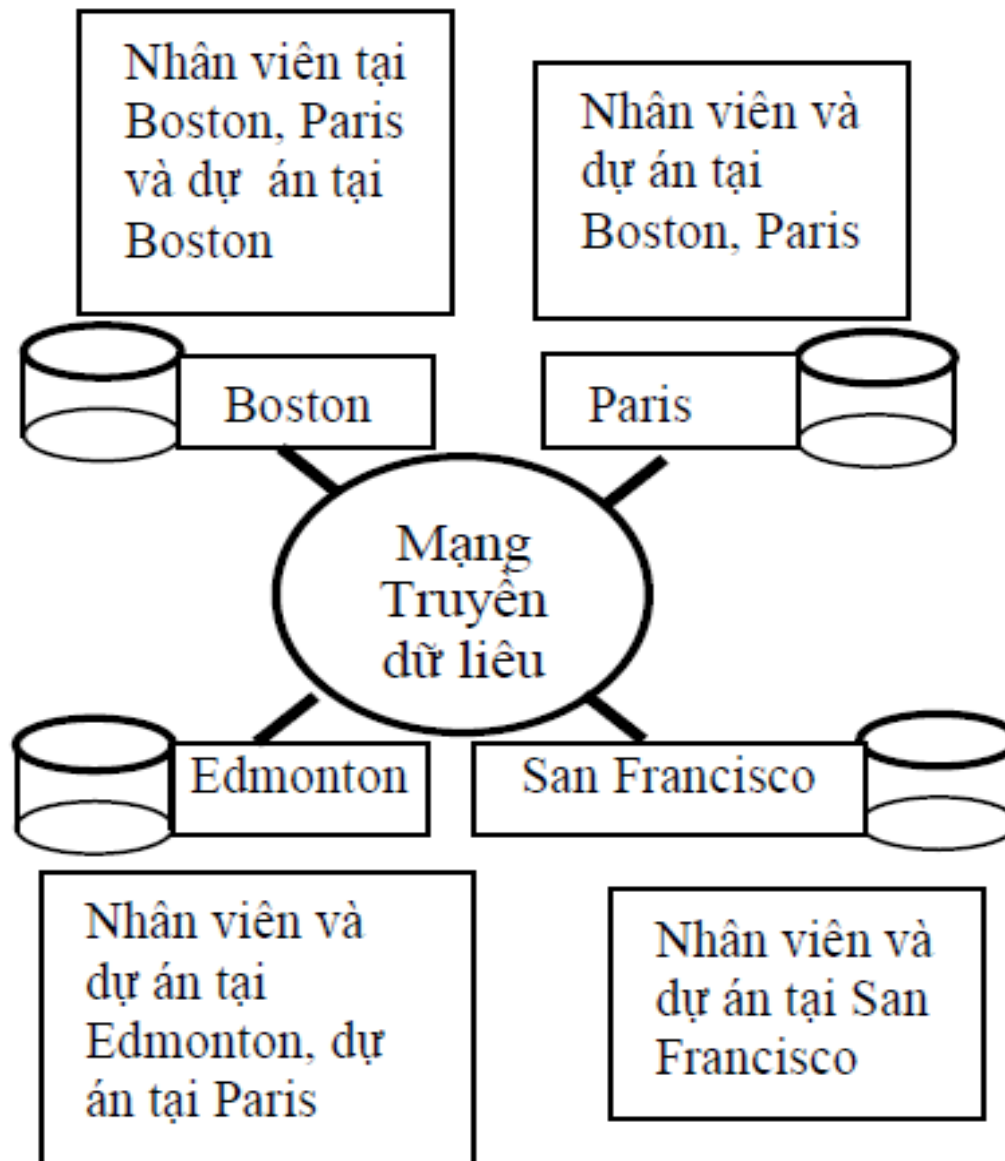
▶ **Nhân bản (Replication)**

Do yêu cầu về hiệu quả và độ tin cậy cho CSDL nên một phần dữ liệu đã được phân mảnh ở trên cần phải được lưu (bản sao) tại một số vị trí khác. Quá trình này được gọi là **nhân bản**.

► **Ví dụ:**

Do tính chất phân tán của công việc kinh doanh của công ty, người ta muốn đặt dữ liệu về các nhân viên của văn phòng ở Edmonton được lưu ở Edmonton, nhân viên của văn phòng ở Boston được lưu ở Boston v.v...

Như vậy cần có một quá trình phân hoạch mối quan hệ này và lưu các phân hoạch tại các vị trí khác nhau. Kết quả phân mảnh và nhân bản trên được thể hiện trên hình sau:



▶ Truy xuất vô hình

Tuy CSDL được phân tán và phân tán trên các nút mạng, nhưng người sử dụng vẫn có thể vấn tin như vấn tin trên CSDL tập trung tại điểm nút thực hiện truy vấn. Việc che đậy này được gọi là **truy xuất vô hình**.

▶ Vô hình liên kết mạng

Người sử dụng được tách ra khỏi mọi chi tiết hoạt động của mạng, thậm chí là không biết có sự hiện diện của mạng nếu được - nghĩa là người sử dụng không biết là mình đang làm việc với CSDL tập trung hay phân tán. Kiểu vô hình này được gọi là **vô hình kết mạng (Network transparency)** hoặc là **vô hình phân tán (Distributed transparency)**.

Vô hình liên kết mạng có thể được chia thành hai loại; vô hình vị trí và vô hình đặt tên.

▶ **Vô hình nhân bản**

Vô hình nhân bản làm cho người sử dụng không thể biết họ đang làm việc với CSDL gốc hay với các bản nhân bản. Vậy vô hình nhân bản làm nhiệm vụ của DBMS.

▶ **Vô hình phân mảnh (Fragmentation transparency)**

- Vô hình phân mảnh làm cho người sử dụng không cần tham gia vào việc phân mảnh và không thể biết họ đang làm việc với CSDL gốc hay với các mảnh đã được phân mảnh từ CSDL gốc.
- **Phân mảnh** là chia CSDL thành các mảnh dữ liệu (Fragment) nhỏ hơn và xử lý mỗi mảnh nhận được như một CSDL độc lập - tức là như một quan hệ. Phân mảnh chỉ được thực hiện khi nó tăng hiệu quả, và có độ tin cậy. Có hai kiểu phân mảnh cơ bản là phân mảnh ngang (Horizontal fragmentation) và phân mảnh dọc (Vertical fragmentation).

6. Triển vọng của các hệ cơ sở dữ liệu phân tán

- ❑ . **Yêu cầu độ tin cậy qua các giao dịch phân tán**
 - Một giao dịch (transaction) là một đơn vị tính toán cơ bản, nhất quán và tin cậy được, bao gồm một loạt các thao tác CSDL như các hành động nguyên tử (atomic action).
 - Nó biến đổi CSDL từ trạng thái nhất quán này sang trạng thái nhất quán khác ngay cả khi có một số lượng giao dịch được thực hiện đồng thời và ngay cả khi có sự cố xảy ra.

6. Triển vọng của các hệ cơ sở dữ liệu phân tán

❑ Nâng cao hiệu năng

Hiệu năng của DBMS phân tán sẽ được nâng cao dựa vào hai điều kiện:

- Một DBMS phân tán có khả năng phân mảnh CSDL mức khái niệm, cho phép dữ liệu ở gần nơi sử dụng (cũng được gọi là cục bộ hóa dữ liệu - data localization).
- Tính chất song hành của các hệ phân tán có thể được tận dụng để thực hiện song hành liên vấn tin và nội vấn tin. Song hành liên vấn tin là khả năng thực hiện cùng một lúc nhiều câu vấn tin còn song hành nội vấn tin là tách một câu vấn tin thành các câu vấn tin con, mỗi câu sẽ được thực hiện tại một vị trí và truy xuất các phần khác nhau của CSDL phân tán.

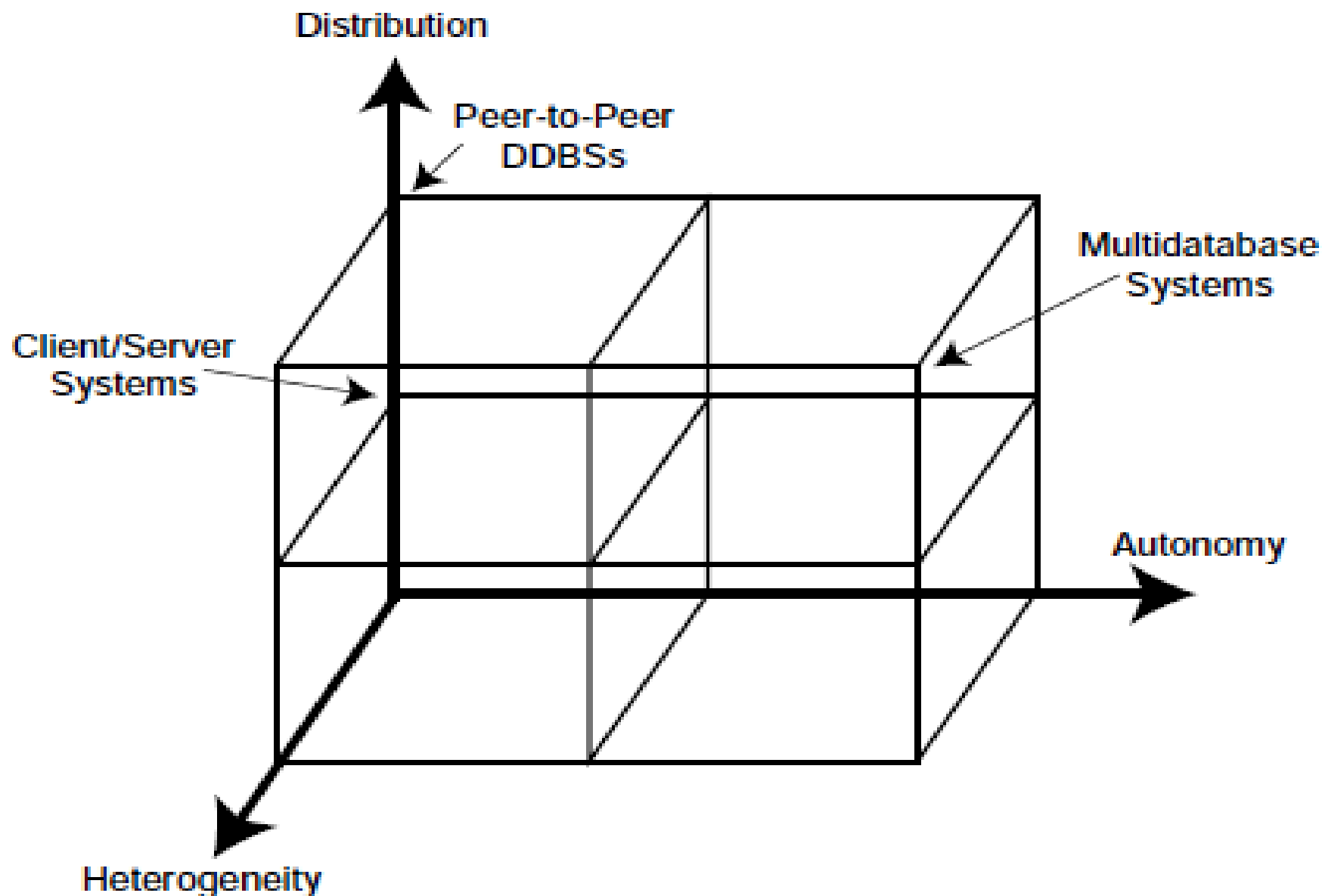
7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.

- ▶ **Kiến trúc của một hệ thống là xác định cấu trúc của nó.**
 - Các thành phần của các hệ thống được xác định,
 - Chức năng của mỗi thành phần được mô tả,
 - Các mối quan hệ trung gian và tương tác giữa các thành phần này được định nghĩa.
- ▶ **Mục tiêu định nghĩa kiến trúc là xây dựng các DBMS phân tán với khả năng cung cấp được các chức năng như:**
 - Vô hình,
 - Độ tin cậy qua các giao dịch phân tán,
 - Hiệu quả và đặc tính mở rộng,...

7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.

- ☐ Kiến trúc hệ Client/Server
- ☐ Kiến trúc hệ Peer-To-Peer
- ☐ Kiến trúc Multi-DBMS

7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.

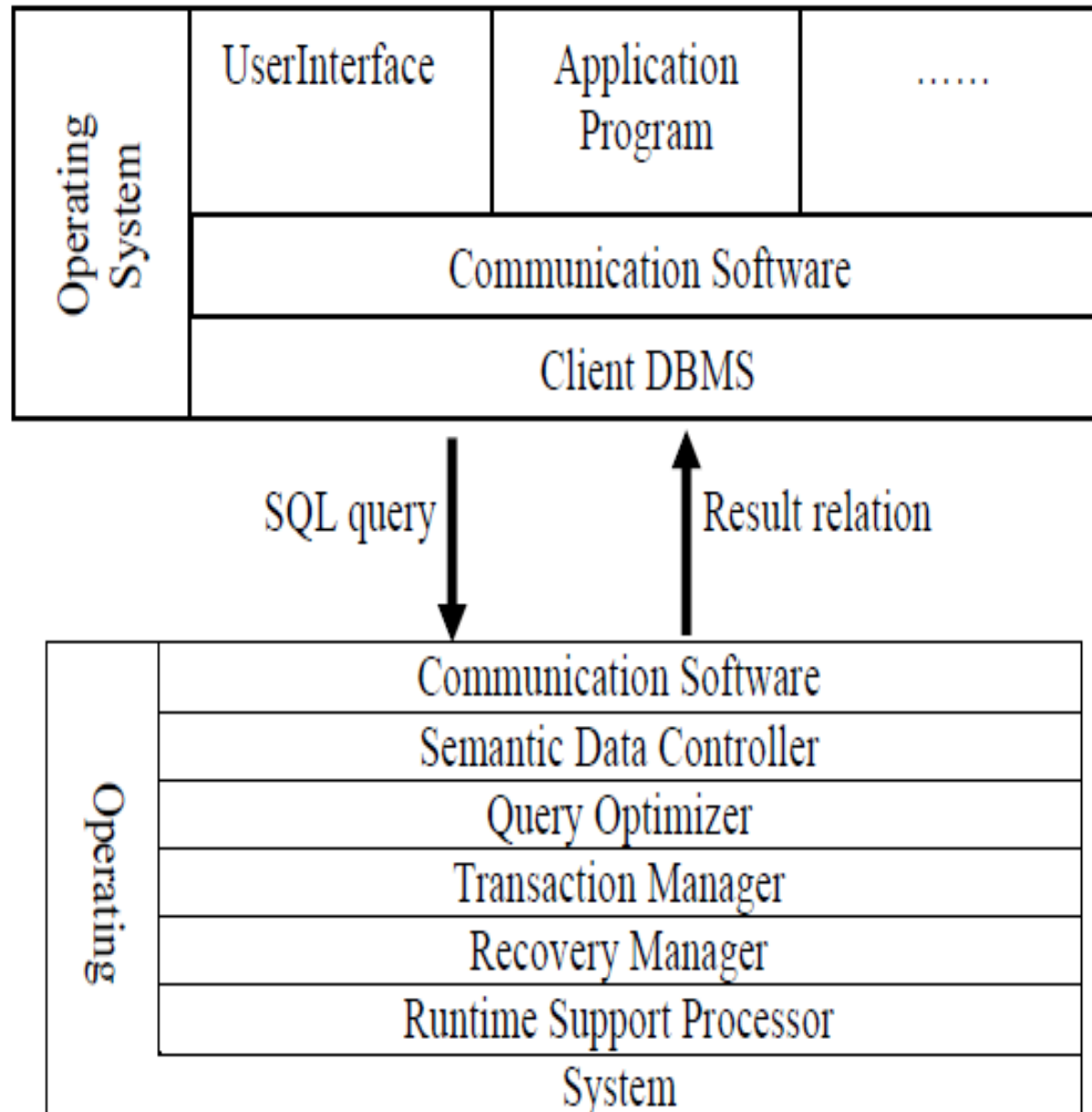


7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.

❑ Kiến trúc Client/Server

- ❖ Chia chức năng thành hai lớp: Chức năng chủ và chức năng khách.
- ❖ Đây là kiến trúc hai cấp có tác dụng làm giảm tính phức tạp không những cho DBMS mà cả khi phân tán dữ liệu.
 - Server chịu trách nhiệm thực hiện mọi xử lý và tối ưu hóa vận tin, quản lý giao dịch và quản lý thiết bị,...
 - Tại các Client ngoài các ứng dụng giao diện, còn có riêng một DBMS chịu trách nhiệm quản lý dữ liệu nhận từ Server và có thể quản lý cả các phiên giao dịch.

Kiến trúc Client/Server



Kiến trúc Client/Server

► Các kiến trúc Client/Server đặc trưng nhất:

(1) **n-Client** ----- **1-Server**

Kiến trúc này không có nhiều khác biệt với CSDL tập trung, vì CSDL được lưu trữ trên một Server duy nhất và có 1 phần mềm để quản lý CSDL này.

(2) **n-Client** ----- **n-Server**

Kiến trúc này có hai cách quản lý:

- Mỗi Client tự quản lý kết nối với Server
- Mỗi Client chỉ biết Server trực tiếp của mình

7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.

❑ Kiến trúc Peer-To-Peer

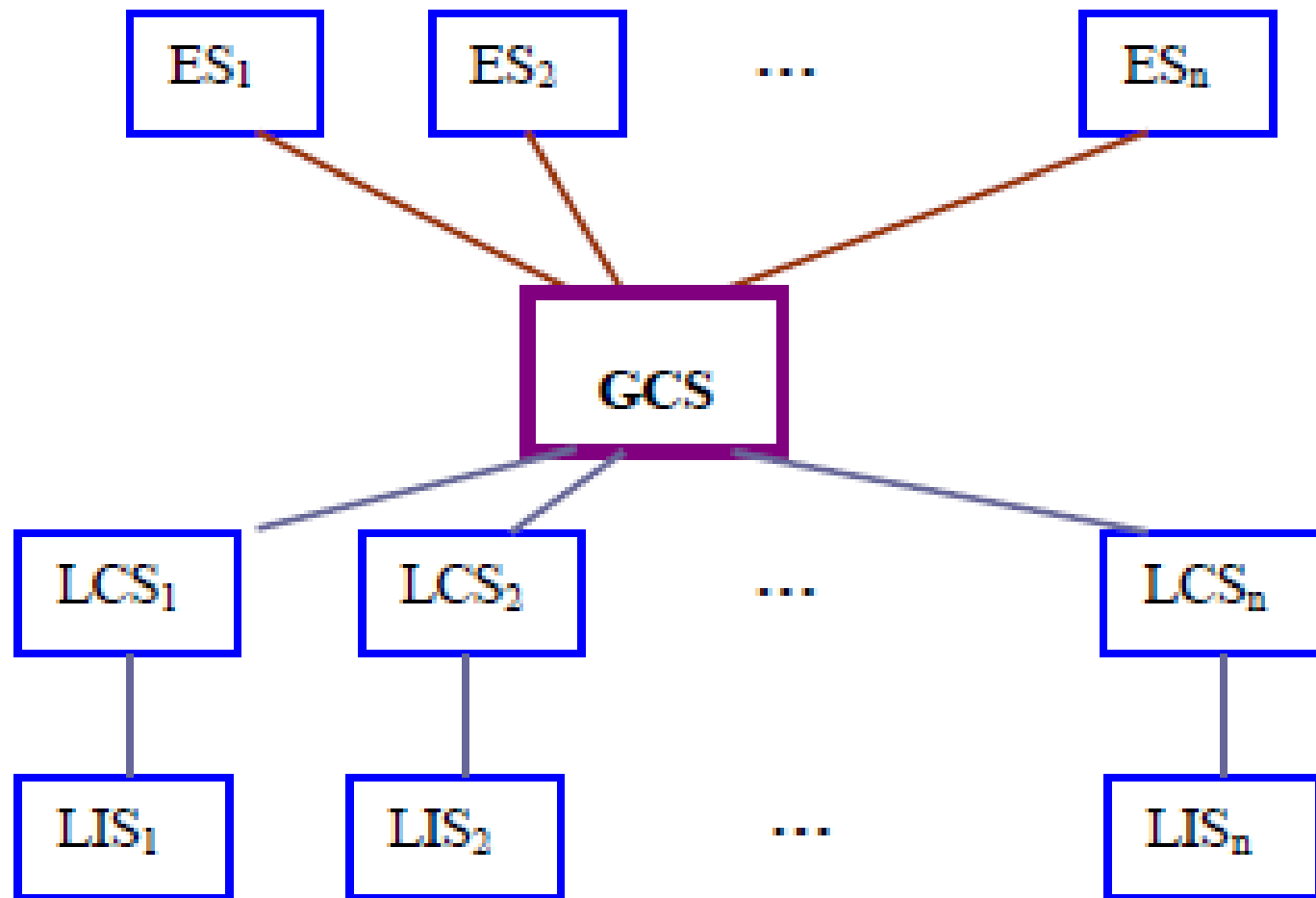
- Trong Peer-To-Peer, việc tổ chức dữ liệu vật lý trên mỗi Workstation có thể khác nhau, như vậy phải có một định nghĩa riêng cho mỗi Workstation mà ta gọi là **lược đồ cục bộ LIS** (Local Internal Schema).
- Dữ liệu trong một CSDL phân tán thường được phân mảnh và nhân bản. Để xử lý sự phân mảnh và nhân bản, chúng ta cần mô tả tổ chức logic của dữ liệu được lưu tại các Workstation nên cần phải có thêm **lược đồ khái niệm cục bộ LCS** (Local Conceptual Schema) trong kiến trúc.
- Như vậy có thể thấy GCS là hợp của các LCS.

7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.

❑ Kiến trúc Peer-To-Peer (tt)

- Bức tranh toàn cục về dữ liệu của cả công ty hay cả xí nghiệp,... được mô tả bởi **lược đồ toàn cục GCS** (Global Conceptual Schema), nó được dùng để mô tả cấu trúc logic của dữ liệu được lưu tại các Workstation.
- Cuối cùng là các ứng dụng và truy cập CSDL được gọi là **lược đồ ngoài ES** (External Schema).

Kiến trúc Peer- To- Peer




7. Kiến trúc hệ quản trị cơ sở dữ liệu phân tán.

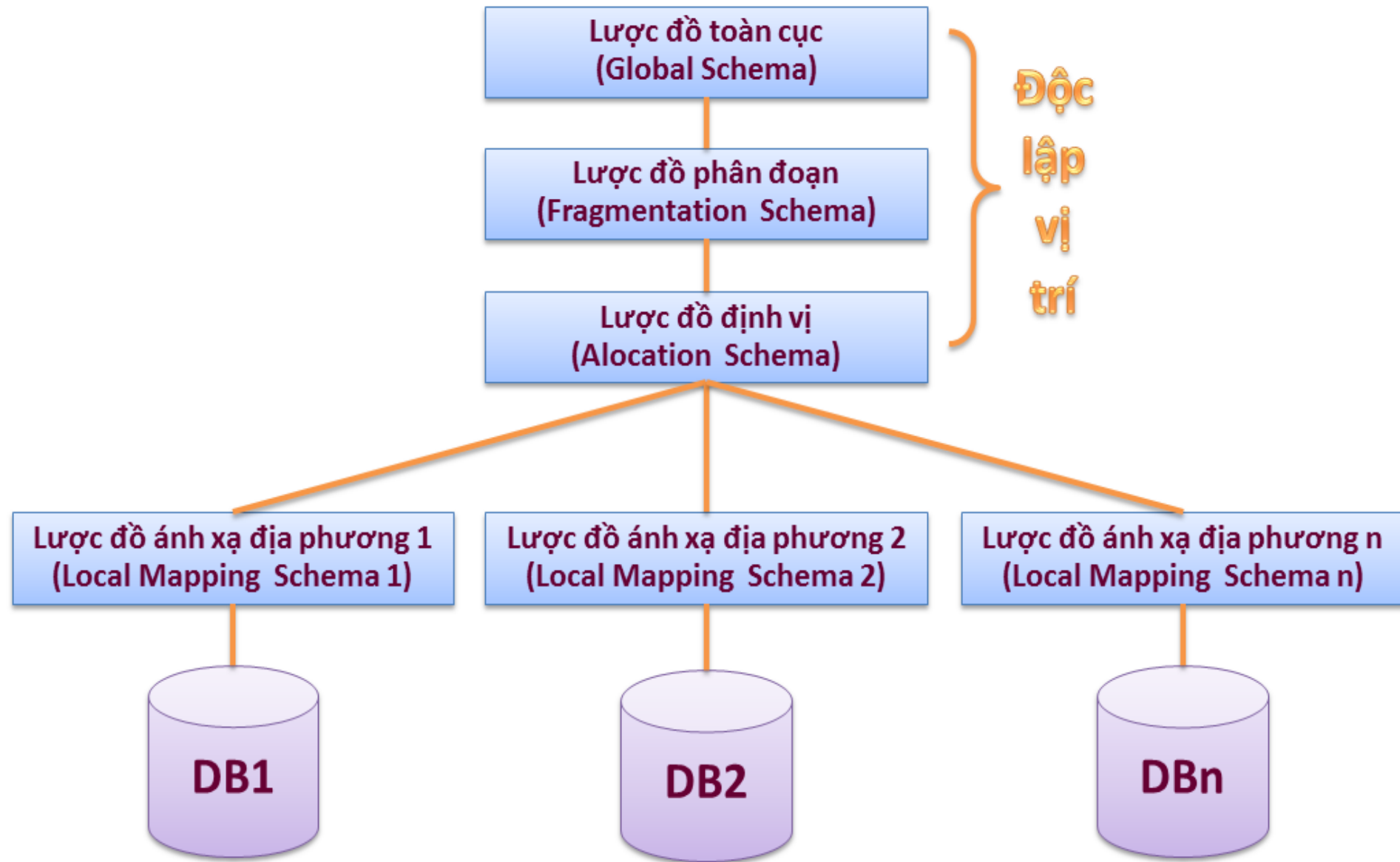
□. Kiến trúc đa hệ CSDL

- ▶ **Các mô hình sử dụng lược đồ khái niệm toàn cục.**
- ▶ **Trong phức hệ CSDL, GCS là tích hợp các lược đồ ngoài của các CSDL tự vận hành cục bộ**

8. Một số vấn đề căn bản khi nghiên cứu CSDL phân tán

- ▶ **Kiến trúc cơ bản của CSDL phân tán.**
 - ▶ **Phân mảnh dữ liệu.**
 - ▶ **Tính trong suốt phân tán.**
 - ▶ **Tính trong suốt phân tán dùng cho ứng dụng chỉ đọc.**
 - ▶ **Tính trong suốt phân tán dùng cho ứng dụng cập nhật.**
 - ▶ **Các loại truy xuất CSDL phân tán.**
- 

Kiến trúc cơ bản của một cơ sở dữ liệu phân tán



▶ **Lược đồ toàn cục:**

- Xác định toàn bộ dữ liệu được lưu trữ trong CSDLPT.
- Được định nghĩa như trong CSDL tập trung.
- Trong mô hình quan hệ: lược đồ toàn cục là các quan hệ và mối liên kết giữa chúng.

▶ **Lược đồ phân đoạn:**

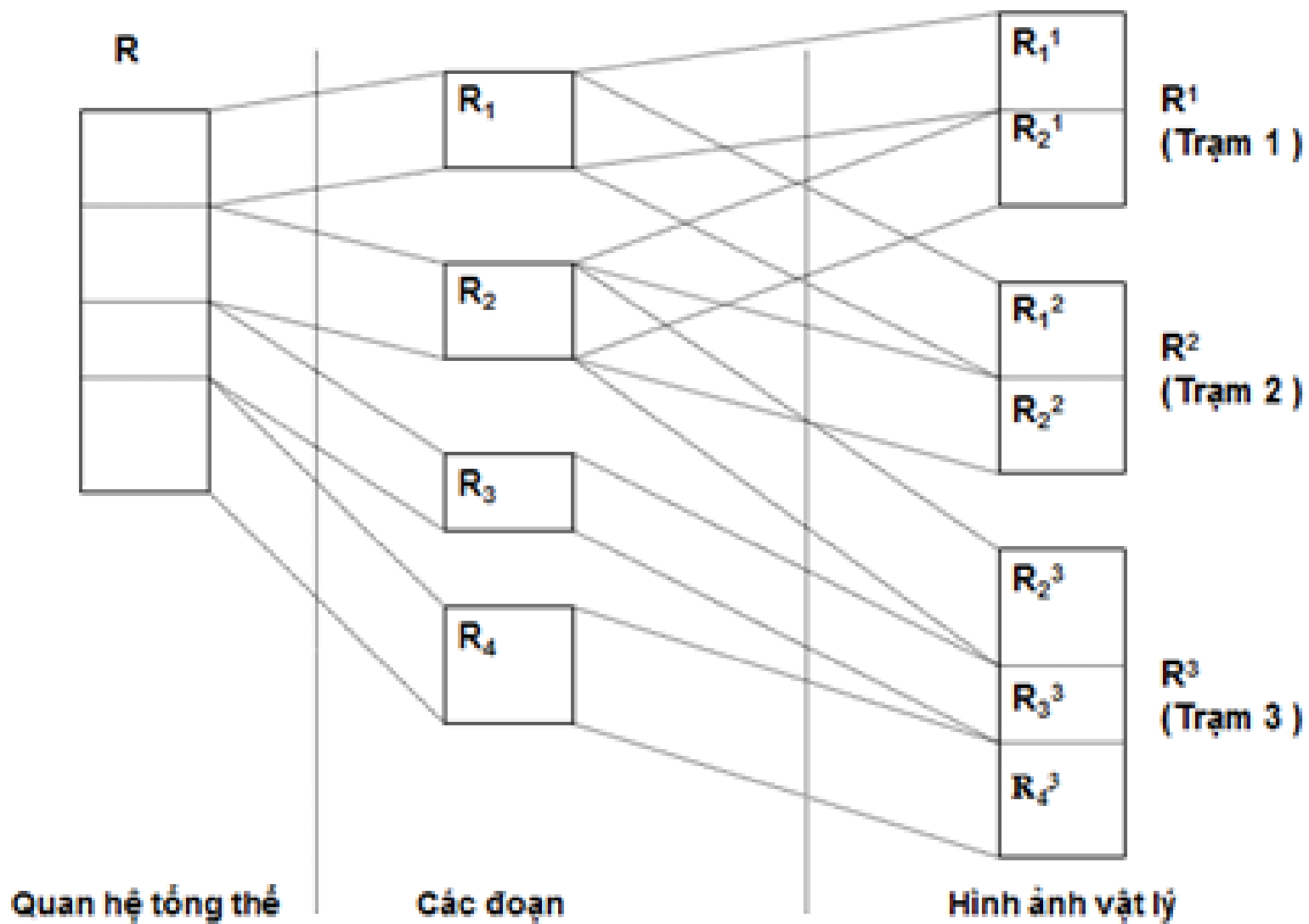
- Mỗi quan hệ tổng thể có thể được chia thành các phần không giao nhau gọi là phân đoạn (Fragment).
- Có nhiều cách khác nhau để phân đoạn: Phân đoạn dọc, phân đoạn ngang, phân đoạn hỗn hợp.
- Các đoạn được mô tả bằng tên của quan hệ tổng thể cùng với chỉ mục đoạn. Ví dụ Ri là đoạn thứ i của quan hệ toàn cục R.

► **Lược đồ định vị:**

- Xác định đoạn dữ liệu nào được định vị tại trạm nào trên mạng.
- R_{ij} : Cho biết đoạn thứ i của quan hệ tổng thể R được định vị trên trạm j .

► **Lược đồ ánh xạ địa phương:**

- Ánh xạ các ảnh vật lý và các đối tượng được lưu trữ tại một trạm.



Kiến trúc cơ bản của một cơ sở dữ liệu phân tán

- ▶ Ba yếu tố được suy ra từ kiểu kiến trúc này là:
 1. Tách rời khái niệm phân đoạn dữ liệu với khái niệm định vị dữ liệu.
 2. Biết được dữ liệu dư thừa
 3. Độc lập với các DBMS địa phương
- ▶ *Ba yếu tố này tương ứng với ba mức trong suốt tương ứng.*

Kiến trúc cơ bản của một cơ sở dữ liệu phân tán

1. Tách rời khái niệm phân đoạn dữ liệu với khái niệm định vị dữ liệu.

❖ **Phân đoạn dữ liệu**, bao gồm những công việc mà người lập trình ứng dụng làm việc với quan hệ tổng thể, phân chia quan hệ tổng thể thành các đoạn.

- Thông qua tính **trong suốt phân đoạn** (*fragmentation transparency*) người lập trình sẽ nhìn thấy được những đoạn dữ liệu bị phân chia như thế nào.

❖ **Định vị dữ liệu** lại liên quan đến các công việc của người sử dụng và người lập trình ứng dụng trên các đoạn dữ liệu được định vị tại các trạm.

- Thông qua tính **trong suốt vị trí** (*location transparency*) người lập trình sẽ biết được vị trí của các đoạn dữ liệu

Kiến trúc cơ bản của một cơ sở dữ liệu phân tán

2. Biết được dữ liệu dư thừa

- ❖ Người lập trình ứng dụng có thể biết được dư thừa dữ liệu ở các trạm.
- ❖ Trên hình vẽ trên, chúng ta thấy rằng hai ảnh vật lý R_2 và R_3 có trùng lặp dữ liệu. Do đó các đoạn dữ liệu trùng nhau có thể tránh được khi xây dựng các khối ảnh vật lý.

Kiến trúc cơ bản của một cơ sở dữ liệu phân tán

3. Độc lập với các DBMS địa phương

- ❖ Tính chất này còn được gọi là *trong suốt ánh xạ địa phương (local mapping transparency)*,
- ❖ Cho phép chúng ta khảo sát các vấn đề về quản lý CSDL phân tán mà không cần phải hiểu rõ mô hình dữ liệu của DBMS địa phương đang sử dụng.

Phân mảnh dữ liệu (Fragmentation)

- ▶ Việc phân tán dữ liệu được thực hiện trên cơ sở cấp phát các tập tin cho các nút trên một mạng máy tính. Các nút mạng thường nằm ở các vị trí địa lý khác nhau trải rộng trên một diện tích lớn. Do vậy để tối ưu việc khai thác thông tin thì dữ liệu không thể để tập trung mà phải **phân tán trên các nút của mạng**.
- ▶ Một quan hệ không phải là một đơn vị truy xuất dữ liệu tốt nhất. (Ví dụ)
- ▶ Do vậy phân rã một quan hệ thành nhiều mảnh, mỗi mảnh được xử lý như một đơn vị sẽ cho phép thực hiện nhiều giao dịch đồng thời. Một câu truy vấn ban đầu có thể được chia ra thành một tập các truy vấn con, các truy vấn này có thể được thực hiện song song trên các mảnh sẽ giúp cải thiện tốc độ hoạt động của hệ thống.

- ▶ **Thể hiện của các quan hệ chính là các bảng**, vì thế vấn đề là tìm những cách khác nhau để chia một bảng thành nhiều bảng nhỏ hơn.
- ▶ **Có hai phương pháp khác nhau:** Chia bảng theo chiều dọc và chia bảng theo chiều ngang.
- **Chia dọc** ta được các quan hệ con mà mỗi quan hệ chứa một tập con các thuộc tính của quan hệ gốc – gọi là phân mảnh dọc.
- **Chia ngang** một quan hệ ta được các quan hệ con mà mỗi quan hệ chứa một số bộ của quan hệ gốc – gọi là phân mảnh ngang.
- Ngoài ra còn có một khả năng **hỗn hợp**, đó là phân mảnh kết hợp cách phân mảnh ngang và dọc.

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)

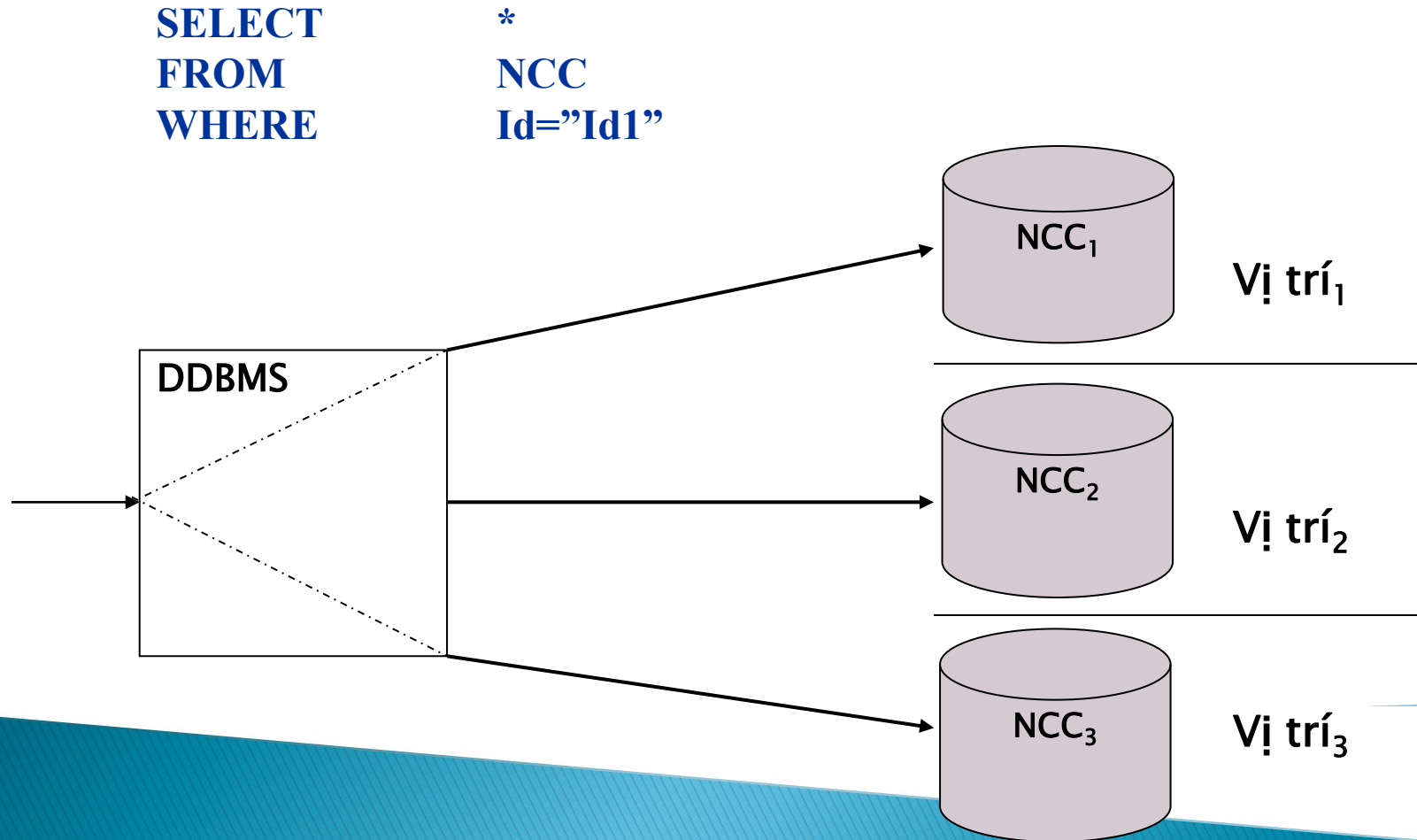
NCC2 (Id, Tên, Tuổi)

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



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"*.

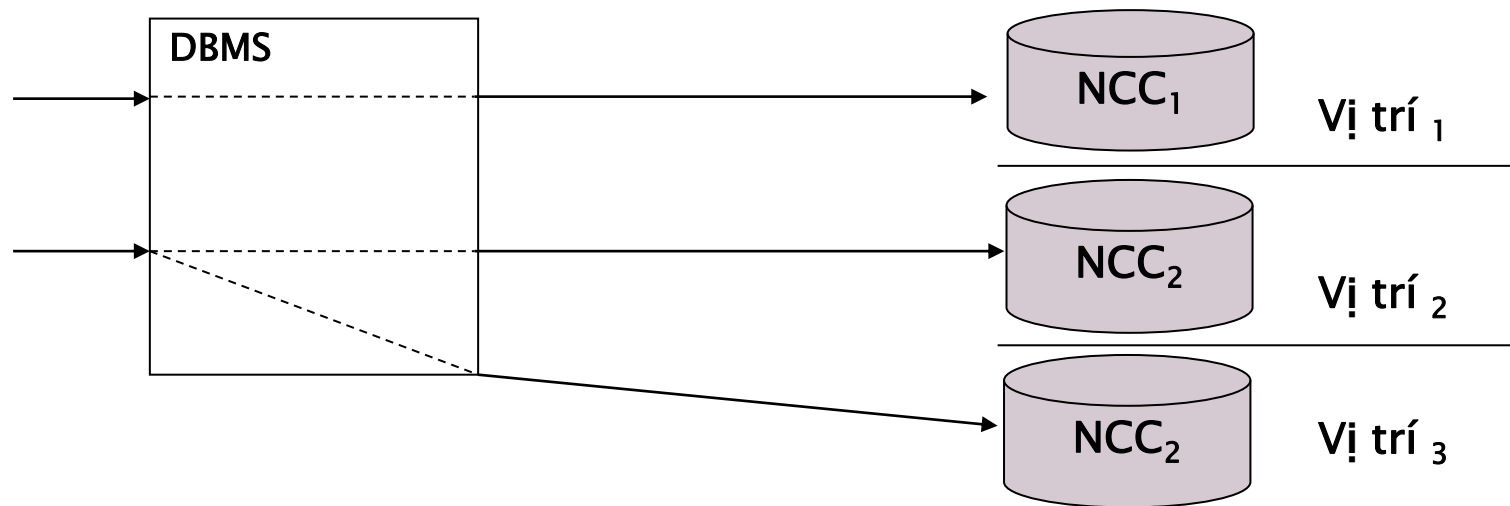
```
SELECT      *  
FROM        NCC1  
WHERE       Id="Id1"
```

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, \dots
- ❖ Ở đây quan hệ NCC_2 được sao làm hai bản trên hai vị trí₂ và vị trí₃, ta chỉ cần tìm thông tin trên quan hệ NCC_2 mà không cần quan tâm nó ở vị trí nào.

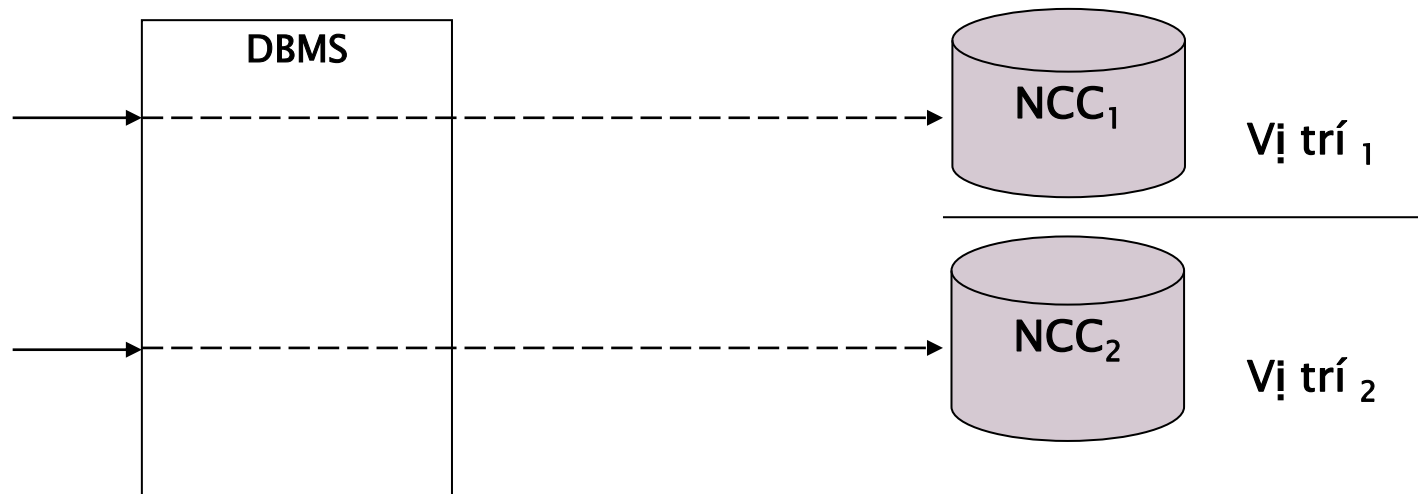


Sự trong suốt về vị trí

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:

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

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

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

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

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

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

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

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

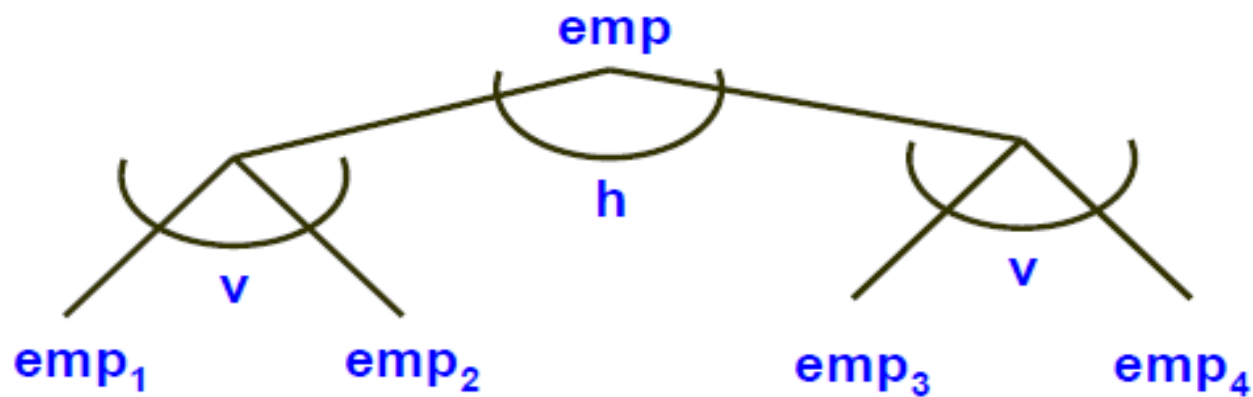
- ▶ Mục dữ liệu bị sửa không có trong vị từ định tính.
- ▶ Mục dữ liệu bị sửa có trong vị từ định tính và giá trị của vị từ định tính không bị thay đổi khi thay thế dữ liệu cũ và dữ liệu mới.
- ▶ Mục dữ liệu bị sửa có trong vị từ định tính và giá trị của vị từ định tính bị thay đổi khi thay thế dữ liệu 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



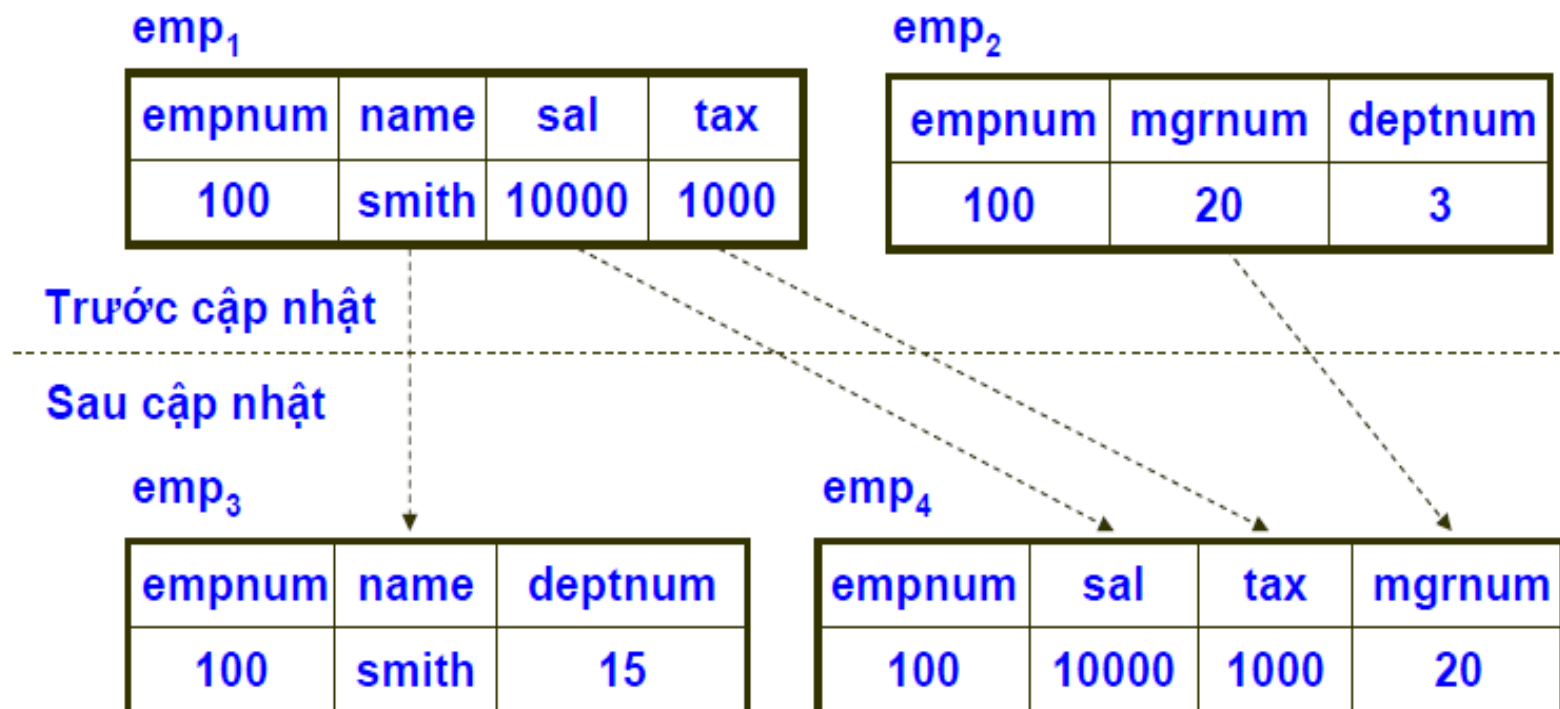
$emp_1 = \Pi_{empnum, name, sal, tax} \sigma_{deptnum \leq 10} emp$

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

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

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

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);
```

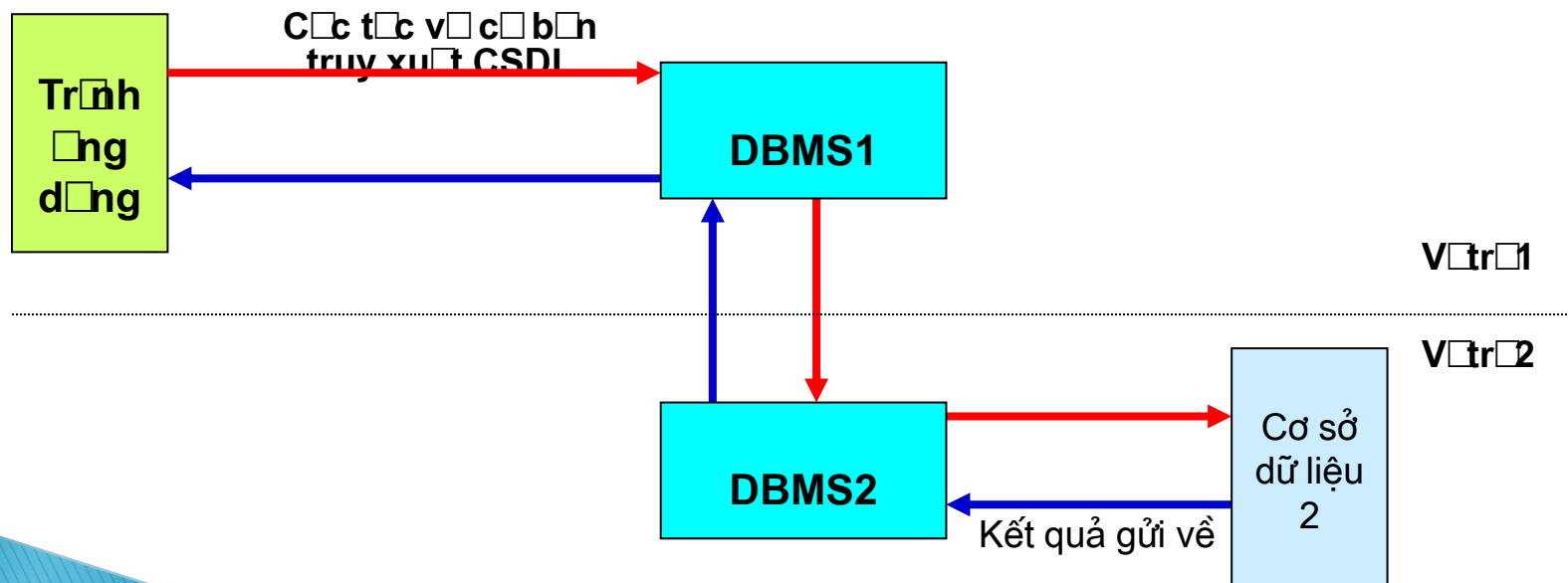
Tính trong suốt phân tán dùng cho ứng dụng cập nhật

```
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;
```

Các loại truy xuất CSDL phân tán

1. Truy xuất từ xa thông qua các tác vụ cơ bản:

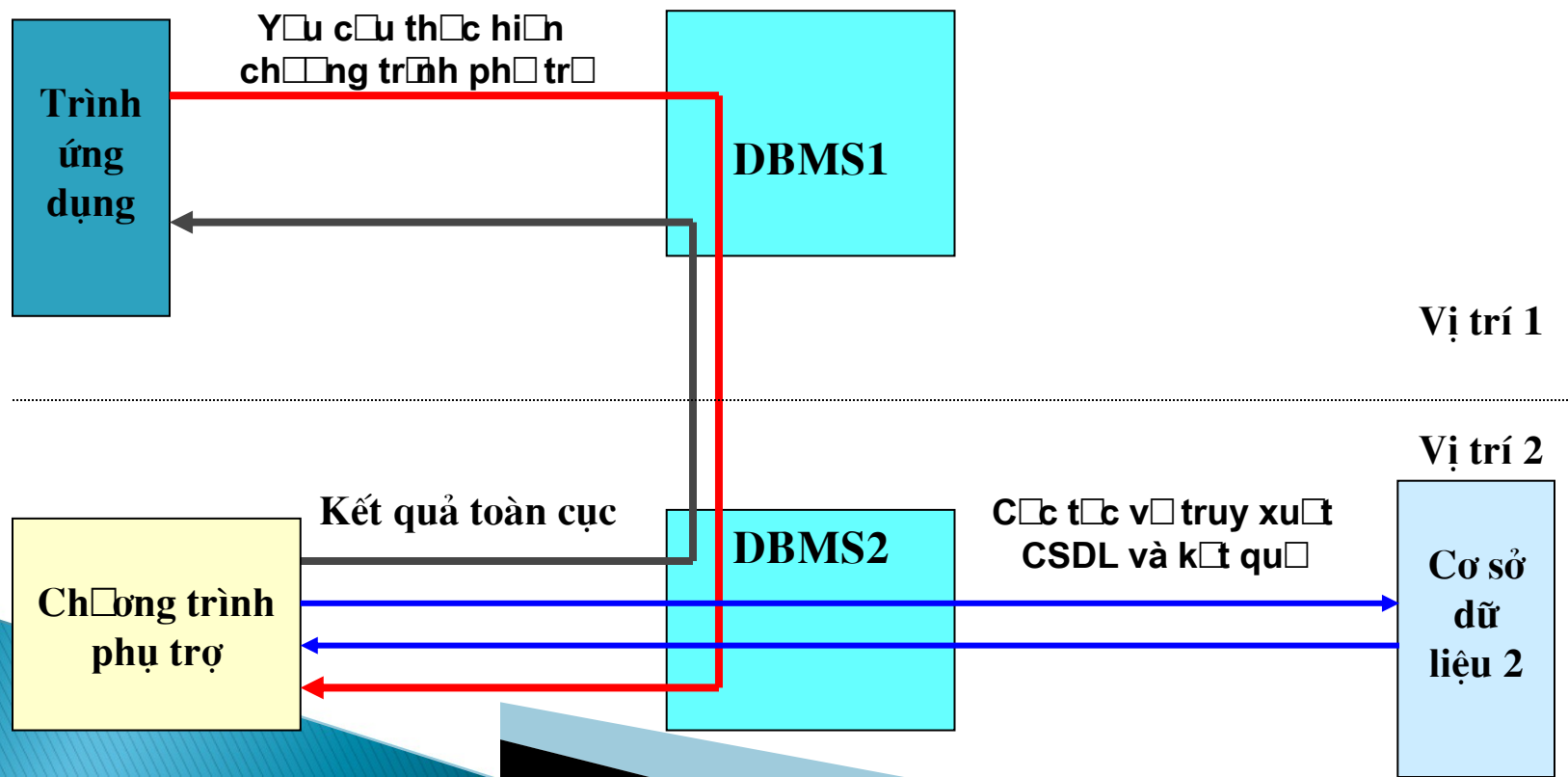
- Ứng dụng phát ra một yêu cầu truy xuất CSDL ở một vị trí nào đó. Yêu cầu này sẽ được hệ quản trị CSDL phân tán gửi đến vị trí chứa dữ liệu đó. Thực hiện xong sẽ gửi kết quả về.



Các loại truy xuất CSDL phân tán

2. Truy xuất từ xa thông qua chương trình phụ trợ

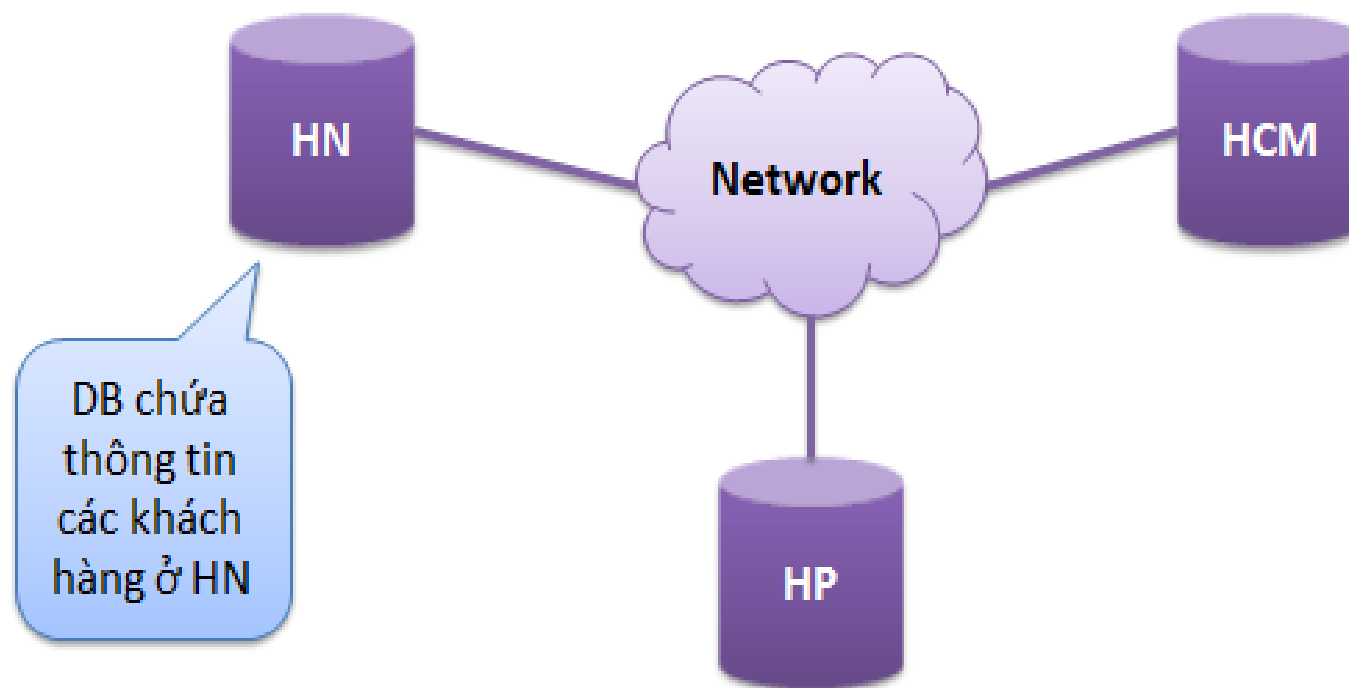
- ▶ Một ứng dụng yêu cầu thực hiện một chương trình phụ trợ đặt tại vị trí từ xa. Chương trình phụ trợ này sẽ truy xuất CSDL từ xa và trả lại kết quả cho ứng dụng đang yêu cầu.



Review

- ▶ **Cơ sở dữ liệu phân tán (Distributed Database)** là cơ sở dữ liệu được phân mảnh và được lưu trữ trên các trạm trong hệ thống mạng.
- ▶ Cơ sở dữ liệu phân tán là quan trọng trong kinh tế, tổ chức và kỹ thuật với nhiều lý do khác nhau. Chúng có thể được cài đặt trên một mạng máy tính có phạm vi rộng lớn hoặc nhỏ bé.
- ▶ Hiện nay, các DDBMSs thương mại đều tích hợp các ứng dụng phân tán nên rất tiện cho người sử dụng.

Review



Review

- ▶ **Hệ quản trị cơ sở dữ liệu phân tán (DDBMS)**
 - Cho phép người dùng tạo, sử dụng csdl.
 - Đảm bảo an ninh (cấp phát quyền, 1 nhóm người được sử dụng, ...)
 - Đảm bảo tính trong suốt của csdl (Transperence)
 - Người dùng sử dụng như csdl tập trung.
 - Truy vấn tập trung → Truy vấn phân tán.
- ▶ **Các ứng dụng:**
 - ▶ Ứng dụng cục bộ: Chỉ quan tâm tới dữ liệu ở 1 trạm.
 - ▶ Ứng dụng toàn cục: Liên quan đến nhiều trạm.

Review: Ưu nhược điểm của cơ sở dữ liệu phân tán

▶ **Ưu điểm:**

- Dữ liệu gần với nơi xử lý → Hiệu suất cao.
- Tính sẵn sàng của hệ thống cao: Nếu một trạm bị lỗi sẽ không ảnh hưởng tới các trạm khác trong hệ thống.
- Việc tăng các trạm sử dụng trong hệ thống là đơn giản nên việc mở rộng CSDL là dễ dàng.

▶ **Nhược điểm:**

- ▶ Lưu trữ: Ngoài lược đồ CSDL như trong CSDL tập trung (Thuộc tính, kiểu dữ liệu, ...) còn thêm các lược đồ phân đoạn CSDL, lược đồ định vị CSDL (cho biết các đoạn được lưu trữ ở đâu).
- ▶ Xử lý: Truy vấn tập trung là đơn giản còn truy vấn phân tán phức tạp.
- ▶ An toàn: CSDL được lưu trữ ở nhiều nơi nảy sinh vấn đề: đảm bảo an toàn dữ liệu khi truyền qua mạng.